Erasmus University Rotterdam

Erasmus School of Economics

Master's Thesis Quantitative Finance

GARCH Tree Forecasting

Name student: D. Arbouw

Student number: 389011

Supervisor: dr. O. Kleen

Second assessor: dr. A. Tetereva

Date current version: 26-2-2022

Abstract

This thesis investigates the forecasting performance of the GARCH tree model. This is done by applying the unconditional and conditional Giacomini and White (2006) tests, and MCS procedure of Hansen et al. (2011). Using a dataset of daily S&P500 returns, we find that the GARCH tree is able to outperform the GARCH and GJR-GARCH forecasts on a 1-day horizon, but this is only true when assuming t-distributed innovations and adding more splitting variables to the GARCH tree. At longer horizons, the performance is similar or worse than the competing models. This is especially true for the random forests. Extensions show the forecast errors depend on the number of terminal nodes and that using the AIC or BIC does not lead to the best GARCH tree model. Other factors influencing the main results are the block length of the stationary bootstrap for the random forests, the loss function and chosen stock index. From a practical perspective, the applied variance targeting estimation method greatly reduced the computation time.

Keywords: GARCH tree • variance targeting • • forecasting • predictive ability tests • model confidence set

Contents

1 Introduction **1**

2 Methodology **4**
2.1 GARCH Model & Variance Targeting 4
2.2 GARCH Tree Model 5
2.3 Tree Specifications 8
2.4 GARCH Random Forest 9
2.5 Stationary Bootstrap 9
2.6 GJR-GARCH 10
2.7 GARCH Tree Forecasting 10
2.7.1 Design 10
2.7.2 Giacomini-White Tests 11
2.7.3 Model Confidence Set 12

3 Data **13**

4 Results **17**
4.1 Model Estimates 17
4.2 Forecasting performance 21
4.3 Extensions 26
4.3.1 Varying Maximum Number of Terminal Nodes 26
4.3.2 Varying Stationary Bootstrap Block Length 27
4.3.3 Squared Error Loss 28
4.3.4 Other Indices 30
4.3.5 Lower Frequency 31

5 Conclusion **32**

References **34**

A Appendix **36**
A.1 Time Series of κ and α 36
A.2 Squared Error Loss: 20-day-ahead Forecasts 38
A.3 Forecasting Results - 1-month-ahead 38

1 Introduction

Volatility is a key subject in finance. The conditional variance of financial time series is important for portfolio allocation problems, risk management and asset pricing. The size and sign of daily price movements is related to the arrival of news to financial markets. Therefore, accurate forecasts of the conditional variance aid, for example, central banks to make good decisions, risk management departments at large financial institutions to hedge risk or trading algorithms to be profitable. These benefits has led to an high interest in modelling the conditional variance of financial assets, where volatility is usually modelled using the Generalized Autoregressive Conditional Heteroskedastic (GARCH) framework of Bollerslev (1986).

The success of GARCH-type models lies in the fact that they are able to capture some important characteristics of financial assets, namely (1) that returns are not normally distributed but have heavier tails, (2) that returns are not correlated over time, and (3) that squared or absolute returns are correlated over time and have long memory. The latter is usually referred to as volatility clustering. Volatility clustering means that the second moment of financial returns depends on its past. These temporal dependencies lead to the alternation of periods of high volatility with periods of low volatility. The workhorse version in the GARCH-type framework is the GARCH(1,1) model where today's volatility is a deterministic function of of yesterday's volatility and yesterday's return. Although volatility in GARCH models is not directly observable, it is perfectly predictable one day ahead because it depends only on variables known the day before. The quality of the GARCH(1,1) predictions has been a widely discussed topic in the literature. Like many others, Hansen and Lunde (2005) have studied the performance of the traditional GARCH(1,1) forecasts to a large number of different extensions of the GARCH model. They find that the GARCH(1,1) model does well in forecasting exchange rate volatility but that the alternatives provide more accurate forecasts when forecasting financial asset volatility. The crucial reason why the other models do better than the GARCH(1,1) model is the leverage effect present in the conditional variance of financial time series. This leverage or asymmetric effect originates from the finding that negative returns have a larger impact on volatility than positive returns. The leverage effect thus creates a beneficial nonlinear relation in the volatility equation of the GARCH model.

The leverage effect is a simple and straightforward extension. A more flexible but also more complicated version of the GARCH model was introduced by Audrino and Bühlmann (2001). Their model makes use of a decision tree to relate yesterday's volatility and yesterday's return to today's volatility. Decision trees, arising from the machine learning literature, introduce thresholds to create nonlinear relations between input and output and have shown promising results in other applications. The GARCH tree of Audrino and Bühlmann (2001) allows for different GARCH parameters at each terminal node of the decision tree. Its main advantage is that it nests the standard GARCH(1,1) model while at the same time allowing for complex data-driven relations. Because Audrino and Bühlmann (2001) have not investigated the forecasting performance of their GARCH tree model, the research question of this thesis is as follows

How well can the GARCH tree forecast volatility compared to the GARCH(1,1) model?

To answer the research question we compare the forecasting performance of the GARCH tree with the GARCH(1,1) (henceforth simply GARCH) using data of the S&P500 stock market index. Because the leverage effect is so prominent in financial volatility, the GJR-GARCH model of Glosten et al. (1993) is included to see whether the GARCH tree is a real improvement or just capturing the effects of existing models. The GARCH tree is very flexible, which is why the GARCH tree will be specified in many different forms. As shown by Audrino and Bühlmann (2001), the estimation results of the GARCH tree depend on the distributional assumption of the innovations. These will be either normally or t-distributed. Other forms make use of additional splitting variables to determine the thresholds or random forests to combine multiple GARCH trees. After the forecasting exercise, several extensions perform a sensitivity analysis to investigate the effect of the assumptions made beforehand.

The GARCH tree model of Audrino and Bühlmann (2001) is part of the more general class of semiparametric ARCH models of Linton and Mammen (2005). In Linton and Mammen (2005), the news impact curve, which describes the relation between volatility and new return observations, is estimated nonparametrically and based on kernel smoothing. This general version differs from the type of models that use a certain amount of regimes or thresholds to capture nonlinearities in volatility, like the leverage effect. For example, Medeiros and Veiga (2009) introduced a multiple regime GARCH model. The number of regimes in their model is based on a sequential testing procedure where new regimes are added until the null hypothesis of a redundant regime can no longer be rejected. This is different from the GARCH tree model where the number of regimes is chosen using a model selection criterion.

The tree structure has been exploited by others as well. Audrino and Trojani (2011) use the GARCH tree in a multivariate setting with thresholds in volatilities and correlations. Goulet Coulombe (2020) develops a random forest model, which combines multiple trees, in a macroeconomic setting. There, the random forest is used to obtain time varying parameters leading to superior forecasting performance over fixed or rolling window alternatives. Regarding rolling windows, Oh and Patton (2021) introduces a local estimation method where local is not necessarily based on time but could also be determined by other variables. In their GARCH forecasting study, the local estimation method with realized variance and time as variables that define the distance of certain observations, significantly outperformed the standard GARCH model. This study relates to ours by using variables outside of the model itself to determine its parameter values, which is what here the splitting variables in the tree will do. It is different from the GARCH tree in the sense that the GARCH tree uses hard instead of smooth thresholds for the parameters and that time is not a variable in the GARCH tree.

The forecasting results of Medeiros and Veiga (2009) , Oh and Patton (2021), and Liu et al. (2020), indicate the forecasting performance of GARCH models can be improved using multiple regimes, although in these cases the transition is smooth. The GJR-GARCH model of Glosten et al. (1993) captures the leverage effect by a hard threshold between negative and positive returns and is know to outperform the GARCH model. The GARCH tree also uses hard thresholds and can easily incorporate the leverage effect, which is why we can expect the GARCH tree to do better than standard GARCH as well.

Several contributions have been made to the existing literature. We have performed a fore-

casting study to show the potential of the GARCH tree model. In addition, more splitting variables were added to the GARCH tree, which has only been modeled with past return and variance previously. The implementation of the GARCH tree with the variance targeting estimation method also shows the computation time can be reduced substantially, making the GARCH tree more accessible in practice. Finally, the extensions of Section 4.3 reveal the specification of the trees is important for their performance.

Our main analysis shows that for 1-day-ahead forecasts, the GARCH tree is able to do better than the GARCH and GJR-GARCH model. This is, however, only true for the GARCH tree with t-distributed innovations and the additional splitting variables included. The other GARCH tree implementations perform on par with or worse than the traditional models. Especially the random forests, which were expected to do better given its model averaging and variable selection characteristics, disappoint. At longer horizons, the GARCH tree models are not able to do better, which could be because of the recursion we applied to obtain multi-step-ahead forecasts. They are obtained with the parameters of the most recent terminal node, which might be irrelevant at long horizons, causing the volatility prediction to revert too fast or too slow to the unconditional variance.

The main findings are extended in several directions. First, because in the estimation of the GARCH trees the AIC statistic failed to select a lower than the maximum number of terminal nodes as optimal, we manually lowered the number of terminal nodes in the GARCH trees. This shows that the maximum of six terminal nodes does not lead to the lowest average loss. Three nodes seems to be sufficient. Replacing the AIC with the BIC does not really overcome this problem. Second, the expected block length of the stationary bootstrap has a large impact on the performance of the random forests. Although at a block length of 500 the average losses are at a minimum, the random forests still do worse than all other models. Third, using the squared error loss function instead of the QLIKE loss function sheds a different light on the GARCH trees underperforming in the main analysis. However, this is most likely because the squared error loss is more sensitive to outliers. The QLIKE loss is therefore preferred over the squared error loss. Fourth, we test whether our results also hold for other stock indices. For the DAX and Nikkei, the GARCH tree outperforms the traditional GARCH models at the 1-day-ahead and 5-day-ahead horizons. For the FTSE, the differences are too small to be significant. Finally, at a lower observation frequency (weekly and monthly), do the GARCH trees not outperform the traditional models. Possibly because of the low amount of observations.

The structure of the paper is outlined as follows. Section 2 contains the methodology and explains the GARCH tree model in variance targeting form. Moreover, it discusses the various specifications of the GARCH tree used in the thesis and how their performance is evaluated. Section 3 provides an overview of the S&P500 dataset. It also contains information on the additional splitting variables used in the GARCH trees. Thereafter, Section 4 provides the estimation results, forecasting performance, and extensions of the GARCH tree models. Specifically, it shows the outcomes of the unconditional and conditional tests of Giacomini and White (2006), and the MCS procedure of Hansen et al. (2011). The extensions cover changes in the GARCH tree specifications and using a different dataset or loss function. A summary of the results, its shortcomings and suggestions for further research can be found in Section 5.

2 Methodology

2.1 GARCH Model & Variance Targeting

The original GARCH model of Bollerslev (1986) is a widely used volatility model for asset returns. Although the model is relatively simple, its forecasts are known to be hard to beat by other variations and alternatives (see Hansen and Lunde (2005)). In the GARCH model, today's volatility σ_t^2 is a function of yesterday's volatility σ_{t-1}^2 and yesterday's return r_{t-1}. Assuming the conditional mean is a nonzero constant, the GARCH model for the conditional volatility of the asset is:

$$\begin{aligned} r_t - \mu &= \epsilon_t = \sigma_t z_t \\ \sigma_t^2 &= \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2, \end{aligned} \tag{1}$$

where

$$\omega > 0, \quad \alpha \geq 0, \quad \beta \geq 0, \quad \text{and} \quad \alpha + \beta \leq 1.$$

Here μ is the mean return, z_t is the innovation, while ω, α, and β, are the parameters in the volatility equation. The parameter restrictions make sure volatility is positive and stationary. Volatility in this model is unobserved and can be obtained after estimating the parameters and using starting value σ_0. The parameter estimates are found assuming a particular distribution for the innovations z_t and maximizing the related likelihood function. The success of the GARCH model lies in its ability to capture the stylized facts of financial returns, which are a low autocorrelation in returns, a high autocorrelation in squared returns, and non-normality.

Although the model is straightforward, one of its drawbacks is the number of parameters required to estimate when the dimensionality increases. This is because every additional volatility equation needs three new parameters. As will be shown later, the GARCH tree model faces this problem. As a solution, Engle and Mezrich (1996) proposed the variance targeting estimation (VTE) method where the unconditional variance is computed first. The unconditional variance is defined by $\gamma = \frac{\omega}{(1-\alpha-\beta)} = \frac{\omega}{\kappa}$, which, using Francq et al. (2011), results in the following reparametrization

$$\begin{aligned} r_t - \mu &= \epsilon_t = \sigma_t z_t \\ \sigma_t^2 &= \kappa\gamma + \alpha \epsilon_{t-1}^2 + (1 - \kappa - \alpha)\sigma_{t-1}^2, \end{aligned} \tag{2}$$

with new constraints

$$\kappa > 0, \quad \gamma > 0, \quad \alpha \geq 0, \quad \text{and} \quad \kappa + \alpha \leq 1.$$

The unconditional variance γ is estimated by the sample variance $\frac{1}{T}\sum_{t=1}^{T} \epsilon_t^2$, before the κ and α parameters are calculated. Because in this case the sample variance requires demeaned returns, the mean return μ is also approximated by the sample mean in the first step of the estimation process. The volatility equation in this reparametrization has become a weighted average of the unconditional variance, the square of yesterday's demeaned return and yesterday's volatility together summing up to one. Note that once the unconditional variance is known, the volatility equation needs only two parameters. This reduces the dimension of the optimization of the likelihood function in the second step, a useful feature for the computationally intensive GARCH tree model.

Francq et al. (2011) have compared the VTE method with the standard quasi-maximum likelihood estimation (QMLE) method for GARCH models. Asymptotically, the QMLE provides smaller variance around its estimators and the VTE estimates can be imprecise. In finite samples, however, simulation experiments show that both estimation methods are equally accurate in terms of RMSE and their applications to stock market data lead to very similar parameter estimates.

2.2 GARCH Tree Model

Decision trees, or simply trees, are statistical learning models that divide the regressor space into several regions (James et al., 2013). These regions defined by a set of thresholds or splitting rules lead to specific values for the dependent variable(s). The different regions are usually referred to as the terminal nodes of the tree. The set of splitting rules, where each rule divides one dimension of the regressor space in two, can visually be represented as a tree and creates a nonlinear relation between the regressors and the dependent variable. Friedman et al. (2009) describe trees as the best "off-the-shelf" method for data mining because of its interpretability, invariance to data transformations, and internal variable selection. Moreover, the possibility to combine multiple trees is an additional advantage to obtain accurate predictions.

In Audrino and Bühlmann (2001), a tree is combined with a GARCH model resulting in a local GARCH model at each terminal node. Initially, a standard GARCH model as in (1) is fit to asset returns. Then, this GARCH model is split in two based on a splitting rule. For example, two GARCH models could arise where one is for negative returns and one for positive returns, each with its own set of parameters ω, α, and β. The mean μ is not node-specific. The choice of splitting variable and splitting value is such that, among all possible options, the resulting split leads to the largest increase in likelihood. One does not have to stop at two different GARCH specifications, these can again be split until a certain number of terminal nodes is reached. Because a fully grown tree is likely to overfit the data, the tree is pruned such that a subtree with less terminal nodes remains. Audrino and Bühlmann (2001) implement the pruning strategy by searching for the subtree with the lowest AIC statistic. The resulting GARCH tree model can then be used to forecast volatility. At each point in time the splitting rules and the current values of the splitting variables determine which terminal node is relevant and which parameters to use in the GARCH volatility equation. The recursion in (1) for volatility then predicts next periods volatility.

