Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Enhancing CBOE VIX Forecasting: A Comparative Study of GARCH, ARIMA, HAR, and Tree-Based Models

Anna Grefhorst (605634)

Abstract

This paper investigates models for estimating and forecasting the CBOE VIX. The GARCH option pricing model largely underestimates the VIX, thus we investigate the performance of autoregressive models, Extreme Gradient Boosting (XGBoost) models and a Local Linear Forest (LLF) model. The research utilizes data from January 2, 1990, to August 10, 2009, to develop point and directional forecasts. Results show that GARCH models perform poorly when only returns are considered, but accuracy improves with the inclusion of VIX data. Autoregressive models perform better due to high autocorrelation in the VIX series, while the Heterogeneous Autoregressive (HAR) model exhibits the lowest mean squared error, handling outliers effectively. The XGBoost model using short-, intermediate-, and long-term average VIX prices yields the lowest mean absolute prediction error of 4.77%, while the LLF model achieves the lowest mean absolute error and performs well based on the aforementioned performance measures. In predicting the direction of the VIX, the XGBoost model incorporating VIX moving averages achieves the highest accuracy at 55.38%. However, its performance does not demonstrate statistically significant improvement over a naive prediction model that consistently forecasts downward movement.

Supervisor:	Evgenii Vladimirov
Second assessor:	Bernhard van der Sluis
Date final version:	30th June 2024

1 Introduction

Forecasting stock market volatility is a topic of interest for many people in society. Investors want to know the level of uncertainty in the future they have to account for now to make profitable decisions whereas hedge funds can manage their risk exposure and implement strategies to mitigate potential losses. On another note, it would be almost impossible to price an option contract without having an impression of the future fluctuations of the underlying stock. In the literature, authors have agreed that volatility follows a stochastic process and popular models developed by Heston (1993) and Stein and Stein (1991) have been developed based on this assumption. With the recent advances in machine learning, a new world of models has opened. They often excel in predictive accuracy compared to the aforementioned traditional models. This research seeks to improve forecasts of volatility indices of the American stock market by considering the volatility implied by Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) option pricing models (Duan, 1995), Autoregressive Integrated Moving Average (ARIMA) models, the Heterogeneous Autoregressive (HAR) model by Corsi (2009), and the tree-based Extreme Gradient Boosting (XGBoost) model and Local Linear Forest (LLF) model.

The research on forecasting the American volatility index is of great importance due to its potential to enhance decision-making for investors and risk managers, thereby improving both financial stability and returns. This study addresses a significant gap in the existing literature by exploring the predictive power of machine learning models compared to traditional GARCH and autoregressive models. Ultimately, forecasting the volatility indices of different markets is thus not only of scientific relevance, but also has vital implications for policymakers, hedge funds, and investors. Hence, the problem statement of this paper is as follows: how do GARCH, autoregressive, and tree-based machine learning models compare in predicting the CBOE VIX and what is the impact of incorporating average prices on their predictive performance?

This paper also focuses on forecasting a volatility index instead of the volatility of a stock. It is important to note the difference between forecasting volatility and forecasting a volatility index. Forecasting volatility has been mainly important in the context of pricing derivatives, and options in particular. Poon and Granger (2003) provide a comprehensive review of the findings of 93 papers on volatility forecasting, where volatility is defined as the standard deviation of a set of observations, such as a stock. Although volatility can be used as a measure of risk, it also appears as an input for important option pricing formulas as seen in Black and Scholes (1973). Conversely, S. A. Degiannakis (2008) was the first to propose modelling the CBOE VIX—a volatility index that measures the constant 30-day expected volatility of the U.S. stock market—as a dependent variable, rather than using either interday conditional volatility or intraday realized volatility. The distinction between volatility and the volatility index is significant because the VIX encapsulates market expectations of the S&P 500's volatility, which can be influenced by investor sentiment and market conditions.