The GARCH tree implementation used here will now be discussed. We adjust the GARCH tree of Audrino and Bühlmann (2001) to the relevant VTE method of (2) to reduce the total amount of parameters in the model. Because the tree splits the regressor space in multiple regions, consider the partition

$$\mathcal{P} = \{\mathcal{R}_1, ..., \mathcal{R}_k\}, \quad \cup_{j=1}^{k} \mathcal{R}_j = \mathbb{R} \times \mathbb{R}^+, \quad \mathcal{R}_i \cap \mathcal{R}_j = \emptyset \quad (i \neq j) \tag{3}$$

Hence, the partition $\mathcal{P}$ consists of k terminal nodes $\mathcal{R}_j$. For each terminal node $\mathcal{R}_j$, a separate GARCH model is fitted resulting in the following GARCH tree model

$$r_t - \mu = \epsilon_t = \sigma_t z_t$$
$$\sigma_t^2 = \sum_{j=1}^{k} (\kappa_j \gamma + \alpha_j \epsilon_{t-1}^2 + (1 - \kappa_j - \alpha_j)\sigma_{t-1}^2) I_{[(\epsilon_{t-1}, \sigma_{t-1}^2) \in \mathcal{R}_j]} \tag{4}$$

The indicator function makes sure a separate GARCH specification is used at each terminal node, based on past demeaned return ϵ_{t-1} and past volatility σ_{t-1}^2. At each terminal node we have two parameters for the volatility equation instead of three if we were to use the standard GARCH parametrization. This saves us a total of $k-1$ parameters to estimate. Note that if $k = 1$ the GARCH tree is just a normal GARCH model in VTE form. To build the tree, we introduce a splitting value d_1 and splitting variable index $\iota_1 \in \{1, 2\}$ which partitions

$$\mathbb{R} \times \mathbb{R}^+ = \mathcal{R}_{left} \cup \mathcal{R}_{right} \tag{5}$$

where $\mathcal{R}_{left} = \{(\epsilon_{t-1}, \sigma_{t-1}^2) \in \mathbb{R} \times \mathbb{R}^+; (\epsilon_{t-1}, \sigma_{t-1}^2)_{\iota_1} \leq d_1\}$ and $\mathcal{R}_{right}$ follows similarly with the relation '$>$' instead. Hence, we split the space of the splitting variables in two, using either yesterday's demeaned return or yesterday's volatility where $\mathcal{R}_{left}$ and $\mathcal{R}_{right}$ have their own GARCH specification. The tree grows further by iteratively introducing new splitting values and splitting variable indices (d_m, ι_m) to expand the partition $\mathcal{P}^{(old)} = \cup_j \mathcal{R}_j$ into $\mathcal{P}^{(new)} = \cup_{j \neq j^*} \mathcal{R}_j \cup (\mathcal{R}_{j^*,left} \cup \mathcal{R}_{j^*,right})$ at every iteration. The final partition $\mathcal{P} = \{\mathcal{R}_1, ..., \mathcal{R}_k\}$ is a GARCH tree with k terminal nodes.

To estimate the splitting rules and parameter estimates of the GARCH tree a stagewise procedure is followed. Algorithm 1 provides a detailed description how to grow the GARCH tree. Initially, the mean μ and unconditional variance γ are estimated by the sample mean and variance. After setting up a GARCH model for the entire dataset, additional terminal nodes are iteratively introduced based on the maximum gain in the log-likelihood. The reduced log-likelihood approach to measure this gain relieves the computational burden of the nonlinear optimization problem, because otherwise the parameters of the terminal nodes not being split have to be reestimated at the same time. This would increase the dimension of the likelihood optimization and thereby increase the computation time. Moreover, using the starting values $\hat{\theta}^{(m-1)\setminus *}$ and $\hat{\theta}^*$ to get $\hat{\theta}^m$ provides an additional shortcut to obtain sensible estimates of the GARCH tree parameters. The search for the best splitting variable and splitting value is done on a grid. Like Audrino and Bühlmann (2001), the grid consists of the empirical α-quantiles of each splitting variable, with $\alpha = i/\text{mesh}$, $i = 1, ...,\text{mesh}-1$. Here, mesh $= 8$.

The final stage of the GARCH tree estimation process is pruning. To prevent the tree from overfitting to the data, the tree built with Algorithm 1 is pruned back. The pruning process is done by going back from $\mathcal{P}^{(M)}$ to $\mathcal{P}^{(1)}$, evaluating all possible subtrees in between. Denote subtree i as $\mathcal{P}_i$. The best subtree is the one with the lowest AIC statistic, where

$$AIC^{\mathcal{P}_i} = -2l(\hat{\theta}^{\mathcal{P}_i}; r_2^T) + 2(\dim(\hat{\theta}^{\mathcal{P}_i}) + 2) \tag{6}$$

is the AIC statistic of subtree $\mathcal{P}_i$. Starting values for $\hat{\theta}^{\mathcal{P}_i}$ are averages of the relevant nodes below $\mathcal{P}_i$. Let $\hat{\mathcal{P}}$ denote the partiton of the optimal subtree, then the final GARCH tree model is as in (4) with partition $\hat{\mathcal{P}}$ and estimates $\hat{\theta}^{\hat{\mathcal{P}}}$.

Algorithm 1 Growing the GARCH tree model

Given return data $r_1, ..., r_T$ and choice of number of terminal nodes M;
for $m \leftarrow 1$, M **do**
if $m = 1$ **then**
Set $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} r_t$, $\hat{\gamma} = \frac{1}{T}\sum_{t=1}^{T}(r_t - \hat{\mu})^2$, and obtain $\hat{\theta}^{(1)}$ of the GARCH(1,1) model, as given in (2), by maximizing its log-likelihood

$$l(\theta; r_2^T) = \sum_{t=2}^{T} \log(f(r_t|\mathcal{I}_{t-1}, \hat{\mu}, \hat{\gamma}; \theta)),$$

where $\theta^{(1)} = (\kappa_1, \alpha_1)$ and $\mathcal{I}_{t-1}$ is the information set at time $t-1$. Denote this initial partition by $\mathcal{P}^{(1)} = \mathbb{R} \times \mathbb{R}^+$.
else
Given $\mathcal{P}^{(m-1)} = \{\mathcal{R}_1, ..., \mathcal{R}_{m-1}\}$, search for the splitting variable and splitting value that lead to the optimal new partition $\mathcal{P}^{(m)}$. This is done by:
I Consider a splitting rule (d_m, ι_m) that divides the corresponding terminal node $\mathcal{R}_{j^*}$ into $\mathcal{R}_{j^*} = \mathcal{R}_{j^*,left} \cup \mathcal{R}_{j^*,right}$. To facilitate the optimization of the terminal node split, the volatility equation of (4) is now

$$\begin{aligned}\sigma_t^2 = &\sum_{j \neq j^*} (\kappa_j \hat{\gamma} + \alpha_j \epsilon_{t-1}^2 + (1 - \kappa_j - \alpha_j)\sigma_{t-1}^2) I_{[(\epsilon_{t-1}, \sigma_{t-1}^2) \in \mathcal{R}_j]} \\ &+ \sum_{i \in \{j^*_{left}, j^*_{right}\}} (\kappa_i \hat{\gamma} + \alpha_i \epsilon_{t-1}^2 + (1 - \kappa_i - \alpha_i)\sigma_{t-1}^2) I_{[(\epsilon_{t-1}, \sigma_{t-1}^2) \in \mathcal{R}_i]}\end{aligned} \tag{7}$$

Let $\theta^* = \{\kappa_i, \alpha_i; i \in \{j^*_{left}, j^*_{right}\}$ and $\theta^{(m-1)\backslash *} = \{\kappa_j, \alpha_j; j = 1, ..., m-1, j \neq j^*\}$ summarize the volatility parameters of the terminal node currently being split and those that are not.
II Combine the GARCH tree model of (4) with (7) and obtain $\hat{\theta}^*$ by maximizing the reduced log-likelihood

$$l((\hat{\theta}^{(m-1)\backslash *}, \theta^*); r_2^T) = \sum_{t=2}^{T} \log(f(r_t|\mathcal{I}_{t-1}, \hat{\mu}, \hat{\gamma}, \hat{\theta}^{(m-1)\backslash *}; \theta^*)$$

Use the components of $\hat{\theta}^{(m-1)}$ of $\mathcal{R}_{j^*}$ as starting values for the new terminal nodes $\mathcal{R}_{j^*,left}$ and $\mathcal{R}_{j^*,right}$ to estimate θ^*.
III Optimize the reduced log-likelihood of II) by using different splitting rules in I) and II). Save the optimal $\hat{\theta}^*$ and splitting rule (d_m, ι_m).

Estimate the GARCH tree model of (4) by maximum likelihood for the new optimal partition $\mathcal{P}^{(m)}$. With $\hat{\theta}^{(m-1)\backslash *}$ and $\hat{\theta}^*$ as starting values, obtain $\hat{\theta}^m$, where $\theta^m = \{\kappa_j, \alpha_j; j = 1, ..., m\}$.
end if
end for

2.3 Tree Specifications

Section 2.2 has described the general procedure to estimate the GARCH tree model. Here, we will discuss the various implementations of the GARCH and GARCH tree model to evaluate the performance of the GARCH tree method. Comparing multiple specifications and extensions of the vanilla GARCH and GARCH tree models should provide more robust evidence whether the GARCH tree method is able to outperform the standard model. This is also because the GARCH tree is more flexible, allowing for more complicated models.

A universal characteristic of all GARCH tree models is the maximum number of terminal nodes M that determines the size of the trees before pruning. As suggested by Audrino and Bühlmann (2001), $M = 6$. Their empirical applications to the German DAX stock index and the BMW stock price have no more than 5 terminal nodes after pruning. Allowing for a possibly larger tree also increases the computational time disproportionately for every additional terminal node. As baseline versions of both models, the GARCH and GARCH tree models will both be estimated assuming a standard normal distribution for the innovations z_t, hence $z_t \sim N(0,1)$. The density of returns is then

$$f(r_t|\mathcal{I}_{t-1};\mu,\theta) = \frac{1}{\sqrt{2\pi\sigma_t^2}} e^{\frac{1}{2}\frac{(r_t-\mu)^2}{\sigma_t^2}} \tag{8}$$

The normal distribution is still often used because of its QMLE properties (Fan et al., 2014). The estimates are consistent but lose efficiency if the true distribution is not normal. Note that if the innovation distribution is normal the return distribution is not. Still, using heavier tailed distributions could provide a better model fit and more accurate forecasts. A simulation study of Audrino and Bühlmann (2001) shows that under a normal distribution the GARCH tree could select too many terminal nodes, overspecifying the true model. Furthermore, Wilhelmsson (2006) shows that a GARCH model with t-distributed innovations outperforms one with normal innovations for the S&P500. Therefore, I also estimate the GARCH and GARCH tree model with standardized t-distributed innovations, thus $z_t \sim t(0,1,\mathrm{v})$, where v represents the degrees of freedom (Bollerslev, 1987). In this case the density equals

$$f(r_t|\mathcal{I}_{t-1};\mu,\theta) = \frac{\Gamma(\frac{\mathrm{v}+1}{2})}{\Gamma(\frac{\mathrm{v}}{2})} \frac{1}{\sqrt{\pi(\mathrm{v}-2)\sigma_t^2}} \Big(1 + \frac{(r_t-\mu)^2}{(\mathrm{v}-2)\sigma_t^2}\Big)^{-\frac{\mathrm{v}+1}{2}}, \qquad \mathrm{v} > 2. \tag{9}$$

Recently, Oh and Patton (2021) have shown that GARCH forecasts can be improved by using two exogenous volatility measures, namely realized variance (RV) and the VIX index. RV measures volatility by using the variation in intraday price changes, while the VIX is an option-implied index of stock market volatility. In Oh and Patton (2021), GARCH parameters were estimated locally based on the similarity of current RV or VIX values with their past. The resulting forecasts performed significantly better. It seems likely that including these two variables as splitting variables in the GARCH tree will improve forecasts as well. Besides RV and the VIX, we add three economic indices to the GARCH tree. Schwert (1989), Engle et al. (2013) and Amendola et al. (2019) have documented, among others, a relation between volatility and macroeconomic variables. Although these effects are usually found at a low frequency, the three economic indices can be applied to daily observations. The economic indices are the

daily Economic Policy Uncertainty Index of Baker et al. (2016), the weekly National Financial Conditions Index of Brave and Kelley (2017), and the daily Aruoba-Diebold-Scotti Business Conditions Index of Aruoba et al. (2009) (see Section 3 for more details). All three indices capture US economy-wide information and could add more forecasting power to the GARCH tree. The weekly NFCI is turned to a daily index by using the last known weekly observation to fill the daily observations. To sum up, two GARCH trees are added to the comparison with RV, the VIX, and the economic indices as additional splitting variables, either with normally or t-distributed innovations.

2.4 GARCH Random Forest

As an extension to the GARCH tree specifications of Section 2.3, we will combine a set of GARCH trees to make predictions. A single tree has low bias but high variance and the idea of using multiple trees is to reduce the variance while keeping bias low (Friedman et al., 2009). This is done by averaging the forecasts of the individual trees to create the actual prediction. A random forest is a collection of de-correlated trees where each tree is estimated over a different bootstrap sample. The trees are de-correlated to further reduce variance. The de-correlation is accomplished by limiting the choice of splitting variables at each split and in each tree. Instead of searching for the best splitting variable among all candidates, the choice is limited to only a few resulting in a wider variety of less correlated trees. The amount of candidates m is usually set according to some rule. Here we use $m = p/3$ of Friedman et al. (2009), where p is the total amount of splitting variables. It is common to not prune the individual trees because the problem of overfitting for a single tree is averaged out, hence each tree will have six terminal nodes. For the same reason the innovation distribution in the random forest will be standard normal. The GARCH random forest is made of GARCH trees with all additional splitting variables mentioned in Section 2.3. The total amount of splitting variables in the random forest is thus 7 which leads to $7/3 \approx 2$ splitting candidates at each split. The number of trees in the forest is set to 100, where each tree is estimated with a bootstrap sample generated by the stationary bootstrap method of Section 2.5.

2.5 Stationary Bootstrap

The bootstrap samples are created with the stationary bootstrap method of Politis and Romano (1994). This bootstrap method is able to generate stationary pseudo time-series datasets by resampling the original data. Opposite to block bootstrap methods which build time series by stacking blocks of observations of fixed length, the stationary bootstrap selects blocks of random length depending on a predetermined probability p. Given original data $X_1, ...X_N$ of size N, a bootstrap sample $X_1^*, ...X_N^*$ is generated according to the following rule: if X_i^* is based on X_j then X_{i+1}^* is equal to X_{j+1} with probability $1 - p$ or chosen randomly from all N observations with probability p. Because the block length follows a geometric distribution in the stationary bootstrap, the specification of the bootstrap is sometimes expressed by the expected block length w, where $w = \frac{1}{p}$. Similar to Oh and Patton (2021), the expected block length is set to 10, such that $p = 0.1$. For other papers using the stationary bootstrap in a GARCH setting, see for example Hansen and Lunde (2005), Rapach and Strauss (2008), Lee and Long (2009), and Kim

et al. (2016).

2.6 GJR-GARCH

The GARCH tree and random forest are thus far only competing with the standard GARCH model. As the conditional variance of financial returns reacts differently to positive and negative returns, it is likely that the GARCH tree will do better than standard GARCH, simply because the GARCH tree is able to capture the leverage effect. Therefore, we include the GJR-GARCH model of Glosten et al. (1993) in the comparison. The parametrization of the GJR-GARCH model in VTE form is

$$\begin{aligned} r_t - \mu = \epsilon_t &= \sigma_t z_t \\ \sigma_t^2 &= \kappa\gamma + \alpha\epsilon_{t-1}^2 + \phi\epsilon_{t-1}^2 I_{(\epsilon_{t-1}<0)} + (1 - \kappa - \alpha - \frac{\phi}{2})\sigma_{t-1}^2, \end{aligned} \tag{10}$$

with constraints

$$\kappa > 0, \quad \gamma > 0, \quad \alpha \geq 0, \quad \phi \geq 0, \quad \text{and} \quad \kappa + \alpha + \frac{\phi}{2} \leq 1.$$

The additional term for ϵ_{t-1}^2 captures the different reaction of volatility to negative returns. Although the extension is simple, Engle and Ng (1993) have shown that the GJR-GARCH model is the best parametric model to measure the leverage effect. If the GJR-GARCH model performs similar to the GARCH tree this could indicate that the GARCH tree is an overspecified/more difficult than necessary model for volatility. Because the our goal is not to find the best GJR-GARCH model, the model is only estimated with normally distributed innovations.