As mentioned before, the literature on volatility models is tightly connected to pricing options. Perhaps the most influential paper is by Black and Scholes (1973), where the theoretical valuation formula for options is derived. This paper opened the floor for more discussion on option pricing models, such as the class of stochastic volatility models. Wiggins (1987) models the volatility of returns as a stochastic process and provides a comparison to the Black-Scholes model

while testing its empirical robustness when not all assumptions hold while Heston (1993) derives a closed-form solution for the price of a European call option again under the assumption of stochastic volatility. Realized volatility models, such as the one by Andersen, Bollerslev, Diebold and Labys (2003), incorporate high–frequency intraday data into the measurement, modelling, and forecasting of volatility. However, not all stylized facts for asset returns were captured in these models. For example, the phenomenon of volatility clustering was not yet captured, nor was the negative correlation between stock returns and volatility. This is where GARCH models come into play. The GARCH models (Engle & Bollerslev, 1986) gained popularity due to their ability to capture the phenomenon of volatility clustering and the fat tails of the returns.

The first ideas about using GARCH models in the context of pricing options were proposed by Duan (1995). Given that an asset follows a GARCH process, a locally risk-neutral valuation relationship (LRNVR) can be developed to determine the price of an option on this asset. Heston and Nandi (2000) also took on this idea and captured both the stochastic nature of volatility and the correlation between volatility and spot returns. To connect the GARCH models to fitting the CBOE VIX, Hao and Zhang (2013) derive the formulas for the implied VIX of the GARCH models, but they concluded that the class of GARCH models fails to fit the VIX under the LRNVR. Soon after, Christoffersen, Feunou, Jacobs and Meddahi (2014) developed a new type of affine discrete-time model that allows for closed-form option valuation formulas using the conditional moment-generating function, which is a special case of the model developed by Heston and Nandi, and outperformed the GARCH model by capturing the volatility of variance. More recently, W. Zhang and Zhang (2020) found an adjustment of the LRNVR that was proposed by Duan (1995) which resulted in the GARCH option pricing model being able to capture the negative variance risk premium, which they failed to do in 2013.

Given the large growth in volatility trading, recent literature also focused on forecasting volatility as well as volatility indices instead of only assessing the fit. Ahoniemi (2008) models the CBOE VIX, whereafter forecasts are produced using an ARIMA(1,1,1) model including exogenous regressors. If one were to trade options based on the forecasts of this paper, a positive return would be obtained. However, similar to S. A. Degiannakis (2008), adding GARCH terms did not improve forecasts. Liu, Guo and Qiao (2015) suggest using GARCH(1,1), GJR, and Heston–Nandi models for this goal but also experienced that the VIX forecasts were again too low and unable to incorporate the variance risk premium. H. Wang (2019) adds VIX components to the HAR model (Corsi, 2009) to model the realized volatility of several stock markets resulting in an improvement in predictive performance over the AR(1) benchmark. Moving away from the model-based approaches, S. Degiannakis, Filis and Hassani (2018) was the first to apply the non-parametric Singular Spectrum Analysis in the context of forecasting volatility indices for multiple markets, including the EURO STOXX 50 volatility index (VSTOXX) for Europe. More recently, Wu, He and Xie (2023) established a promising future for the REGARCH-MIDAS-RA model for forecasting, as opposed to more traditional GARCH models.

With the recent popularity of machine learning, S. Wang, Li, Liu, Chen and Tang (2024) implemented several machine learning methods such as Extreme Gradient Boosting (XGBoost) and Neural Networks to forecast next-day returns of VIX constant-maturity futures. Similarly, Prasad, Bakhshi and Guha (2023) implemented several deep learning methods to predict the VIX

for India, resulting in highly accurate predictions. Taking into account that the computation methodology is the same for the India VIX as for the CBOE VIX, the tree-based methods proposed in this paper could have a promising future, which is also supported by Kleen and Tetereva (2022) who used Local Linear Forests (LLF) to forecast realized volatility for S&P 500 stocks.

The remainder of this paper will be as follows. Section 2 explains the source and structure of the data, and provides summary statistics. The models, estimation methods and performance measures are discussed in Section 3, along with an explanation of risk-neutral valuation and properties of the VIX as a time series. Results are presented in Section 4, along with a discussion on the model performance. Finally, Section 5 summarizes this paper's findings and presents suggestions for further research.

2 Data

To assess the goodness-of-fit and predictive performance of the volatility indices implied by the models, this paper compares them to measures for investor's expected volatility for the American market. In order, the prices of the S&P 500 index and the corresponding CBOE VIX index will be investigated, obtained respectively from the Wharton Research Data Services[1] and the CBOE website[2]. As a risk-free rate in the GARCH models, the daily 3-month U.S. Treasury Bills rate is used and is obtained from the Federal Reserve Website[3]. This research uses data from January 2, 1990, to August 10, 2009, for fitting the models in accordance with Hao and Zhang (2013), while forecasting is done on data from August 11, 2009, to August 10, 2010.

Summary statistics for the data are shown below. Note that for the S&P 500, the summary statistics for the log returns are given as the data only appears in this form in the GARCH option pricing models.

Variable	Mean	St.Dev.	Min	Max	Skewness	Kurtosis
VIX	20.22	8.41	9.31	80.86	7.24	0.12
S&P 500	0.0002	0.01	-0.09	0.11	-0.19	12.25
T-Bill	3.85	1.85	0	7.99	-0.24	-0.64

Table 1: Summary statistics for the data from January 2nd 1990 to August 10th 2009.

3 Methodology

This section discusses the concept of risk neutrality and the structure and estimation of the GARCH models. Furthermore, the VIX time series is inspected and the specification of the ARIMA, HAR, XGBoost, and Local Linear Forest (LLF) models is proposed.

[1]Wharton Research Data Services. (2024). WRDS. Retrieved May 23, 2024, from https://wrds.wharton.upenn.edu

[2]CBOE Exchange, Inc. (2024). Retrieved May 23, 2024, from https://www.cboe.com/tradable$_p roducts/vix/vix_h istorical_d ata/$

[3]S&P Dow Jones Indices LLC, S&P 500 [SP500], Retrieved May 23, 2024, from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/SP500

3.1 Locally risk-neutral valuation relationship

Duan (1995) derived the Locally Risk-Neutral Valuation Relationship (LRNVR), which changes the probability measure from a physical to a risk-neutral one. This allowed him to derive a formula for the implied VIX, which could be used in practice to model the CBOE VIX. This subsection discusses why this change in probability measures is needed and how it changes the GARCH models.

The conventional approach to pricing options often involves considering both probabilities: the risk-neutral probability measure $\mathbb{Q}$, under which investors behave as if they were risk-neutral, and the real-world probability measure $\mathbb{P}$, which describes the actual probabilities of events as perceived by investors with risk included. Considering the complex nature of the GARCH(p, q) processes, it is deemed necessary to introduce a specific condition related to variances under the risk-neutral measure. The LRNVR specifies that the one-period ahead conditional variance is invariant to a change in probability measure, which means that

$$Var^{\mathbb{P}}(\ln(\frac{X_t}{X_t - 1})|\phi_{t-1}) = Var^{\mathbb{Q}}(\ln(\frac{X_t}{X_t - 1})|\phi_{t-1}). \tag{1}$$

The LRNVR has several implications for the GARCH processes under measures $\mathbb{P}$ and $\mathbb{Q}$. First, under the physical measure, the returns are equal to

$$\ln \frac{X_t}{X_{t-1}} = r + \lambda\sqrt{h_t} - \frac{1}{2}h_t + \epsilon_t, \tag{2}$$

where

$$\epsilon_t|\phi_{t-1} \sim N(0, h_t),$$

and the volatility process is equal to

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i h_{t-i}, \tag{3}$$

while the LRNVR implies the returns under the risk-neutral measure follow

$$\ln \frac{X_t}{X_{t-1}} = r - \frac{1}{2}h_t + \xi_t, \tag{4}$$

where

$$\xi_t|\phi_{t-1} \sim N(0, h_t),$$

and the volatility process equals

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i (\xi_{t-i} - \lambda\sqrt{h_{t-i}})^2 + \sum_{i=1}^{p} \beta_i h_{t-i}, \tag{5}$$

where X_t is the price of the asset, r is the constant interest rate, λ is the risk premium, and ϕ_t is the information set of all information up to and including time t. Both ϵ_t and ξ_t are error terms but under different measures. One can see that the GARCH process remains largely intact concerning local risk neutralization. However, the conditional variance process under the risk-neutralized pricing measure, is not a GARCH process. To see this, we look at the