2.7 GARCH Tree Forecasting

2.7.1 Design

For the forecasting exercise we have S&P500 return data from 4/1/2000 until 12/11/2021, a total of $T = 5480$ observations. This dataset is divided into an estimation sample running from 2000-2010 (2755 observations) and an out-of-sample period from 2011 until 2021 (2725 observations). For simplicity, all models are estimated only once based on the estimation sample. In the forecasting analysis we will compare the performance of eight different models. These are the standard GARCH, the GARCH tree, the GARCH tree with RV, VIX, and the economic indices as additional splitting variables, the GARCH random forest, and GJR-GARCH, where the latter two are only with normal innovations while the others are also estimated with t-distributed innovations. Their performance will be assessed by using 1-,5-, and 20-day forecast horizons. 1-day-ahead forecasts of all models are easily obtained because the volatility equations only require things known the day before. For $s \geq 2$, the s-step-ahead forecast of the standard GARCH in VTE form is

$$\begin{aligned} \hat{\sigma}_{t+s|t}^2 &= E(\kappa\gamma + \alpha\epsilon_{t+s-1}^2 + (1 - \kappa - \alpha)\sigma_{t+s-1}^2 | \mathcal{I}_t) \\ &= \kappa\gamma + \alpha\hat{\sigma}_{t+s-1|t}^2 + (1 - \kappa - \alpha)\hat{\sigma}_{t+s-1|t}^2 \\ &= \hat{\sigma}_{t+s-1|t}^2 + \kappa(\gamma - \hat{\sigma}_{t+s-1|t}^2), \end{aligned} \tag{11}$$

because $E(\epsilon^2_{t+s-1}|\mathcal{I}_t) = E(E(\epsilon^2_{t+s-1}|\mathcal{I}_{t+s-2})|\mathcal{I}_t) = E(\sigma^2_{t+s-1}|\mathcal{I}_t) = \hat{\sigma}^2_{t+s-1}$. This recursion is applied to get 5-day- and 20-day-ahead predictions. The size of κ controls the speed of forecasts reverting to the unconditional variance γ. The GARCH tree forecasts also rely on 11, but since the regressor space is split into k terminal nodes, and each terminal node j has its own set of parameters (κ_j, α_j), the recursion is now

$$\hat{\sigma}^2_{t+s|t} = \hat{\sigma}^2_{t+s-1|t} + \kappa_j(\gamma - \hat{\sigma}^2_{t+s-1|t}). \tag{12}$$

The GARCH tree specification of 4 actually requires the terminal node to forecasts volatility at time t to be based on the values of the regressor at time $t-1$. However, for multi-step-ahead forecasting these values of the regressor are unknown. We therefore forecast s-step-ahead volatility at time $t+s$ using the most recent time t terminal node. The forecasts of the GARCH random forest are obtained by averaging the forecasts of the individual trees. GJR-GARCH forecasts follow the same recursion as in 11.

The forecasts are evaluated against a proxy for the true conditional variance of daily returns, which I take as the daily RV. RV is an unbiased and, more importantly, a less noisy estimate of the true conditional variance than the squared return (Andersen et al., 2003). The prediction error measuring the accuracy of each model is calculated with the QLIKE loss function

$$QLIKE_t = \frac{RV_t}{\hat{\sigma}^2_t} - \log\Big(\frac{RV_t}{\hat{\sigma}^2_t}\Big) - 1, \tag{13}$$

which measures the loss based on standardized forecast errors (Oh & Patton, 2021). RV_t is the RV at time t and $\hat{\sigma}^2_t$ is the volatility prediction. Many different loss functions exist when evaluating volatility forecasts (see Hansen and Lunde (2005)). However, Patton (2011) shows that for certain loss functions, unlike the QLIKE loss function, forecasting the conditional variance is not optimal when minimizing the expected loss. Moreover, the QLIKE loss is less affected by extreme observations and has less variance than the squared error loss. An additional advantage of QLIKE loss is that the ranking of any two forecasts is invariant to scaling of the data. A possible downside is its heavier penalty on under-prediction, when the volatility prediction is less then the benchmark volatility, than on over-prediction (Patton, 2011).

2.7.2 Giacomini-White Tests

To assess the performance of the volatility models we will use the unconditional predictive ability test and the conditional predictive ability test of Giacomini and White (2006) (GW). The tests take into account the effect of estimation uncertainty and can be applied to both nested and nonnested models. The unconditional GW test compares two models and tests whether one performed significantly better over the out-of-sample period. The null hypothesis of equal unconditional predictive ability is rejected if the difference in the average loss is nonzero. Here, we will make use of the unconditional test by evaluating each model at each forecasting horizon against the standard GARCH model with normally distributed innovations to see if adding the more complicated tree structure is worth the effort.

For the unconditional test, let $\Delta\bar{L}_i$ denote the average QLIKE loss difference between model i and the standard GARCH benchmark. The unconditional predictive ability test then makes

use of the test statistic

$$t_i = \frac{\Delta \bar{L}_i}{\hat{\sigma}_n / \sqrt{n}}, \tag{14}$$

where $\hat{\sigma}_n$ is an estimate of the standard deviation of the difference in QLIKE losses and n is the number of out-of-sample predictions. For a given significance level α, the null hypothesis is rejected when $|t_i| > z_{\alpha/2}$, where $z_{\alpha/2}$ is the $(1 - \alpha/2)$ quantile of the standard normal distribution. As proposed by Giacomini and White (2006), $\hat{\sigma}^2 = n^{-1} \sum_{i=1}^{n} \Delta L_i^2 + 2n^{-1} \sum_{j=1}^{p} \sum_{i=1+j}^{n} \Delta L_i \Delta L_{i-j}$ is used as a HAC estimator of the variance, where p is the lag length. The lag length will be set to ten, similar to Oh and Patton (2021).

The conditional test of equal predicitve ability is also a pairwise test for forecasting performance but unlike the unconditional test makes use of available information to predict when one method provides more accurate forecasts than the other. This available information is stored in a so called test function h_t, a q x 1 vector containing the q variables to make the performance distinction. The test statistic for model i and j is equal to

$$T_{ij} = n\Big(n^{-1} \sum_{t=m}^{T-\tau} h_t \Delta L_{ij,t+\tau}\Big)' \Omega^{-1} \Big(n^{-1} \sum_{t=m}^{T-\tau} h_t \Delta L_{ij,t+\tau}\Big), \tag{15}$$

where $\Delta L_{ij,t+\tau}$ is the QLIKE loss difference between model i and j at time $t+\tau$ with τ being the forecast horizon, m the last observation of the estimation sample, and T the total sample size. Moreover, $\Omega = n^{-1} \sum_{t=m}^{T-\tau} (\Delta L_{ij,t+\tau})^2 h_t h_t' + 2n^{-1} \sum_{j=1}^{\tau-1} \sum_{t=m+j}^{T-\tau} (\Delta L_{ij,t+\tau} L_{ij,t+\tau-j} h_t h_t')$ estimates the covariance matrix with the last term dropping out whenever $\tau = 1$. The null hypothesis of equal conditional predictive ability, $E(h_t \Delta L_{ij,t+\tau}) = 0$, is rejected when $T_{ij} > \chi^2_{q,1-\alpha}$, where $\chi^2_{q,1-\alpha}$ is the $(1 - \alpha)$ quantile of the χ^2_q distribution. The null means the variables in the test function cannot distinguish between the forecasting performance of the two models. In our case, $h_t = (1, \Delta L_{ij,t}, RV_t)'$ contains a constant and last known loss difference and RV. It could be that when model i has a smaller loss at time t than model j, it will also have a lower loss at time $t+\tau$. More importantly, model i could perform better when RV is low or high hence adding RV to the test function also takes into account whether the performance difference is volatility-dependent. The pairwise conditional predictive ability tests will be done for each possible pair of models and for each forecast horizon. The significance level of the unconditional and conditional test will be set at 5%. If the test is rejected this means that the test function h_t can predict the QLIKE loss differences out of sample. To indicate the size of the relative outperformance, Giacomini and White (2006) suggested computing the proportion of times model j is chosen for prediction instead of model i. This is done by calculating $I_{ij} = n^{-1} \sum_{t=m}^{T-\tau} \mathrm{I}(\delta' h_t > 0)$, where δ stores the regression coefficients of a regression of the loss differences $\Delta L_{ij,t+\tau}$ on h_t. If $\delta' h_t > 0$, then model i has a larger expected loss and model j should be chosen.

2.7.3 Model Confidence Set

Finally, to test for any significant differences in performance between the standard GARCH and GARCH tree models, we apply the model confidence set (MCS) procedure of Hansen et al. (2011). The MCS is a set of models that contains the best models with a certain confidence. Given a collection of models, the MCS is obtained by comparing the relative performance of

each model and sequentially dropping models from the set with a relatively bad performance. It does so until the null hypothesis of equal performance among the models in the set cannot be rejected. Here, the performance of each model is measured by the time series of QLIKE losses. Denote $L_{i,t}$ as the QLIKE loss of model i at time t, then

$$d_{ij,t} = L_{i,t} - L_{j,t} \quad \text{for all} \quad i, j \in M^0 \tag{16}$$

defines the relative performance variables of the starting set of models M^0. The superior set of models is defined by $M^* \equiv \{i \in M^0 : E(d_{ij}) \leq 0 \text{ for all } j \in M^0\}$. The MCS procedure tries to find this superior set M^*. To test the hypothesis whether $E(d_{ij}) = 0$, we construct the test statistic $T_R = max|t_{ij}|$, where

$$t_{ij} = \frac{\bar{d}_{ij}}{\sqrt{var(\bar{d}_{ij})}}, \tag{17}$$

with $\bar{d}_{ij} = T^{-1} \sum_t^T d_{ij,t}$. Hence, T_R is the largest standardized loss differential and the model responsible for this will be removed from the set if the test statistic is too large. The distribution of the test statistic T_R is estimated with the stationary bootstrap method of Section 2.5 using 5000 bootstrap replications. In the results, the MCS p-values of the models, denoted by p_{MCS}, will be reported. These p-values are equal to the maximum p-value of the deleted models and the current worst model in the sequential testing procedure. This makes it easier to see whether a certain model is in the superior model set. Here, we will set the significance level at 5%. The MCS procedure is implemented using Sheppard (2018).

3 Data

The data consists of S&P500 closing prices, two measures of S&P500 volatility, and three economic indices. The S&P500 volatility measures are the 5-minute RV and VIX. The closing prices and RV are taken from the Oxford-Man Realized Library, while the VIX data comes from Chicago Board Options Exchange (CBOE) website[1]. RV measures volatility by using the variation in intraday price changes, while the VIX is an option-implied index of stock market volatility. Both datasets are on a daily basis and range from 3/1/2000 until 12/11/2021.

The economic indices are the Economic Policy Uncertainty (EPU) Index of Baker et al. (2016), the weekly National Financial Conditions (NFC) Index of Brave and Kelley (2017), and the daily Aruoba-Diebold-Scotti Business Conditions (ADS) Index of Aruoba et al. (2009). The EPU and NFC indices data are taken from the Federal Reserve Bank of St. Louis and the ADS index is from the Federal Reserve Bank of Philadelphia[2] The NFC index has a weekly frequency, but is turned to a daily series to accommodate with the rest of the data. The ADS index tracks

[1]See https://realized.oxford-man.ox.ac.uk/data, "Data", and https://www.cboe.com/tradable_products/vix/vix_historical_data/, "VIX data from 2004 to present", accessed: 15-11-2021.

[2]See https://fred.stlouisfed.org/series/USEPUINDXD/, "Economic Policy Uncertainty Index for United States",https://fred.stlouisfed.org/series/NFCI, "Chicago Fed National Financial Conditions Index",https://www.philadelphiafed.org/surveys-and-data/real-time-data-research/ads, "Aruoba-Diebold-Scotti Business Conditions Index", accessed 18-01-2022.

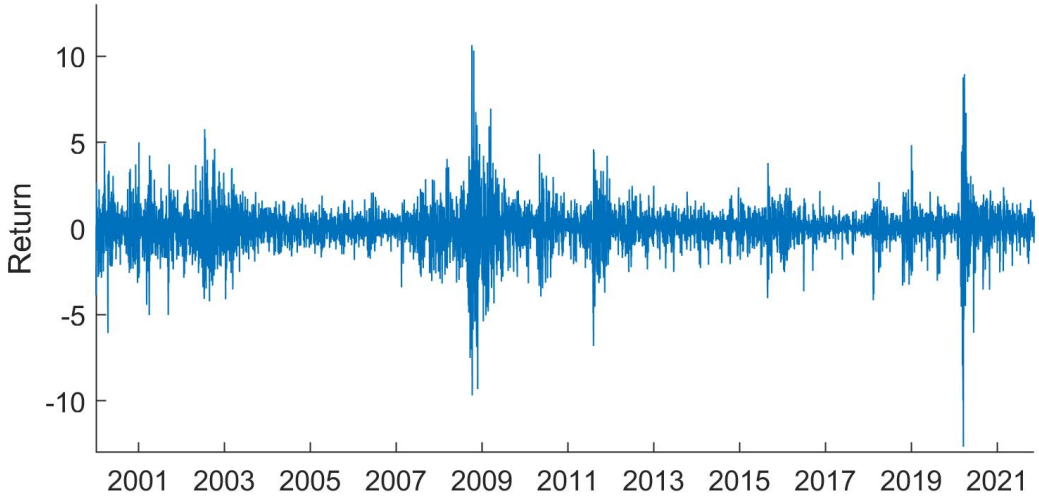

Figure 1: Daily S&P500 returns, 4/1/2000-12/11/2021.

real business activity by combining several macro variables like real GDP and industrial production, the NFC represents the financial conditions in money, debt and equity markets, while the EPU index follows activity of economic policy related news in newspapers.

Because at certain dates either the values for the closing price and RV or the VIX were missing, 22 observations are removed from the dataset. This leaves us with a total of 5481 observations. The S&P500 closing prices are converted to log-returns $r_t = 100 * \log(\frac{P_t}{P_{t-1}})$, where P_t denotes the S&P500 closing price at time t. Figure 1 shows these returns over time. The Financial Crisis and recent COVID-19 pandemic have led to very large returns of more than 10 percent. Compared to most daily returns, which range between $\pm$ 4 percent, these episodes provide very extreme observations. This is outlined visually in Figure 2. Figure 2 plots the distribution of S&P500 returns and a fitted normal density function. Clearly, the returns do not seem to be normally distributed. The return distribution is more peaked and has heavier tails. Table I provides summary statistics of the data. The kurtosis value of this sample is 13.76, much more than 3, the kurtosis of a normal distribution. Moreover, the returns are left-skewed i.e.,

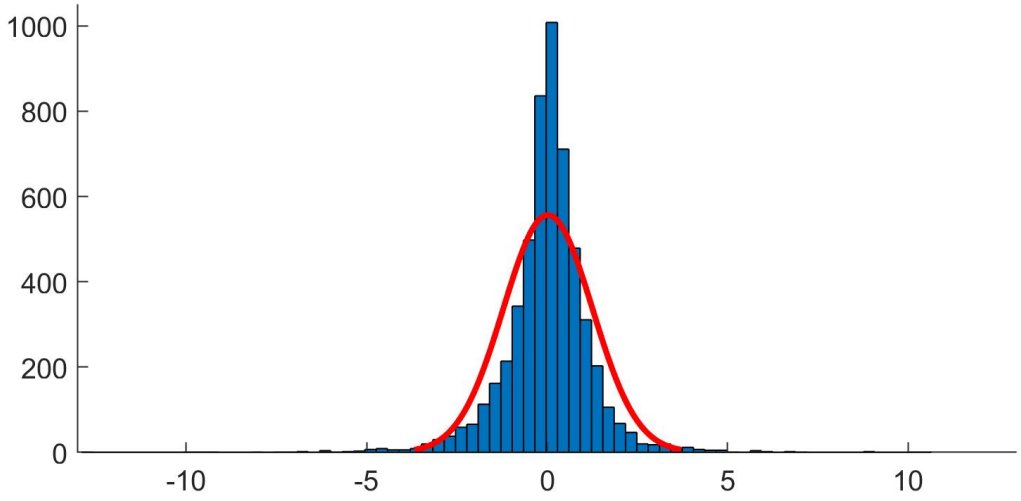

Figure 2: Histogram and normal density of S&P500 returns, 4/1/2000-12/11/2021.

Table I
Descriptive Statistics of S&P500 Data

This table gives an overview of the dataset. The S&P500 return data is daily from 4/1/2000 until 12/11/2021. The last three columns report $\rho(\tau)$, the sample autocorrelations at displacement τ.

	Mean	St.dev.	Min	Max	Skew.	Kurt.	JB value	JB p-value
S&P500	0.021	1.234	−12.670	10.642	−0.392	13.765	26599.743	0.000

positive returns are more common than negative returns. In fact, 54% of all returns is positive. A Jarque-Bera test of normality consequently rejects the null hypothesis that the returns follow a normal distribution (Jarque & Bera, 1980).