innovations ϵ_{t-i}^2 under the $\mathbb{P}$ measure and $(\xi_{t-i} - \lambda\sqrt{h_{t-i}})^2$ under the $\mathbb{Q}$ measure. Under the physical measure, the variance innovation is driven by q central chi-square random variables with one degree of freedom, whereas, in the GARCH process under the risk-neutral measure, the chi-square random variables are not central anymore. After factoring out $\sqrt{h_{t-i}}$ from the parentheses and recognizing that $\frac{\xi_{t-i}}{\sqrt{h_{t-i}}}$ follows a standard normal distribution, we find that the equity premium λ is the noncentrality parameter. Thus, while the risk is locally neutralized under the pricing measure $\mathbb{Q}$, the process driving the conditional variance is still influenced by the equity premium λ. Despite the presence of the equity premium under the LRNVR in the volatility process, Hao and Zhang (2013) argue that no premium for volatility risk is compensated under the LRNVR framework.

3.2 GARCH option pricing models

Under the assumption that the S&P 500 follows a Square-Root Stochastic Autoregressive Volatility process with one lag, abbreviated SR-SARV(1) (Meddahi & Renault, 2004), under measure $\mathbb{Q}$, we can derive closed-form formulas for the implied VIX for three different GARCH specifications. Namely, according to Hao and Zhang (2013), the implied VIX at time t is a linear function of the conditional variance of the next period. In this paper, we consider three cases of SR-SARV(1) processes: GARCH(1,1), AGARCH(1,1), and TGARCH(1,1). The latter two models capture the leverage effect, which suggests that large negative returns have a greater effect on future volatility than positive returns of the same magnitude. As opposed to the AGARCH model, the TGARCH model creates different regimes depending on the sign of the shock from the past period. In detail, they take the forms of the following:

GARCH(1,1):

$$\text{Physical measure: } h_t = \alpha_0 + \alpha_1\epsilon_{t-1}^2 + \beta_1 h_{t-1} \tag{6}$$

$$\text{LRNVR: } h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_1(\xi_{t-1} - \lambda\sqrt{h_{t-1}})^2 + \sum_{i=1}^{p} \beta_1 h_{t-1} \tag{7}$$

TGARCH(1,1)

$$\text{Physical measure: } h_t = \alpha_0 + \alpha_1\epsilon_{t-1}^2 + \theta\epsilon_{t-1}^2 1(\epsilon_{t-1} < 0) + \beta_1 h_{t-1} \tag{8}$$

$$\text{LRNVR: } h_t = \alpha_0 + (\xi_{t-1} - \lambda\sqrt{h_{t-1}})^2 + [\alpha_1 + \theta 1(\xi_{t-1} - \lambda\sqrt{h_{t-1}} < 0] + \beta_1 h_{t-1} \tag{9}$$

AGARCH(1,1)

$$\text{Physical measure: } h_t = \alpha_0 + \alpha_1(\epsilon_{t-1} - \theta\sqrt{h_{t-1}})^2 + \beta_1 h_{t-1} \tag{10}$$

$$\text{LRNVR: } h_t = \alpha_0 + \alpha_1(\xi_{t-1} - \lambda\sqrt{h_{t-1}} - \theta\sqrt{h_{t-1}})^2 + \beta_1 h_{t-1} \tag{11}$$

3.3 Estimation

As in the paper by Hao and Zhang (2013), the GARCH models are estimated using the method of Maximum Likelihood. However, the authors do not specify the initial parameters, nor the

explicit optimization method they use to maximize the likelihood function. Therefore, the starting parameters used in this paper can be found in section A.2 in the Appendix, and estimation was done by the interior point algorithm. There will be three different likelihood functions that need to be maximized: one using the S&P return dataset only, one using the VIX dataset only, and one using both. For the estimation using only returns, the estimation is done under the physical measure, resulting in the log-likelihood function

$$\ln(L_R) = \frac{-T}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\{\ln(h_t) + [\ln(X_t/X_{t-1}) - r - \lambda\sqrt{h_t} + \frac{1}{2}h_t]^2/h_t\}, \tag{12}$$