The sign of the returns in Figure 1 follows a random sequence. However, the size of the returns seem to slowly change over time. Periods of small returns are followed by a period of relatively large returns. The pre-Financial Crisis period 2004-2007 and the volatile crisis period lasting until 2012 is a good example. This phenomenon is usually referred to as volatility clustering. Figure 3 shows the empirical autocorrelation functions of the returns and the squared returns. Squared returns are a simple measure to approximate volatility. While the autocorrelation function of returns is stable around zero, the autocorrelation function of squared returns is slowly declining. Hence, returns are barely related to past returns but return volatility is partially determined by itself. This explains the success of volatility models like GARCH.

The VIX index obtained from the CBOE website is on a different scale than the 5-minute RV of the Oxford-Man Realized Library. The VIX is an annualized volatility index where returns are measured in units and volatility is defined as the standard deviation of prices, while RV is a daily measure of the variance based on returns in decimals. Therefore, using Buncic and Gisler (2016), the VIX is rescaled by $\text{VIX}^2/252$ and RV by 100^2.

Figure 4 plots the square root of three different measures of S&P500 volatility, namely, the squared return, RV, and the VIX. The square root was taken to smooth the figure such that volatility is measured as the standard deviation of returns. It is clear from the graph that the squared return is the most volatile measure of volatility. Andersen and Bollerslev (1998) has shown that the squared return is a bad proxy for the conditional variance. Although being an unbiased estimate of the conditional variance, the idiosyncratic error in returns makes

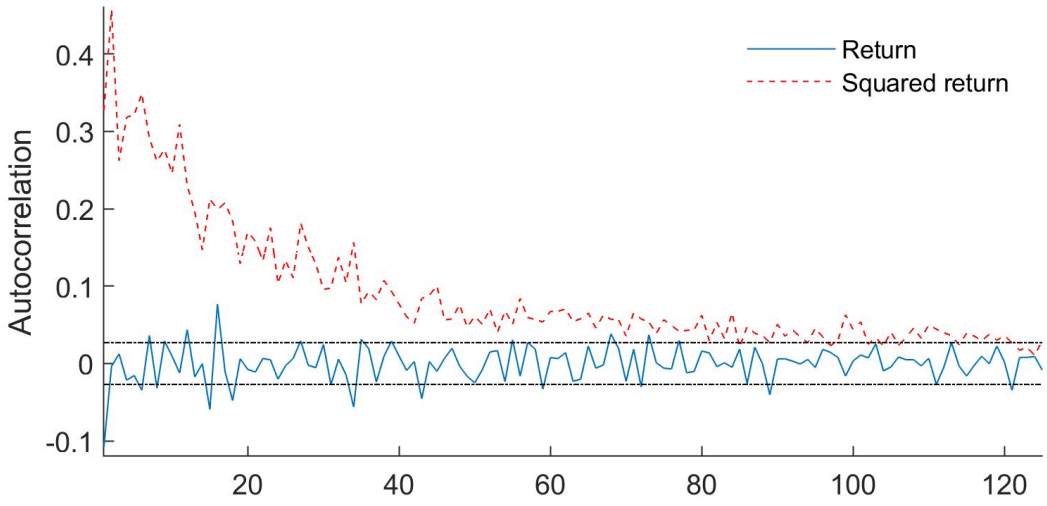

Figure 3: Empirical autocorrelation functions of S&P500 returns, 4/1/2000-12/11/2021.

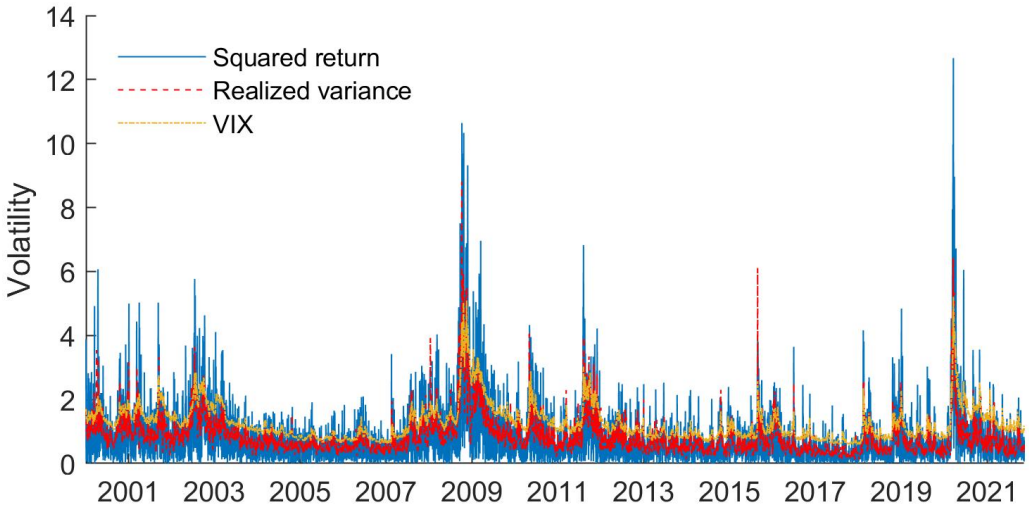

Figure 4: Square root of S&P500 variance measures, 4/1/2000-12/11/2021.

this proxy very noisy. The realized variance is also unbiased but much less noisy because of its use of intraday data. Figure 4 provides visual evidence of these findings.

The time series of the economic indices are plotted in Figure 5. The EPU is at its highest around 9/11, the Financial Crisis and the COVID-19 pandemic. Baker et al. (2016) has already found an association between the EPU and greater stock price volatility at the firm level. After controlling for the VIX, the EPU index adds explanatory power to the volatility of firms with high government exposure. The ADS and NFC indices have zero mean by construction. For the ADS index, positive values indicate better-than-average business conditions, while negative values indicate worse-than-average conditions. For the NFC index, positive values indicate tighter than average financial conditions and negative values indicate looser than average financial conditions. This index has been mostly negative during the sample period. The ADS and NFC index are updated in real time meaning past values change whenever new data is released. The changes tend to be very small, however.

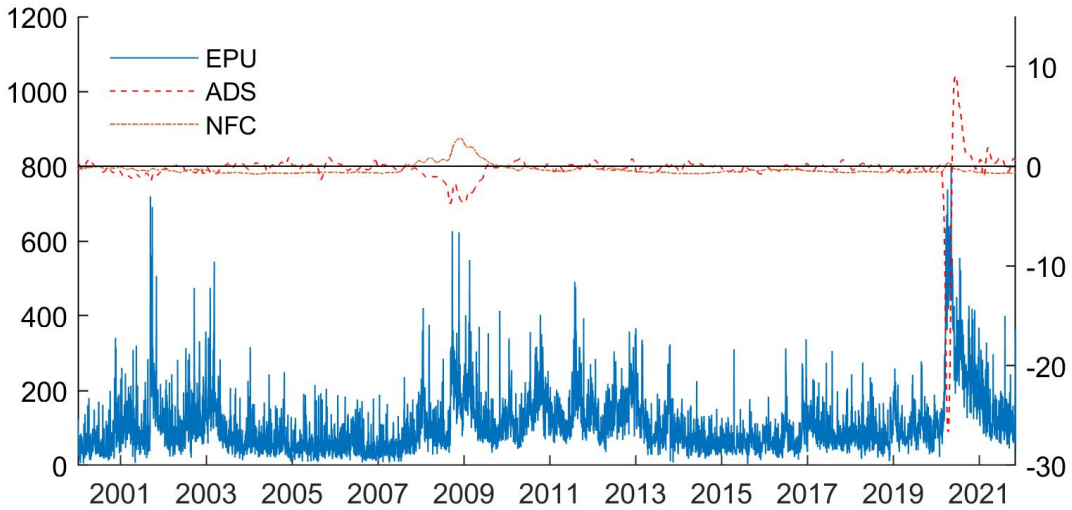

Figure 5: Economic Policy Uncertainty (EPU) (left axis), Aruoba-Diebold-Scotti Business Conditions (ADS) (right axis) and National Financial Conditions (NFC) (right axis) index, 4/1/2000-12/11/2021.

Table II
Computation Time

Comparison of the computation time of growing and pruning the GARCH tree with normally distributed innovations using either QMLE or VTE. The reported time is an average of five runs over S&P500 data from 4/1/2000 until 31/12/2010 (2755 observations).

	Time (sec)
GARCH - N QMLE	94.28
GARCH - N VTE	27.8

4 Results

4.1 Model Estimates

This section gives an overview of the estimation results. Before going through the actual estimates, we will first discuss the advantage of using the VTE method. Table II shows the computation time of estimating and pruning a GARCH tree with normally distributed innovations, either with QMLE as in (1) or with VTE as in (2). Remember from Section 2.3 that the tree first grows to 6 terminal nodes, which means a total of 9 unique subtrees has to be evaluated afterwards while pruning. With QMLE the computation time is around one and a half minutes, while with VTE the computation time is about 30 seconds. Hence, the computation time is reduced to about a third. This gain is not only practically useful for this tree but also indicates the computational gain when estimating the GARCH trees with the additional RV, VIX, and economic splitting variables and the GARCH random forest[3]. Francq et al. (2011) compared the total computation time of estimating a standard GARCH model for 11 equity indices and came up with a reduction of 40%.

The estimation results are reported in Table III[4]. It shows the parameter estimates and log-likelihood of all models mentioned in Section 2.7.1, except for the random forest. The results for the traditional GARCH models are in Panel A, Panel B contains the results for the GARCH trees. The parameter estimates for the conditional mean μ and unconditional variance γ are the same for all models since these are calculated before estimating the volatility equations. The estimates of GARCH with normally and t-distributed innovations are very similar with κ being very close to zero meaning not much weight is given to the unconditional variance in the volatility equation. Volatility does not revert back quickly to its unconditional mean, which is why we have volatility clustering. In the GJR-GARCH model, the α parameter controlling the reaction to past returns is set to zero and therefore volatility only reacts to negative returns. The slightly negative value for μ is unexpected but can be explained by the estimation sample which includes the Internet bubble and Financial Crisis.

Panel B of Table III shows the GARCH tree results with the node-specific values for κ and α for each terminal node $\mathcal{R}$. The first thing to notice is that all four GARCH tree models have six terminal nodes. This means that the AIC criterion was minimized when the trees were fully grown. The empirical applications of Audrino and Bühlmann (2001) had no more than 5 terminal nodes, which could possibly be because of a smaller sample size (1000 observations).

[3]The models are estimated in MATLAB with self-written code. Using any packages or other software might reduce the estimation time or the difference between the two estimation methods.

[4]As starting values for all models we set $\kappa = 0.05$, $\alpha = 0.1$, v = 4, and $\phi = 0.15$. All model constraints were implemented, for each node when applicable. An upper bound of 30 was set on v. Furthermore, $\sigma_1^2 = r_1^2$.

Table III
Estimation Results

This table reports estimation results for all models mentioned in Section 2. Panel A contains the results for the standard GARCH and GJR-GARCH model. Panel B contains the results for the GARCH trees. The models estimated with normally distributed innovations are denoted by N and those with t distributed innovations by t. The GJR-GARCH is only estimated with normally distributed innovations. The GARCH trees with +5 have RV, VIX, EPU, ADS and NFC as additional splitting variables. All models are estimated in VTE form with S&P500 data from 4/1/2000 until 31/12/2010. The table shows the parameter estimates, number of observations and corresponding log-likelihood. The tree terminal nodes $\mathcal{R}_1, ..., \mathcal{R}_6$ are shown graphically in Figures 6, 7, 8, and 9.

Panel A: GARCH			
	N	t	GJR
μ	-0.005	-0.005	-0.005
γ	1.873	1.873	1.873
κ	0.007	0.006	0.008
α	0.085	0.083	0.000
v		8.272	
ϕ			0.151
No. obs	2755	2755	2755
Log-likelihood	-4132.322	-4098.911	-4072.656

Panel B: GARCH Tree				
	Tree N	Tree t	Tree N+5	Tree t+5
μ	-0.005	-0.005	-0.005	-0.005
γ	1.873	1.873	1.873	1.873
$\mathcal{R}_1$ (κ/α)	(0.020/0.015)	(0.015/0.018)	(0.066/0.478)	(0.023/0.199)
$\mathcal{R}_2$ (κ/α)	(0.000/0.146)	(0.000/0.132)	(0.018/0.044)	(0.058/0.066)
$\mathcal{R}_3$ (κ/α)	(0.000/0.138)	(0.000/0.147)	(0.035/0.102)	(0.000/0.166)
$\mathcal{R}_4$ (κ/α)	(0.009/0.093)	(0.012/0.101)	(0.309/0.182)	(0.061/0.290)
$\mathcal{R}_5$ (κ/α)	(0.000/0.000)	(0.000/0.000)	(0.000/0.304)	(0.870/0.130)
$\mathcal{R}_6$ (κ/α)	(0.085/0.026)	(0.074/0.021)	(0.000/0.000)	(0.000/0.000)
v		10.071		11.102
No. obs	2755	2755	2755	2755
Log-likelihood	-4081.810	-4061.402	-4065.419	-4050.881

The terminal nodes $\mathcal{R}_1, ..., \mathcal{R}_6$ cannot be compared directly across the models because the splitting variables and their values might be different. To get a better understanding of the estimated GARCH trees, Figures 6, 7, 8, and 9 plot the way the trees are built. The first split of the GARCH tree with normally distributed innovations, which we will further denote as GARCH tree N, of Figure 6 tries to capture the asymmetry effect like the GJR-GARCH model by splitting the returns at 0.3. Further splitting rules seem to divide the regressor space into regions of high and low volatility for both negative and positive returns. To relate the terminal nodes to the parameter estimates, take for example node $\mathcal{R}_3$ and $\mathcal{R}_6$, where $\mathcal{R}_3$ contains large negative returns and high volatility and $\mathcal{R}_6$ large positive returns and high volatility. Node $\mathcal{R}_3$ has a large value for α of 0.138 while for node $\mathcal{R}_6$ the value is only 0.026. Volatility reacts differently to returns under different circumstances. Note also that in this case κ is zero for $\mathcal{R}_3$ and 0.085 for $\mathcal{R}_6$ meaning volatility returns faster to the unconditional variance when yesterday's return was positive rather than negative. Another interesting case is at node $\mathcal{R}_5$ where both κ and α are zero leading to no change in volatility at all. There, yesterday's return is positive and volatility is low. Meanwhile, Figure 7 plots the GARCH tree with t-distributed innovations

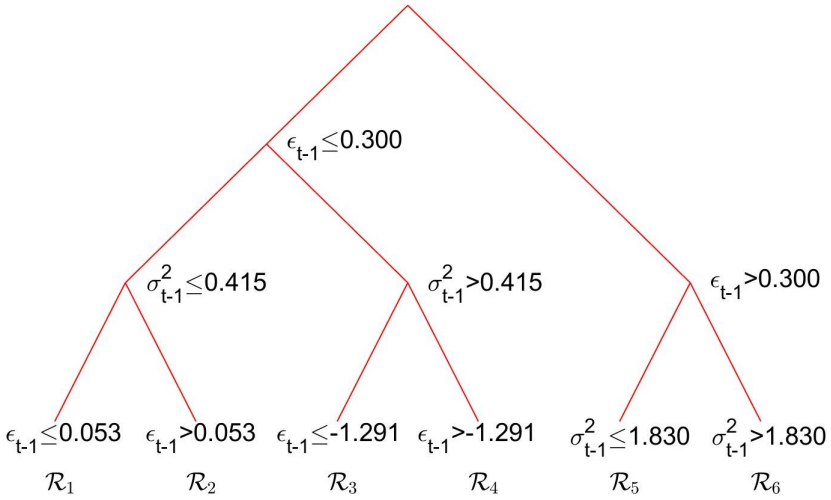

Figure 6: GARCH Tree N estimated partitions and splitting rules using S&P500 data from 4/1/2000 until 31/12/2010. The terminal nodes $\mathcal{R}_1, ..., \mathcal{R}_6$ match those of Table III.

(GARCH tree t). This tree is in terms of splitting rules almost identical to the GARCH tree N. Only the values of σ^2_{t-1} are slightly different because the parameter estimates of the GARCH tree t leads to another volatility recursion but the splitting values are at the same quantiles as for the GARCH tree N. The parameter estimates are also very close. The degrees of freedom of the t distribution is higher for the tree than the standard model.