where T is the total number of time steps and h_t corresponds to the GARCH, AGARCH or TGARCH process under the physical measure.
Similarly, the log-likelihood corresponding to the CBOE VIX only using the risk-neutral measure is equal to

$$\ln(L_V) = \frac{-T}{2}\ln(2\pi\hat{s}^2) - \frac{1}{2\hat{s}^2}\sum_{t=1}^{T}(VIX_t^{Mkt} - VIX_t^{Imp})^2, \tag{13}$$

where $\hat{s}^2$ denotes the variance of the difference between the market VIX and the implied VIX. When using joint parameter estimation, the log-likelihood function becomes

$$\ln(L_T) = \ln(L_R) + \ln(L_V), \tag{14}$$

which we use when we estimate the GARCH models with both returns and VIX data. During the estimation procedure, we have the following stationary conditions for the parameters: for the GARCH(1,1) process

$$\alpha_1(1+\lambda^2) + \beta_1 < 1,$$

for the TGARCH(1,1) process

$$\alpha_1(1+\lambda^2) + \beta_1 + \theta[\frac{\lambda}{\sqrt{2\pi}}e^{-\frac{\lambda^2}{2}} + (1+\lambda^2)N(\lambda)] < 1,$$

and for the AGARCH(1,1) process

$$\alpha_1(1+(\lambda+\theta)^2) + \beta_1 < 1.$$

Note that even when we maximize the likelihood under the physical measure, the stricter stationary conditions under the risk-neutral measure are used.

3.4 GARCH Implied VIX

The CBOE VIX reflects investors' expectation of S&P 500's volatility over the next 30 calendar days (or 21 trading days), which yields the following formula:

$$\left(\frac{VIX_t}{100}\right)^2 = E_t^Q\left[\frac{1}{\tau_0}\int_t^{t+\tau_0}\tilde{h_s}ds\right], \tag{15}$$

where $\tau_0 = 30$ calendar days or 21 trading days. This paper follows the procedure of calculating the VIX as the mean of the variance in the next n subperiods of the following 30 calendar days (or 21 trading days) as in Hao and Zhang (2013), which means for data with daily frequency that $\tau_0 = n = 21$ and hence

$$VIX_t = \frac{1}{n}\sum_{k=1}^{n} E_t^Q[h_{t+k}], \tag{16}$$

with h_{t+k} for the corresponding SR-SARV(1) process under the risk-neutral measure. After the estimation of the models, we get the following VIX formulas:

$$VIX_t = A + Bh_{t+1} \tag{17}$$

$$A = \frac{\alpha_0}{1-\eta}(1-B),$$

$$B = \frac{1-\eta^n}{n(1-\eta)},$$

where for the GARCH model

$$\eta = \alpha_1(1+\lambda^2) + \beta_1,$$

for the TGARCH model

$$\eta = \alpha_1(1+\lambda^2) + \beta_1 + \theta S,$$

where $S = [\frac{\lambda}{\sqrt{2\pi}} e^{-\frac{\lambda^2}{2}} + (1+\lambda^2)N(\lambda)]$ if $u_t = \xi_t/\sqrt{h_t}$ follows i.i.d. standard normal.
Finally, for the AGARCH model

$$\eta = \alpha_1[1 + (\lambda+\theta)^2] + \beta_1.$$

Essentially, the outcomes of equation 17 are compared to the CBOE VIX.

3.5 ARIMA models

A popular method, among others used by Ahoniemi (2006), to fit and forecast the VIX index is using Autoregressive Integrated Moving Average (ARIMA) models. Including autoregressive terms, moving average terms, and differencing operations, this model creates a linear equation to forecast future values of a time series. The ARIMA(p, d, q) model is defined as follows by Kotu and Deshpande (2018):

$$y_t = I + \mu + \alpha_1 y_{t-1} + ... + \alpha_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}, \tag{18}$$

where y_t is the value of the VIX index at time t, μ is a constant, and ϵ_t is the error term at time t. I indicates that the data has been differenced d times in order to obtain stationary data, which is needed for accurate modelling. The idea of this model is based on serial correlation: the autoregressive part accounts for the autocorrelation of order p in the time series, while the moving average part models the dependence on the error of the past q data points.