Figures 8 and 9 plot the estimated GARCH trees with the additional splitting variables RV, VIX, EPU, ADS, and NFC, denoted GARCH tree N+5 and GARCH tree t+5 respectively. Both trees first split the regressor space at a return value of 0.3 and do not split the positive returns any further but focus on the different reactions of volatility to returns below 0.3. Furthermore, both trees use RV as a splitting variable and ignore the VIX. Oh and Patton (2021) found that RV is able to improve volatility forecasts more than the VIX, something the tree estimation procedure recognises as well. The GARCH tree N+5 makes use of all three economic indices, while the GARCH tree t+5 ignores the EPU index. Looking at the corresponding parameter estimates of Table III, the values of κ and α can be much higher than without the additional splitting variables. This is because the additional splitting variables allows the volatility equation

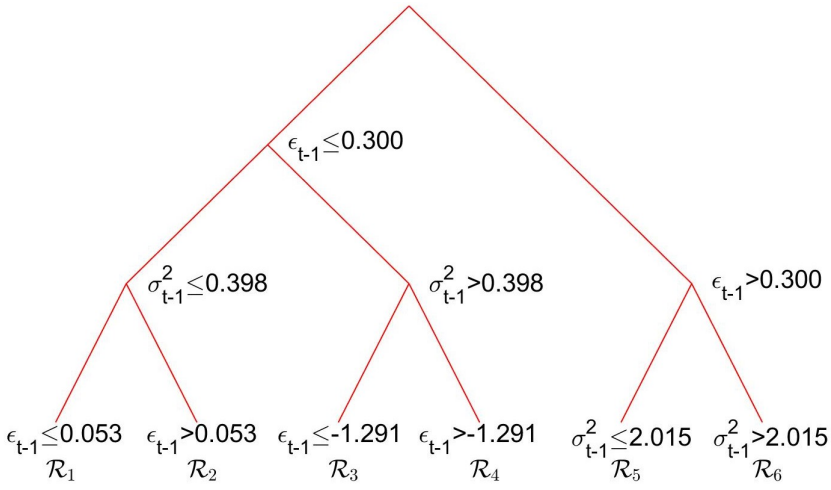

Figure 7: GARCH Tree t estimated partitions and splitting rules using S&P500 data from 4/1/2000 until 31/12/2010. The terminal nodes $\mathcal{R}_1, ..., \mathcal{R}_6$ match those of Table III

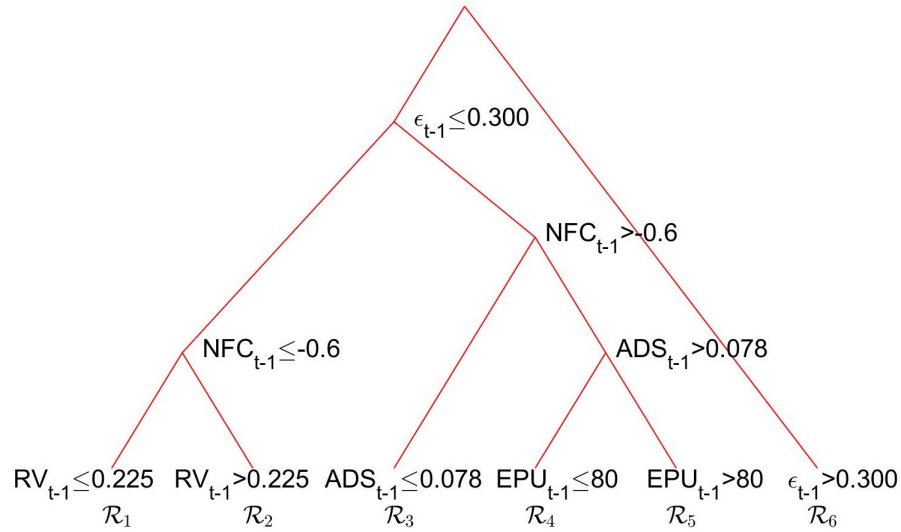

Figure 8: GARCH Tree N+5 estimated partitions and splitting rules using S&P500 data from 4/1/2000 until 31/12/2010. The terminal nodes $\mathcal{R}_1, ..., \mathcal{R}_6$ match those of Table III

to be more specific for certain periods in time. In the GARCH tree N+5 for example, $\mathcal{R}_3$ seems to correspond with $\mathcal{R}_3$ of the GARCH tree N; negative returns, tight financial conditions and below-average business conditions much like the negative returns and high volatility state of the GARCH tree N. However, the other side of the split where instead business conditions are above-average is much less common with very different parameters. Then the values of κ and α are either 0.309 and 0.18 or 0 and 0.3 depending on whether the EPU index is low or high. Similar arguments hold for the terminal nodes $\mathcal{R}_4$ and $\mathcal{R}_5$ of the GARCH tree t+5 targeting only 40 observations per node. As a result, the parameter estimates deviate strongly from the other terminal nodes.

To summarize, the estimation results between the two innovation distributions differ slightly. This hold at least for the traditional GARCH and raw GARCH tree models. Once we introduce more splitting variables the splitting rules start to deviate and potentially target small parts of the data. Moreover, the AIC criterion did not prune any of the GARCH trees.

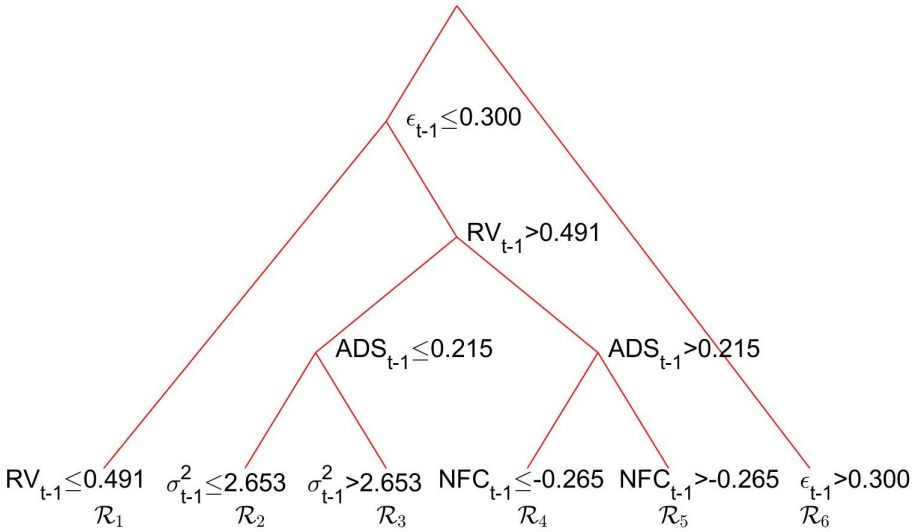

Figure 9: GARCH Tree t+5 estimated partitions and splitting rules using S&P500 data from 4/1/2000 until 31/12/2010. The terminal nodes $\mathcal{R}_1, ..., \mathcal{R}_6$ match those of Table III

4.2 Forecasting performance

We will now discuss the forecasting performance of the GARCH and GARCH tree models. These depend on the QLIKE losses as defined in (13) and are evaluated using the conditional and unconditional GW tests of Section 2.7.2 and the MCS procedure of Section 2.7.3. We have three forecast horizons: 1-day-ahead, 5-day-ahead, and 20-day-ahead. Table IV shows forecasting results for 1-day-ahead predictions. It contains the average and standard deviation of the QLIKE losses of every model, the t statistic of the unconditional GW test (GW stat) with the GARCH tree N as benchmark, and the p-values of the MCS procedure. Of the traditional GARCH models the GJR-GARCH model has the lowest loss. The tests of unconditional predictive ability with respect to the GARCH N model are rejected when $|t_i| > 1.96$. The fact that the test statistic is larger for GARCH t does not mean that GARCH t is better than GJR-GARCH, only that the loss difference had a larger estimated variance. The single GARCH tree models show mixed results. Without adding the additional splitting variables, the QLIKE losses are lower but the unconditional GW test is not rejected. If they are added, the GARCH tree t+5 performs well with a lower average loss than the GJR-GARCH model, while the GARCH tree N+5 does much worse, with significantly worse results than the GARCH N model. The fact that the GARCH trees without the additional variables perform almost identical is not surprising, Figures 6 and 7 show these trees are basically the same. The other two, however, do make different splitting rules, which in case of the GARCH tree N+5 is not for the better. Another disappointing result is the random forest. Initially, we only chose to estimate the random forest with normally distributed innovations but because its performance is dissatisfactory, a random forest with t-distributed innovations is also added. The random forests are the worst models in the table. Because the random forests make use of model averaging and de-correlated trees this is rather unexpected. For now, the performance of the GARCH trees is not necessarily better than the GARCH tree N. Only the GARCH tree t+5 does well, being the only model in the MCS.

Table IV
Forecasting Results - 1-day-ahead

This table reports the out-of-sample forecasting performance of all models based on 1-day-ahead predictions of S&P500 volatility. The competing models are the GARCH model, the GJR-GARCH model, the GARCH tree, and GARCH random forest. All models are estimated in VTE form. The models estimated with normally distributed innovations are denoted by N and those with t distributed innovations by t. The GJR-GARCH is only estimated with normally distributed innovations. The GARCH trees with +5 have RV, VIX, EPU, ADS, and NFC as additional splitting variables, which are also included in the GARCH random forests. The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 1/1/2011 until 12/11/2021. The table shows the average and standard deviation of the QLIKE losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	QLIKE (Std)	GW stat	p_{MCS}
N	0.430 (0.563)	*	0.000
t	0.422 (0.573)	-5.279	0.000
GJR	0.399 (0.523)	-3.623	0.021
Tree N	0.420 (0.546)	-1.148	0.001
Tree t	0.418 (0.560)	-1.243	0.004
Tree N+5	0.553 (1.331)	2.493	0.000
Tree t+5	0.380 (0.625)	-4.758	**1.000**
RF N	0.573 (0.494)	6.403	0.000
RF t	0.617 (0.523)	7.053	0.000

Table V

Conditional Predictive Ability Tests - 1-day-ahead

This table reports results of pairwise tests of equal conditional predictive ability for all models based on 1-day-ahead predictions of S&P500 volatility. The table shows the p-values of equal conditional predictive ability for the models in the corresponding row and column. The loss is the QLIKE loss and the test function is $h_t = (1, \Delta L_t, RV_t)'$. The numbers in parentheses are the proportion of times the model in the column outperforms the model in the row over the out-of-sample period using the decision rule of Section 2.7.2. A plus (minus) sign indicates the test of equal conditional predictive ability is rejected at the 5% level and that the model in the column (row) outperforms the model in the row (column) more than 50% of the time.

	N	t	GJR	Tree N	Tree t	Tree N+5	Tree t+5	RF N
t	0.000^-							
	(0.10)							
GJR	0.000^-	0.000^-						
	(0.12)	(0.18)						
Tree N	0.000^-	0.000^-	0.000^+					
	(0.39)	(0.49)	(0.72)					
Tree t	0.000^-	0.000^-	0.000^+	0.000^-				
	(0.36)	(0.45)	(0.67)	(0.38)				
Tree N+5	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+			
	(0.94)	(0.94)	(0.96)	(0.94)	(0.99)			
Tree t+5	0.000^-	0.000^-	0.000^-	0.000^-	0.000^-	0.000^-		
	(0.10)	(0.13)	(0.23)	(0.18)	(0.22)	(0.03)		
RF N	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+	
	(0.94)	(0.95)	(0.96)	(0.98)	(0.95)	(0.80)	(0.96)	
RF t	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+	0.000^+
	(0.94)	(0.94)	(0.95)	(0.97)	(0.94)	(0.89)	(0.95)	(0.78)

Table V shows the results of the pairwise conditional predictive ability test for all models using the 1-day-ahead QLIKE losses. The entries are the p-values of the tests with the proportions I_{ij} in parentheses. A plus (minus) sign indicates the test of equal conditional predictive ability is rejected at the 5% level and that the model in the column (row) outperforms the model in the row (column) more than 50% of the time. In all cases the null hypothesis of equal conditional predictive ability is rejected. Hence, the relative performance between any set of models can be predicted by lagged relative performance and RV. These results also confirm the good performance of the GARCH tree t+5. Against all other models, this model outperforms the others at least 77% of the time. Compared to the unconditional tests of Table IV, where some models had a lower QLIKE loss than the GARCH N model but failed to reject the null of equal unconditional predictive ability, the conditional tests find evidence of superior conditional performance for these models. This holds for the GARCH tree N and GARCH tree t, which suggests they perform equally well on average but with the use of the variables in the test function we can predict when one does better than the other. The proportion of times the random forests and GARCH tree N+5 are selected provides further evidence of their bad position.

Moving on to the 5-day-ahead forecasts, Table VI shows the results of the unconditional predictive ability test, MCS p-values and average QLIKE loss, similar to Table IV. The average losses have gone up, while the potential gains over the GARCH N model are now much smaller. The only model that rejects equal unconditional predictive ability is the GARCH t model. As in Table IV, the GARCH tree N+5 and random forests perform on average worse than the GARCH N. A reason for the vanishing gains of the GARCH trees could be the recursion applied in (12) to obtain the 5-day-ahead predictions. Because the regressors are unknown at time $t+4$, we have used the time t relevant terminal node possibly causing the volatility prediction to either

Table VI

Forecasting Results - 5-day-ahead

This table reports the out-of-sample forecasting performance of all models based on 5-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 7/1/2011 until 12/11/2021. The table shows the average and standard deviation of the QLIKE losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	QLIKE (Std)	GW stat	p_{MCS}
N	0.559 (1.453)	$*$	0.040
t	0.551 (1.516)	-2.638	**0.991**
GJR	0.550 (1.461)	-0.952	**1.000**
Tree N	0.553 (1.613)	-0.548	**0.991**
Tree t	0.557 (1.714)	-0.163	**0.941**
Tree N+5	0.717 (1.654)	2.818	0.004
Tree t+5	0.551 (1.518)	-0.700	**0.991**
RF N	0.818 (0.855)	5.978	0.000
RF t	0.787 (0.934)	5.839	0.000

revert too fast or too slow to the unconditional variance. The MCS now contains the 5 models with a lower average QLIKE loss than the GARCH N. Because the differences are smaller, the MCS procedure cannot detect which model is best.

Table VII reports the outcomes of the 5-day-ahead conditional predictive ability tests. While in Table V all pairwise hypotheses could be rejected, some tests now come up with insignificant results. Focusing on the GARCH tree models, only the GARCH t and GJR-GARCH models outperform these models more than 50% of the time, except for the GARCH tree t+5. The lagged relative performance and RV are also not able to distinguish between the GARCH tree N, GARCH tree t, and GARCH tree t+5 which tree model is best at a certain point in time.

Table VII

Conditional Predictive Ability Tests - 5-day-ahead

This table reports results of pairwise tests of equal conditional predictive ability for all models based on 5-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The table shows the p-values of equal conditional predictive ability for the models in the corresponding row and column. The loss is the QLIKE loss and the test function is $h_t = (1, \Delta L_t, RV_t)'$. The numbers in parentheses are the proportion of times the model in the column outperforms the model in the row over the out-of-sample period using the decision rule of Section 2.7.2. A plus (minus) sign indicates the test of equal conditional predictive ability is rejected at the 5% level and that the model in the column (row) outperforms the model in the row (column) more than 50% of the time.

	N	t	GJR	Tree N	Tree t	Tree N+5	Tree t+5	RF N
t	0.000^{-}							
	(0.02)							
GJR	0.129	0.422						
	(0.18)	(0.36)						
Tree N	0.001^{-}	0.002^{+}	0.002^{+}					
	(0.38)	(0.59)	(0.65)					
Tree t	0.012^{-}	0.006^{+}	0.054	0.429				
	(0.46)	(0.73)	(0.81)	(0.81)				
Tree N+5	0.018^{+}	0.013^{+}	0.005^{+}	0.013^{+}	0.022^{+}			
	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)			
Tree t+5	0.188	0.253	0.627	0.066	0.178	0.003^{-}		
	(0.21)	(0.49)	(0.59)	(0.39)	(0.24)	(0.00)		
RF N	0.000^{+}	0.000^{+}	0.000^{+}	0.000^{+}	0.000^{+}	0.005^{+}	0.000^{+}	
	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.97)	(0.99)	
RF t	0.000^{+}	0.000^{+}	0.000^{+}	0.000^{+}	0.000^{+}	0.012^{+}	0.000^{+}	0.000^{-}
	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.94)	(0.99)	(0.02)

Table VIII

Forecasting Results - 20-day-ahead

This table reports the out-of-sample forecasting performance of all models based on 20-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 31/1/2011 until 12/11/2021. The table shows the average and standard deviation of the QLIKE losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	QLIKE (Std)	GW stat	p_{MCS}
GARCH - N	0.781 (2.057)	$*$	**0.397**
GARCH - t	0.774 (2.188)	-0.796	**1.000**
GJR-GARCH	0.789 (2.073)	1.129	**0.397**
GARCH Tree - N	0.796 (2.284)	0.705	**0.397**
GARCH Tree - t	0.845 (2.948)	1.204	**0.265**
GARCH Tree - $N+5$	1.003 (2.211)	4.463	0.000
GARCH Tree - $t+5$	0.857 (2.044)	4.057	0.001
GARCH Random Forest - N	1.016 (1.181)	3.417	0.000
GARCH Random Forest - t	0.971 (1.266)	3.069	0.001

Tables VIII and IX show the results for the unconditional and conditional predictive ability test using the 20-day-ahead forecasts. In the unconditional case, the test outcomes are insignificant or in favour of the GARCH N model. The average QLIKE losses of the GARCH trees are all higher than the one for GARCH N. The MCS excludes the GARCH tree N+5, GARCH tree t+5 and random forests. The use of the additional splitting variables negatively impacts the performance on a long horizon. The conditional tests also have a hard time finding significant differences in performance; most p-values are too high. Among the better performing models, no pairwise test rejects the null hypothesis.

Table IX

Conditional Predictive Ability Tests - 20-day-ahead

This table reports results of pairwise tests of equal conditional predictive ability for all models based on 20-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The table shows the p-values of equal conditional predictive ability for the models in the corresponding row and column. The loss is the QLIKE loss and the test function is $h_t = (1, \Delta L_t, RV_t)'$. The numbers in parentheses are the proportion of times the model in the column outperforms the model in the row over the out-of-sample period using the decision rule of Section 2.7.2. A plus (minus) sign indicates the test of equal conditional predictive ability is rejected at the 5% level and that the model in the column (row) outperforms the model in the row (column) more than 50% of the time.

	N	t	GJR	Tree N	Tree t	Tree N+5	Tree t+5	RF N
t	0.664							
	(0.02)							
GJR	0.479	0.384						
	(0.92)	(1.00)						
Tree N	0.688	0.392	0.764					
	(0.91)	(0.93)	(0.82)					
Tree t	0.550	0.366	0.487	0.646				
	(0.97)	(0.97)	(0.94)	(0.99)				
Tree N+5	0.000^{+}	0.000^{+}	0.000^{+}	0.004^{+}	0.267			
	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)			
Tree t+5	0.004^{+}	0.017^{+}	0.003^{+}	0.242	0.911	0.015^{-}		
	(1.00)	(1.00)	(0.99)	(1.00)	(0.62)	(0.02)		
RF N	0.011^{+}	0.021^{+}	0.015^{+}	0.080	0.277	0.931	0.089	
	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.96)	(0.98)	
RF t	0.049^{+}	0.080	0.055	0.136	0.559	0.879	0.215	0.000^{-}
	(0.99)	(0.99)	(0.98)	(0.99)	(0.99)	(0.01)	(0.98)	(0.00)

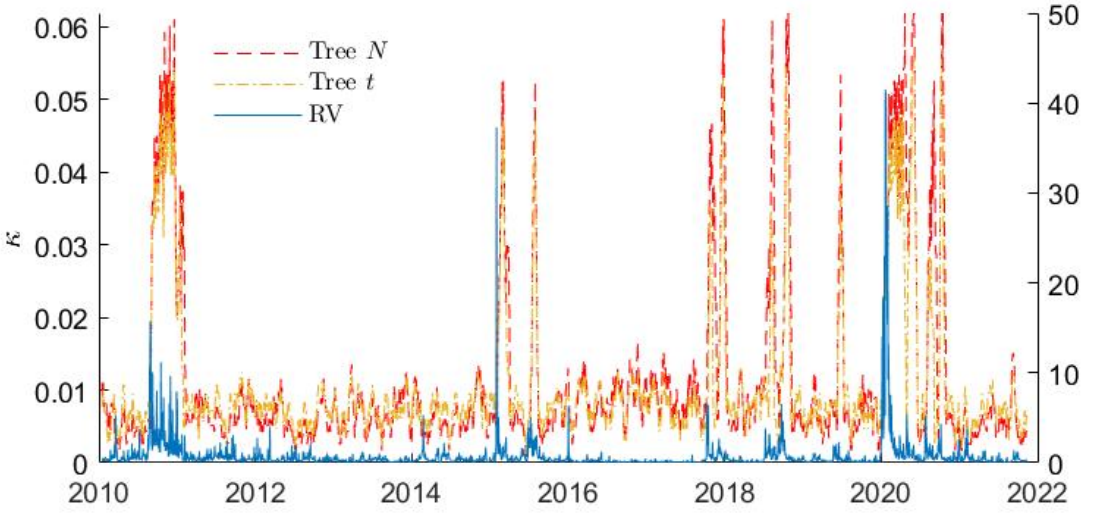

Figure 10: Realized variance and a 10-day moving average of the κ parameter of the GARCH Tree N and GARCH Tree t over the out-of-sample period 1/1/2011-12/11/2021.

For more insight into the behavior of the volatility predictions of the GARCH trees, Figures 10 and 11 plot a 10-day moving average of the κ and α parameter with RV for the GARCH tree N and GARCH tree t over the out-of-sample period. The κ parameter, controlling the speed at which forecasts revert to the unconditional variance, is relatively high during volatile periods. The α parameter seems to increase shortly when RV is high but stays at a fixed bandwith over the entire period. Compare this to the parameter time series of the other tree models. Appendix A.1 contains the figures for the GARCH tree N+5, GARCH tree t+5, and random forests. Note that the values κ and α can actually attain are in Table III for the single tree models. For κ, the response to periods of high RV is delayed compared to Figure 10. In case of the GARCH tree N+5, this parameter also rises when RV is low. The relative size of κ is also more than 10 times higher in the random forest. The behavior of α differs as well. Where in Figure 11 the parameter has a strong saw-like pattern, it has extended periods of high or low values in the other models, especially in the random forests. Its values are high during

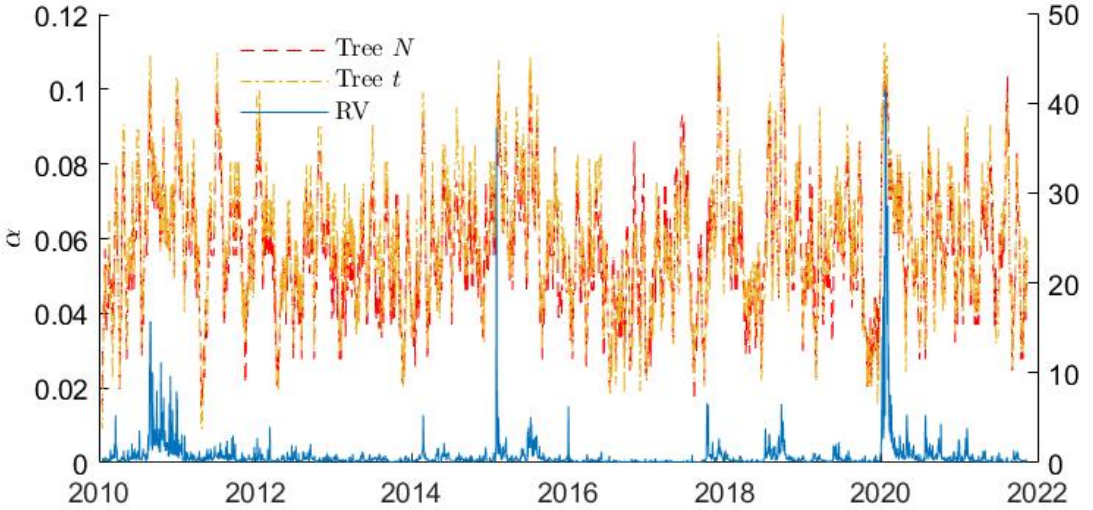

Figure 11: Realized variance and a 10-day moving average of the α parameter of the GARCH Tree N and GARCH Tree t over the out-of-sample period 1/1/2011-12/11/2021.

periods of low volatility, where one would expect high values in volatile periods because then volatility should increase due to large returns. Similar to κ, the α parameter is much larger for the random forests than for the other models.

This section has shown that the forecasting performance of the GARCH tree models is not always an improvement over the traditional GARCH models. Only in case of the 1-day-ahead forecasts does the GARCH tree t+5 clearly better than the others. At longer horizons the GARCH trees perform equal or worse. The distributional assumption only seems to matter for the GARCH trees when additional splitting variables are added. The random forests stand out by their bad performance. Why the performance is not always beneficial is part of the next section.

4.3 Extensions

4.3.1 Varying Maximum Number of Terminal Nodes

The mixed results of the previous section motivate for a detailed investigation of the sensitivity of the results with respect to the choices made beforehand. As became clear from the GARCH tree estimates of Section 4.1 neither of the trees was pruned. All trees had the maximum of six terminal nodes. The AIC criterion did not select any of the subtrees because the criterion was at its lowest for the fully grown trees. The degrading performance of the GARCH trees on a longer horizon suggest the GARCH trees might have been overparameterized at their fully grown size. Therefore, we investigate the performance of the GARCH trees with less terminal nodes. To get a full overview at each amount of terminal nodes, the trees are not pruned this time. Table X shows the average QLIKE loss at each forecasting horizon for the GARCH tree N, GARCH tree t, GARCH tree N+5 and GARCH tree t+5, while varying the number of terminal nodes from 2 to 6. In addition, an asterisk is shown at the amount of terminal nodes that would have been selected if we used the BIC criterion instead of the AIC criterion. For the 1-day-ahead forecasts, the four GARCH tree models except the GARCH tree N+5 already perform well with only 3 terminal nodes. There, they are at their best or very close to their best. The GARCH tree N+5 gets only worse from 3 terminal nodes onwards. Remember that at 1 terminal node we are back at the GARCH N and GARCH t models. At the 5-day horizon, the results are similar to the 1-day-ahead forecasts with 3 terminal nodes already reaching the minimum QLIKE loss. The best result for the GARCH tree N+5 can now be found in Table VI, with 1 terminal node. At the 20-day horizon, 1 terminal node is optimal for all models. For short term forecasting, introducing more than 3 terminal nodes seems unfavorable for a low average QLIKE loss. At the long horizon the tree structure is not able to make any gains over the standard GARCH recursion. The BIC criterion selects 2 terminal nodes for the GARCH N, GARCH t, and GARCH t+5 models. In case of the GARCH N+5, the BIC would have chosen 6 terminal nodes as optimal. Compared with the average QLIKE losses at each terminal node, the BIC does not choose the amount of terminal nodes corresponding to the lowest QLIKE loss. Therefore, the BIC also seems to fail as a criterion to select the amount of terminal nodes of the GARCH trees.

Table X

Forecasting Results - Varying Number of Terminal Nodes M

This table reports the out-of-sample forecasting performance of the four single GARCH tree models for different values of M, the number of terminal nodes of the tree. See Table IV for more details. The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 1/1/2011 until 12/11/2021. The table shows the average and standard deviation of the QLIKE losses for 1-day-ahead (QLIKE1), 5-day-ahead (QLIKE5), and 20-day-ahead (QLIKE20) forecasts. the values of M with an asterisk * would be selected if the BIC criterion was used.

Panel A: GARCH Tree - N			
	QLIKE1 (Std)	QLIKE5 (Std)	QLIKE20 (Std)
$M = 2^*$	0.424 (0.563)	0.566 (1.585)	0.812 (2.015)
$M = 3$	0.421 (0.559)	0.556 (1.654)	0.783 (2.174)
$M = 4$	0.429 (0.611)	0.568 (1.627)	0.863 (2.789)
$M = 5$	0.427 (0.611)	0.594 (1.673)	0.873 (2.783)
$M = 6$	0.420 (0.546)	0.553 (1.613)	0.796 (2.284)
Panel B: GARCH Tree - t			
	QLIKE1 (Std)	QLIKE5 (Std)	QLIKE20 (Std)
$M = 2^*$	0.414 (0.566)	0.559 (1.640)	0.807 (2.063)
$M = 3$	0.396 (0.559)	0.542 (1.687)	0.822 (2.809)
$M = 4$	0.409 (0.550)	0.562 (1.744)	0.840 (2.857)
$M = 5$	0.396 (0.541)	0.543 (1.774)	0.806 (2.908)
$M = 6$	0.418 (0.560)	0.557 (1.714)	0.845 (2.948)
Panel C: GARCH Tree - $N + 5$			
	QLIKE1 (Std)	QLIKE5 (Std)	QLIKE20 (Std)
$M = 2$	0.424 (0.563)	0.566 (1.585)	0.812 (2.015)
$M = 3$	0.472 (0.518)	0.613 (1.247)	0.864 (1.870)
$M = 4$	0.451 (0.516)	0.620 (1.189)	0.926 (1.990)
$M = 5$	0.454 (0.546)	0.625 (1.201)	0.932 (2.001)
$M = 6^*$	0.553 (1.331)	0.717 (1.654)	1.003 (2.211)
Panel D: GARCH Tree - $t + 5$			
	QLIKE1 (Std)	QLIKE5 (Std)	QLIKE20 (Std)
$M = 2^*$	0.414 (0.566)	0.559 (1.640)	0.807 (2.063)
$M = 3$	0.377 (0.556)	0.540 (1.570)	0.818 (1.969)
$M = 4$	0.380 (0.578)	0.546 (1.625)	0.827 (2.037)
$M = 5$	0.383 (0.592)	0.548 (1.616)	0.826 (2.036)
$M = 6$	0.380 (0.625)	0.551 (1.518)	0.857 (2.044)

4.3.2 Varying Stationary Bootstrap Block Length

One thing that stands out from 4.2 is the poor performance of the random forest models across all horizons. The random forest forecasts were created by fitting trees to bootstrap samples and then averaging the predictions. The bad performance could originate from the bootstrap samples. If the bootstrap samples are not a good representation of the behavior in the actual data, then the fitted trees are not going to produce reliable forecasts. An indication for this could be the values of the κ and α parameters, which were much higher than for the single GARCH tree models, as mentioned in Section 4.2. The only parameter controlling the behavior of the bootstrap samples is the expected block length w of the stationary bootstrap method. This parameter was set to 10 for the random forests. A larger value for the expected block length would let the bootstrap samples behave more like the actual data. Therefore, Table XI shows the average QLIKE losses of the random forests across all forecasting horizons and for various values of the expected block length w, ranging from 10 to 2000 (the estimation sample size is 2755). Note that when a block starting close tot the end of the sample is selected it continues from the first observation onwards to obtain a block of the requested size. Moreover, Table XI shows the average QLIKE loss if instead of fitting the 100 trees of the random forests

Table XI

Forecasting Results - Varying Expected Block Length w

This table reports the out-of-sample forecasting performance of the two GARCH random forest models for different values of w, the average block length of the blocks in the stationary bootstrap. The number of trees in the random forest is 100. The out-of-sample period is from 1/1/2011 until 12/11/2021. The table shows the average and standard deviation of the QLIKE losses for 1-day-ahead (QLIKE1), 5-day-ahead (QLIKE5), and 20-day-ahead (QLIKE20) forecasts. The NO indicates a random forest where the actual data was taken to fit the trees, instead of bootstrap samples.

	RF N			RF t		
w	QLIKE1 (Std)	QLIKE5 (Std)	QLIKE20 (Std)	QLIKE1 (Std)	QLIKE5 (Std)	QLIKE20 (Std)
10	0.573 (0.494)	0.818 (0.855)	1.016 (1.181)	0.617 (0.523)	0.787 (0.934)	0.971 (1.266)
20	0.515 (0.470)	0.752 (0.879)	0.980 (1.254)	0.482 (0.452)	0.732 (0.939)	0.979 (1.253)
30	0.494 (0.463)	0.713 (1.017)	0.945 (1.354)	0.456 (0.455)	0.695 (1.030)	0.950 (1.334)
40	0.492 (0.475)	0.695 (1.049)	0.924 (1.439)	0.463 (0.462)	0.676 (1.089)	0.928 (1.411)
60	0.481 (0.481)	0.677 (1.073)	0.915 (1.438)	0.446 (0.455)	0.638 (1.102)	0.894 (1.565)
120	0.480 (0.488)	0.645 (1.198)	0.876 (1.592)	0.450 (0.479)	0.626 (1.245)	0.870 (1.647)
250	0.489 (0.504)	0.644 (1.236)	0.874 (1.688)	0.440 (0.496)	0.607 (1.312)	0.854 (1.779)
500	0.475 (0.500)	0.617 (1.179)	0.846 (1.773)	0.445 (0.489)	0.592 (1.255)	0.832 (1.842)
1000	0.497 (0.526)	0.647 (1.211)	0.867 (1.755)	0.452 (0.513)	0.599 (1.302)	0.830 (1.849)
2000	0.503 (0.525)	0.639 (1.236)	0.861 (1.797)	0.452 (0.511)	0.596 (1.300)	0.838 (1.835)
NO	0.428 (0.506)	0.572 (1.293)	0.827 (1.985)	0.413 (0.513)	0.556 (1.344)	0.815 (2.073)

to the bootstrap samples, the 100 trees are fitted to the actual data such that only the random variable selection remains of the original random forest procedure (NO in the table). Starting from an expected block length of 10, the QLIKE losses are a decreasing function of the expected block length. After an expected block length of 500, the QLIKE losses start to increase again. This means that around 500 the gains from using a bootstrap sample more similar to the actual data have run out. These gains, however, are still relatively limited compared to the results for the other models: under both distributional assumptions the average QLIKE losses are still above the average QLIKE losses of the GARCH N model. If the trees are fitted to the actual data, keeping only the variable selection intact, the QLIKE losses are lower than for any achieved by the stationary bootstrap samples. This of course raises the question whether the stationary bootstrap is able to generate useful samples for our purpose at all. One possible reason might be the fixed estimation sample such that the random forest are not able to adapt to most recent events. Oh and Patton (2021) found for example that two standard GARCH models with a moving window were able to significantly outperform a GARCH model with a fixed estimation sample. This could potentially overcome the issues with the random forests.

4.3.3 Squared Error Loss

Thus far, we have evaluated the performance of all models based on the QLIKE loss function of equation (13). Given the fact that the QLIKE loss function puts a heavier penalty on under-prediction than on over-prediction, a different loss function might lead to different conclusions regarding the accuracy of the models. Therefore, Tables XII, XIII, and Table XVI in Appendix A.2 show results when using squared error loss for 1-day, 5-day, and 20-day-ahead forecasts respectively. Each table shows the average squared error loss (MSE), GW statistics of the unconditional predictive ability test with the GARCH N as benchmark, and the MCS p-values. The results are very different from the QLIKE results of the previous section. Where the average loss at the 1-day horizon was lower for GARCH t and GJR-GARCH than GARCH N

Table XII

Forecasting Results - 1-day-ahead SE loss

This table reports the out-of-sample forecasting performance of all models based on 1-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 7/1/2011 until 12/11/2021. The table shows the average and standard deviation of the squared error (SE) losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	SE (Std)	GW stat	p_{MCS}
GARCH - N	4.767 (42.903)	$*$	**0.292**
GARCH - t	4.844 (43.334)	1.177	**0.292**
GJR-GARCH	5.712 (51.717)	0.895	**0.167**
GARCH Tree - N	3.462 (35.190)	-1.176	**0.666**
GARCH Tree - t	3.798 (37.452)	-1.024	**0.292**
GARCH Tree - $N+5$	4.647 (39.798)	-0.168	**0.167**
GARCH Tree - $t+5$	6.714 (61.833)	1.094	**0.167**
GARCH Random Forest - N	3.349 (36.586)	-1.088	**0.666**
GARCH Random Forest - t	3.167 (38.682)	-0.997	**1.000**

using QLIKE loss, the MSE are now higher. The unconditional GW tests can not be rejected however. In line with the QLIKE situation, do the GARCH tree N and GARCH tree t provide lower average losses than GARCH N. But where the introduction of the additional splitting variables led to a further improvement of GARCH t in the QLIKE case, do both GARCH trees with the additional splitting variables worse than their counterparts. The performance of the random forests has also changed a lot, with the lowest MSEs in the table. Although the ordering of average losses is quite different from the QLIKE situation, does no model lead to such an improvement over GARCH N that the unconditional predictive ability test can be rejected. Even the MCS cannot make a distinction, as it contains all models. These findings also hold for the 5-day-ahead and 20-day-ahead forecasts. Why are the results under squared error loss so different? A downside of squared error loss is that it blows up large errors. As a small example, consider the 1-day-ahead differences in squared error losses between the GARCH N and random forest t. On average this difference is -1.6. However, if we set the extreme loss differences between the two with an absolute value above 20 to zero, the average difference is 0.12 in favor of GARCH N. By filtering out above 20, we have removed less than 2% of the

Table XIII

Forecasting Results - 5-day-ahead SE loss

This table reports the out-of-sample forecasting performance of all models based on 5-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 7/1/2011 until 12/11/2021. The table shows the average and standard deviation of the squared error (SE) losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	SQL (Std)	GW stat	p_{MCS}
N	6.700 (58.018)	$*$	**0.273**
t	6.804 (58.501)	1.206	**0.273**
GJR	8.147 (73.194)	1.037	**0.254**
Tree N	4.824 (48.730)	-1.220	**0.454**
Tree t	5.215 (51.351)	-1.167	**0.273**
Tree N+5	6.656 (54.657)	-0.044	**0.096**
Tree t+5	9.546 (88.441)	1.090	**0.096**
RF N	4.836 (45.012)	-1.059	**0.304**
RF t	4.313 (46.072)	-1.059	**1.000**

forecasts. An unconditional predictive ability test would be rejected in this sample. If we repeat this exercise with the QLIKE losses by adjusting the same observations, the QLIKE difference becomes larger, from 0.19 to 0.2. Therefore it seems, under squared error loss, that the outliers heavily influence the results disguising the bad performance during most of the sample.

4.3.4 Other Indices

The GARCH tree models have only been applied to S&P500 returns. To investigate whether the results also hold for other indices, we perform the same analysis as in Section 4.2 to the FTSE 100 (UK), DAX 40 (Germany) and Nikkei 225 (Japan) index, without the conditional predictive ability tests. Similar to the S&P500, we collect the closing prices and 5-minute RV from the Oxford-Man Realized Library. We do not include economic indices or a VIX-like volatility measure as additional splitting variables. This is because the economic indices we have used or either not available for other countries or on a much lower frequency. The EPU for

Table XIV
Forecasting Results - Other Indices

This table reports the out-of-sample forecasting performance of all models based on 1-day-ahead, 5-day-ahead, and 20-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). Panel A contains results for the FTSE 100 index, Panel B contains results for the DAX 40, and Panel C for the Nikkei 225 index. The models are estimated once, where the estimation samples are fixed to 5/1/2000-31/12/2010, 4/1/2000-31/12/2010, and 3/2/2000-31/12/2010 respectively. The out-of-sample periods are 4/1/2011-15/10/2021, 3/1/2011-15/10/2021, and 4/1/2011-15/10/2021 respectively. The table shows the average and standard deviation of the QLIKE losses for 1-day-ahead (QLIKE1), 5-day-ahead (QLIKE5), and 20-day-ahead (QLIKE20) forecasts, their t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and their MCS p-values.

Panel A: FTSE 100						
	QLIKE1 (GW stat)	p_{MCS}	QLIKE5 (GW stat)	p_{MCS}	QLIKE20 (GW stat)	p_{MCS}
N	0.278 (*)	**0.140**	0.355 (*)	**0.221**	0.505 (*)	**0.396**
t	0.279 (1.666)	**0.094**	0.355 (0.409)	**0.221**	0.505 (-0.435)	**0.396**
GJR	0.264 (-1.427)	**0.544**	0.337 (-0.340)	**0.731**	0.499 (-0.325)	**0.411**
Tree N	0.261 (-1.841)	**0.602**	0.333 (-1.906)	**1.000**	0.486 (-1.366)	**1.000**
Tree t	0.274 (-0.402)	**0.094**	0.345 (-0.734)	**0.246**	0.513 (0.292)	**0.396**
Tree N+1	0.259 (-1.746)	**0.602**	0.383 (1.867)	0.000	0.605 (3.368)	0.000
Tree t+1	0.256 (-2.099)	**1.000**	0.339 (-1.027)	**0.731**	0.527 (0.882)	**0.244**
Panel B: DAX 40						
	QLIKE1 (GW stat)	p_{MCS}	QLIKE5 (GW stat)	p_{MCS}	QLIKE20 (GW stat)	p_{MCS}
N	0.325 (*)	0.000	0.380 (*)	0.000	0.528 (*)	0.005
t	0.324 (-0.695)	0.000	0.376 (-2.629)	0.000	0.514 (-3.171)	**1.000**
GJR	0.308 (-1.877)	0.000	0.374 (-0.698)	0.000	0.541 (1.614)	0.005
Tree N	0.302 (-2.864)	0.000	0.383 (0.292)	0.000	0.581 (3.870)	0.001
Tree t	0.308 (-2.072)	0.000	0.379 (-0.171)	0.000	0.540 (1.131	0.005
Tree N+1	0.275 (-5.334)	**1.000**	0.348 (-3.399)	**0.273**	0.566 (0.915)	0.005
Tree t+1	0.278 (-5.173)	**0.492**	0.342 (-4.038)	**1.000**	0.532 (0.144)	0.005
Panel C: Nikkei 225						
	QLIKE1 (GW stat)	p_{MCS}	QLIKE5 (GW stat)	p_{MCS}	QLIKE20 (GW stat)	p_{MCS}
N	0.632 (*)	0.000	0.707 (*)	0.000	0.846 (*)	0.000
t	0.629 (-0.911)	0.000	0.697 (-3.306)	0.000	0.819 (-4.756)	**0.928**
GJR	0.615 (-2.173)	0.000	0.703 (-0.558)	0.000	0.868 (3.057)	0.000
Tree N	0.604 (-2.260)	0.000	0.688 (-1.491)	0.000	0.811 (-1.387)	**1.000**
Tree t	0.791 (7.015)	0.000	0.968 (7.223)	0.000	1.058 (6.458)	0.000
Tree N+1	0.503 (-6.625)	**1.000**	0.625 (-4.810)	**1.000**	0.820 (-1.525)	**0.928**
Tree t+1	0.505 (-6.556)	0.000	0.626 (-4.778)	**0.176**	0.820 (-1.498)	**0.928**

example is on a monthly basis for the UK, Germany and Japan (discontinued). An option implied volatility measure is not included because or results so far have totally ignored the VIX. Oh and Patton (2021) have shown that superior forecasting can be achieved using RV rather than the VIX. Table XIV shows the 1-day, 5-day, and 20-day-ahead average QLIKE losses with the GW statistic of the unconditional predictive ability test and MCS p-values, for all three indices. In case of the FTSE, no model is able to significantly outperform the GARCH N model, except the GARCH tree t with RV as additional splitting variable (t+1) on a 1-day horizon. For the DAX and Nikkei, significant gains can be made with the GARCH trees on a 1-day and 5-day horizon, similar to Section 4.2. Especially for the GARCH trees with RV as additional variable, since they appear in the MCS. The table supports the previous finding that short term forecasting with GARCH trees can be beneficial over the traditional GARCH models.

4.3.5 Lower Frequency

While most of the time volatility is modeled using daily returns, one could be interested in fitting a GARCH to less frequent observations to forecast volatility at long horizons. In Section 4.2, for example, we have evaluated the performance of the GARCH trees with 5-day-ahead and 20-day ahead horizons using a recursion based on the last known terminal node of the GARH tree. Because the terminal nodes have parameter values that deviate strongly from each other, long term recursion based forecasts might revert too quickly or too slow to the unconditional variance. To still forecast at long horizons, one could fit the GARCH tree to less frequent observations such that long term forecasts become one step ahead forecasts. Table XV and Table XVII in Appendix A.3 show results of 1-step-ahead forecasts after turning the data to weekly and monthly observations respectively. These observations have been created by using the Monday closing prices with either one week or four weeks in between to calculate weekly and monthly returns[5]. The pre 2011 and post 2011 split for the estimation sample and out-of-sample

Table XV
Forecasting Results - 1-week-ahead

This table reports the out-of-sample forecasting performance of all models based on 1-week-ahead predictions of S&P500 volatility after adjusting the data to a weekly frequency (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 10/1/2000-28/12/2010. The out-of-sample period is from 3/1/2011 until 8/11/2021. The table shows the average and standard deviation of the QLIKE losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	QLIKE (Std)	GW stat	p_{MCS}
GARCH - N	1.559 (0.652)	$*$	0.003
GARCH - t	1.529 (0.646)	-2.277	**0.133**
GJR-GARCH	1.505 (0.611)	-2.797	**0.184**
GARCH Tree - N	1.994 (0.833)	4.141	0.000
GARCH Tree - t	1.869 (0.783)	3.448	0.000
GARCH Tree - N + 5	1.460 (0.681)	-2.129	**0.184**
GARCH Tree - t + 5	1.455 (0.677)	-2.300	**1.000**
GARCH Random Forest - N	1.621 (0.623)	2.458	0.000
GARCH Random Forest - t	1.592 (0.612)	1.492	0.002

[5]If Monday next week or 4 weeks later was missing, the Tuesday closing price was used.

period has been maintained[6]. The weekly and monthly RV and VIX are proxied by the averages of daily realizations of RV and VIX. The economic indices were kept the same. The results of Table XV show only the GARCH t, GJR-GARCH, GARCH tree N+5, and GARCH tree t+5, have lower average QLIKE losses than the GARCH N. They also reject equal unconditional predictive ability and form the MCS. In the 1-month-ahead results, only the GJR-GARCH is better than the GARCH N model. This could be because the trees are fitted using a relatively low number of observations (153), which is, given that they are not pruned by the AIC, very little for a 6 terminal node tree. Hence, we can only conclude that for weekly returns the GARCH trees perform similar to the best non-tree models.

5 Conclusion

The thesis has investigated the forecasting performance of the GARCH tree model. This has been done by applying the GARCH tree model in various forms to S&P500 data and comparing its accuracy to the GARCH and GJR-GARCH model. The forms of the GARCH tree model used throughout the thesis are: either assuming a normal or t distribution for the innovations, including or excluding additional splitting variables, and combining multiple trees to create random forests. The additional variables are RV, the VIX, and the EPU, ADS, and NFC index. Using the QLIKE losses to measure how accurate the forecasts are, we have implemented several ways to see if the gains in forecasting performance are significant. These are the unconditional and conditional tests of Giacomini and White (2006), and the MCS procedure of Hansen et al. (2011). Our main analysis shows that for 1-day-ahead forecasts, the GARCH tree is able to do better than the GARCH and GJR-GARCH model. This is, however, only true for the GARCH tree with t-distributed innovations and the additional splitting variables included. The other GARCH tree implementations perform on par with or worse than the traditional models. Especially the random forests, which were expected to do better given its model averaging and variable selection characteristics, disappoint. At longer horizons, the GARCH tree models are not able to do better, which could be because of the recursion we applied to obtain multi-step-ahead forecasts. They are obtained with the parameters of the most recent terminal node, which might be irrelevant at long horizons, causing the volatility prediction to revert too fast or too slow to the unconditional variance.

The main findings are extended in several directions. First, because in the estimation of the GARCH trees the AIC statistic failed to select a lower than the maximum number of terminal nodes as optimal, we manually lowered the number of terminal nodes in the GARCH trees. This shows that the maximum of six terminal nodes does not lead to the lowest average loss. Three nodes seems to be sufficient. Replacing the AIC with the BIC does not really overcome this problem. Second, the expected block length of the stationary bootstrap has a large impact on the performance of the random forests. Although at a block length of 500 the average losses are at a minimum, the random forests still do worse than all other models. Third, using the squared error loss function instead of the QLIKE loss function sheds a different light on the GARCH trees underperforming in the main analysis. However, this is most likely because the

[6]This leaves 591 (569) and 153 (143) as number of observations for the estimation sample (out-of-sample period).

squared error loss is more sensitive to outliers. The QLIKE loss is therefore preferred over the squared error loss. Fourth, we test whether our results also hold for other stock indices. For the DAX and Nikkei, the GARCH tree outperforms the traditional GARCH models at the 1-day and 5-day-ahead horizons. For the FTSE, the differences are too small to be significant. Finally, at a lower observation frequency (weekly and monthly), do the GARCH trees not outperform the traditional models. Possibly because of the low amount of observations.

Several contributions have been made to the existing literature. We have performed a forecasting study to show the potential of the GARCH tree model. In addition, more splitting variables were added to the GARCH tree, which has only been modeled with past return and variance previously. The implementation of the GARCH tree with VTE also shows the computation time can be reduced substantially making the GARCH tree more accessible in practice. Finally, the extensions of Section 4.3 reveal the specification of the trees is important for their performance.

The applied methodology also has a few shortcomings. In fitting the GARCH tree we have maximized the reduced log-likelihood to determine the splitting rules and starting values for the model likelihood optimization. This could lead to local instead of global maxima and suboptimal parameter estimates. Moreover, the decision to set the splitting values equal to empirical quantiles could result in suboptimal trees. To obtain multi-step-ahead forecasts, we have used the last know terminal node but this might not be entirely valid. The long term forecasting performance of the GARCH trees could therefore be biased.

Further research can use Section 4.3 as a starting point. The fact that the AIC and BIC criterion failed to select the number of terminal nodes with the lowest average QLIKE loss suggests the size of the tree should be based on something else. Perhaps using the performance on a subsample as a criterion, as in Oh and Patton (2021), leads to a better choice. The random forest results are worse than the single tree results. To counter this problem, and to investigate the tree performance more generally, a moving estimation window could improve the performance of the GARCH trees significantly.

References

Amendola, A., Candila, V., & Gallo, G. M. (2019). On the asymmetric impact of macro–variables on volatility. *Economic Modelling*, *76*, 135–152.

Andersen, T. G., & Bollerslev, T. (1998). Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. *International economic review*, 885–905.

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, *71*(2), 579–625.

Aruoba, S. B., Diebold, F. X., & Scotti, C. (2009). Real-time measurement of business conditions. *Journal of Business & Economic Statistics*, *27*(4), 417–427.

Audrino, F., & Bühlmann, P. (2001). Tree-structured generalized autoregressive conditional heteroscedastic models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(4), 727–744.

Audrino, F., & De Giorgi, E. (2007). Beta regimes for the yield curve. *Journal of Financial Econometrics*, *5*(3), 456–490.

Audrino, F., & Trojani, F. (2011). A general multivariate threshold garch model with dynamic conditional correlations. *Journal of Business & Economic Statistics*, *29*(1), 138–149.

Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring economic policy uncertainty. *The quarterly journal of economics*, *131*(4), 1593–1636.

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

Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The review of economics and statistics*, 542–547.

Brave, S. A., & Kelley, D. (2017). Introducing the chicago fed's new adjusted national financial conditions index. *Chicago Fed Letter*, *386*, 2017.

Buncic, D., & Gisler, K. I. (2016). Global equity market volatility spillovers: A broader role for the united states. *International Journal of Forecasting*, *32*(4), 1317–1339.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, 987–1007.

Engle, R. F., Ghysels, E., & Sohn, B. (2013). Stock market volatility and macroeconomic fundamentals. *Review of Economics and Statistics*, *95*(3), 776–797.

Engle, R. F., & Mezrich, J. (1996). Garch for groups. *Risk*, *9*(8), 36–40.

Engle, R. F., & Ng, V. K. (1993). Measuring and testing the impact of news on volatility. *The journal of finance*, *48*(5), 1749–1778.

Fan, J., Qi, L., & Xiu, D. (2014). Quasi-maximum likelihood estimation of garch models with heavy-tailed likelihoods. *Journal of Business & Economic Statistics*, *32*(2), 178–191.

Francq, C., Horvath, L., & Zakoıan, J.-M. (2011). Merits and drawbacks of variance targeting in garch models. *Journal of Financial Econometrics*, *9*(4), 619–656.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2009). *The elements of statistical learning* (Vol. 2). Springer series in statistics New York.

Giacomini, R., & White, H. (2006). Tests of conditional predictive ability. *Econometrica*, *74*(6), 1545–1578.

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, *48*(5), 1779–1801.

Goulet Coulombe, P. (2020). The macroeconomy as a random forest. *Available at SSRN 3633110*.

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.

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics letters*, *6*(3), 255–259.

Kim, J.-M., Jung, H., & Qin, L. (2016). Linear time-varying regression with a dcc-garch model for volatility. *Applied Economics*, *48*(17), 1573–1582.

Lee, T.-H., & Long, X. (2009). Copula-based multivariate garch model with uncorrelated dependent errors. *Journal of Econometrics*, *150*(2), 207–218.

Linton, O., & Mammen, E. (2005). Estimating semiparametric arch () models by kernel smoothing methods 1. *Econometrica*, *73*(3), 771–836.

Liu, M., Taylor, J. W., & Choo, W.-C. (2020). Further empirical evidence on the forecasting of volatility with smooth transition exponential smoothing. *Economic Modelling*, *93*, 651–659.

Medeiros, M. C., & Veiga, A. (2009). Modeling multiple regimes in financial volatility with a flexible coefficient garch (1, 1) model. *Econometric Theory*, *25*(1), 117–161.

Oh, D. H., & Patton, A. J. (2021). Better the devil you know: Improved forecasts from imperfect models. *Available at SSRN 3925545*.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, *160*(1), 246–256.

Politis, D. N., & Romano, J. P. (1994). The stationary bootstrap. *Journal of the American Statistical association*, *89*(428), 1303–1313.

Rapach, D. E., & Strauss, J. K. (2008). Structural breaks and garch models of exchange rate volatility. *Journal of Applied Econometrics*, *23*(1), 65–90.

Schwert, G. W. (1989). Why does stock market volatility change over time? *The journal of finance*, *44*(5), 1115–1153.

Sheppard, K. (2018). *Matlab function reference financial econometrics*.

Wilhelmsson, A. (2006). Garch forecasting performance under different distribution assumptions. *Journal of Forecasting*, *25*(8), 561–578.

A Appendix

A.1 Time Series of κ and α

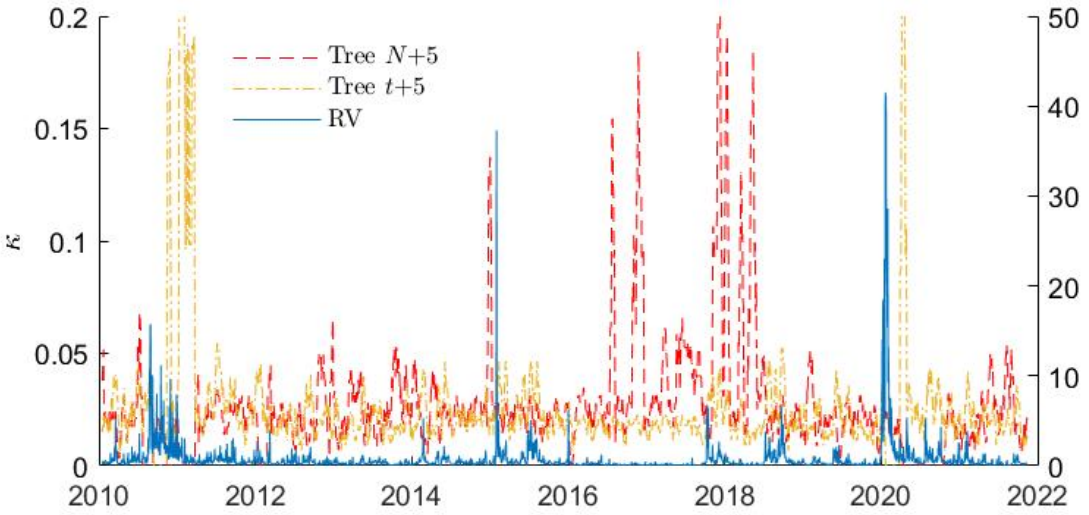

Figure 12: Realized variance and a 10-day moving average of the κ parameter of the GARCH Tree N+5 and GARCH Tree t+5 over the out-of-sample period 1/1/2011-12/11/2021.

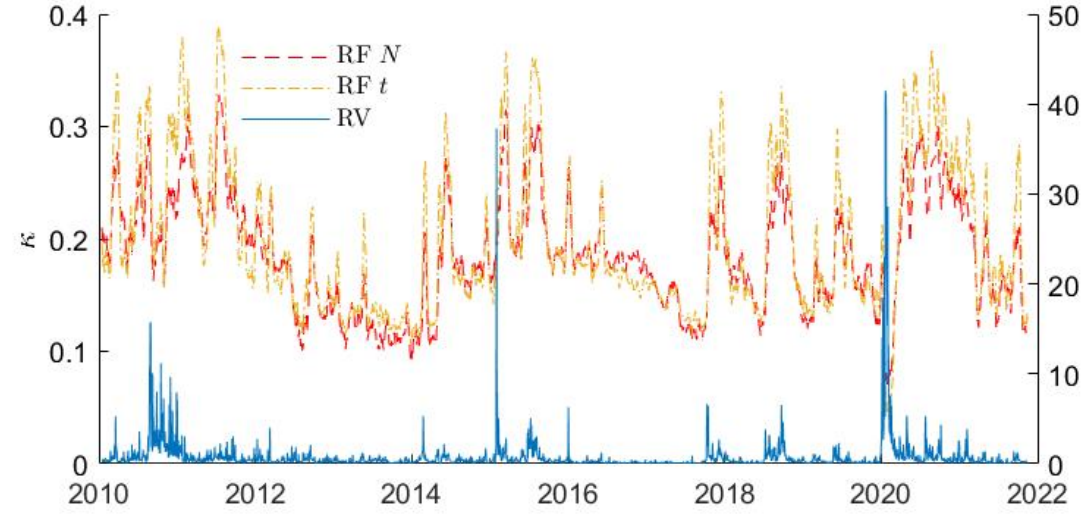

Figure 13: Realized variance and a 10-day moving average of the κ parameter of the Random Forest N and Random Forest t over the out-of-sample period 1/1/2011-12/11/2021.

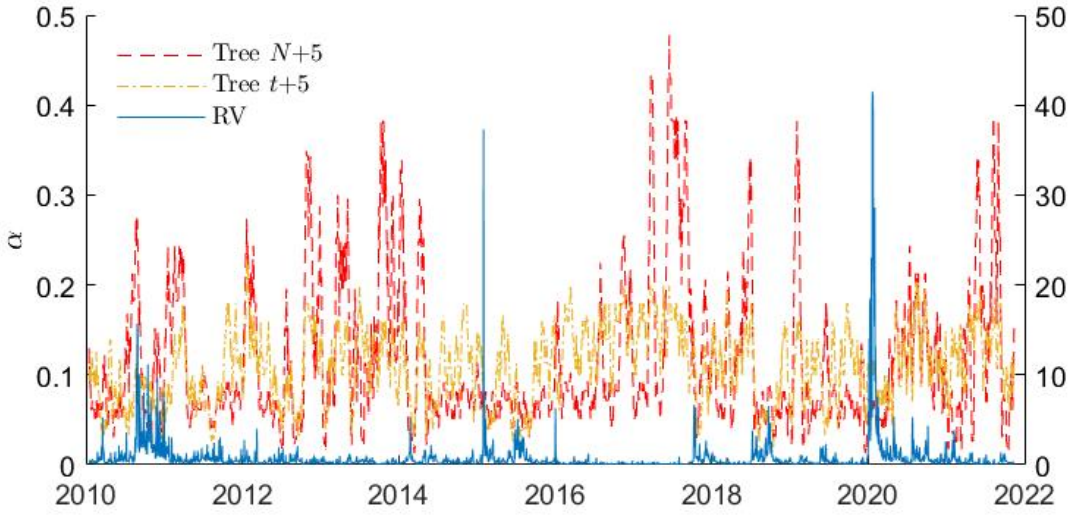

Figure 14: Realized variance and a 10-day moving average of the α parameter of the GARCH Tree N+5 and GARCH Tree t+5 over the out-of-sample period 1/1/2011-12/11/2021.

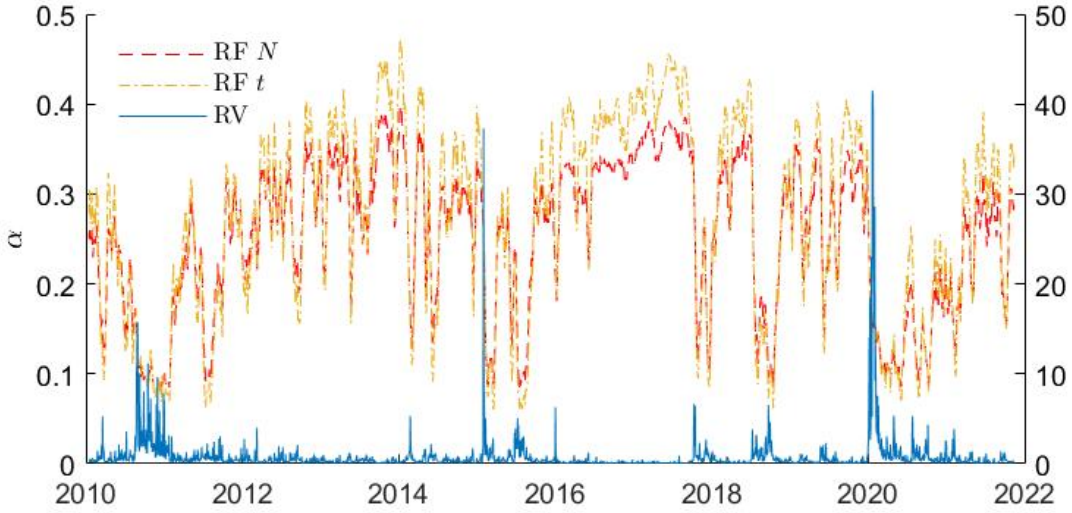

Figure 15: Realized variance and a 10-day moving average of the α parameter of the Random Forest N and Random Forest t over the out-of-sample period 1/1/2011-12/11/2021.

A.2 Squared Error Loss: 20-day-ahead Forecasts

Table XVI

Forecasting Results - 20-day-ahead SE

This table reports the out-of-sample forecasting performance of all models based on 20-day-ahead predictions of S&P500 volatility (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 4/1/2000-31/12/2010. The out-of-sample period is from 7/1/2011 until 12/11/2021. The table shows the average and standard deviation of the squared error (SE) losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	SQL (Std)	GW stat	p_{MCS}
N	9.260 (77.022)	$*$	**0.264**
t	9.520 (78.474)	1.327	**0.215**
GJR	11.121 (95.924)	1.056	**0.215**
Tree N	7.376 (78.072)	-1.143	**0.355**
Tree t	7.780 (82.446)	-1.028	**0.264**
Tree N+5	8.869 (71.239)	-0.284	**0.215**
Tree t+5	14.340 (128.897)	1.134	**0.215**
RF N	7.515 (69.213)	-0.913	**0.264**
RF t	6.186 (60.467)	-1.079	**1.000**

A.3 Forecasting Results - 1-month-ahead

Table XVII

Forecasting Results - 1-month-ahead

This table reports the out-of-sample forecasting performance of all models based on 1-month-ahead predictions of S&P500 volatility after adjusting the data to a monthly frequency (See Table IV for more details on model specifications). The models are estimated once, where the estimation sample is fixed to 31/1/2000-20/12/2010. The out-of-sample period is from 18/1/2011 until 8/11/2021. The table shows the average and standard deviation of the QLIKE losses, the t statistics of the equal unconditional predictive ability test with the GARCH with normally distributed innovations as benchmark (GW stat), and the MCS p-values.

	QLIKE (Std)	GW stat	p_{MCS}
N	2.542 (0.790)	$*$	0.000
t	2.549 (0.798)	1.664	0.000
GJR	2.357 (0.704)	-2.335	**1.000**
Tree N	3.215 (0.901)	2.476	0.000
Tree t	3.213 (0.900)	2.472	0.000
Tree N + 5	2.931 (0.978)	1.990	0.000
Tree t + 5	2.609 (0.943)	0.399	0.000
RF N	2.872 (0.783)	2.572	0.000
RF t	2.897 (0.784)	2.589	0.000