3.5.1 Time series inspection

This subsection discusses some properties of the VIX time series along with implications for the ARIMA model. Using this information, one can first guess what ARIMA models would be appropriate for modelling the CBOE VIX. To start the analysis, we plot the time series of the CBOE VIX index from January 2nd 1990 to August 10th 2009 in Figure 1, its differences in Figure 2 along with the autocorrelation function (ACF) and the partial autocorrelation function (PACF) in Figure 3 and Figure 4.

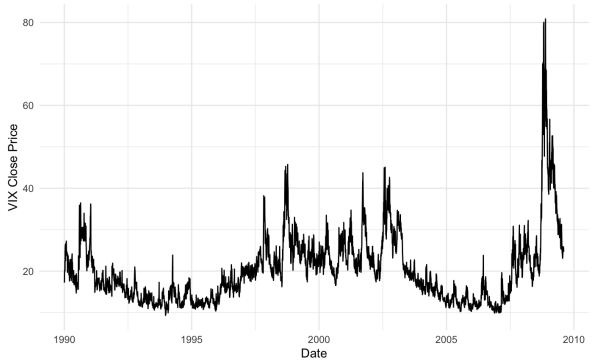


Figure 1: Closing prices of the VIX from January 2nd 1990 to August 10th 2009.

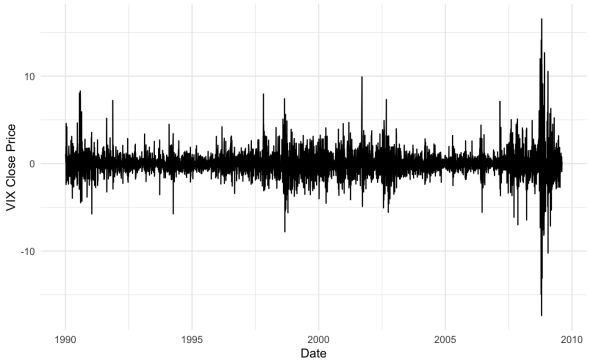


Figure 2: Difference of the VIX from January 2nd 1990 to August 10th 2009.

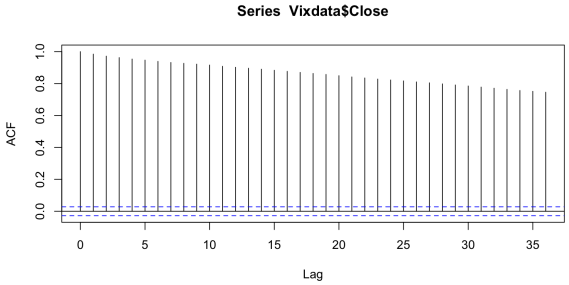


Figure 3: ACF of the VIX data.

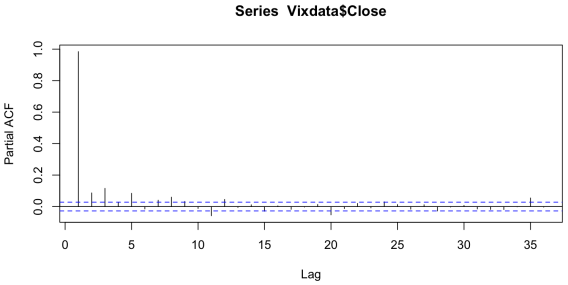


Figure 4: PACF of the VIX data.

Visual inspection of the time series shows that the conditional volatility is not constant, but changes over time. Especially in 2008, the VIX was at the highest level it had been to date, which can be attributed to the global financial crisis. A quick look at the ACF shows that the time series exhibits long-term dependency. Furthermore, the Augmented Dickey-Fuller (ADF) test is performed to investigate if this time series is stationary. The null hypothesis is that there is a unit root present in the data, which means that at least one α is equal to 1. The test statistic is -4.389 with a p-value smaller than 0.01, so the null hypothesis is rejected. The same conclusion is drawn from an ADF test on the differenced series. Hence, stationarity cannot yet be rejected which corresponds to the result obtained in Saha, Malkiel and Rinaudo (2019).

To fit an appropriate ARIMA model, an algorithm by Hyndman and Athanasopoulos (2018) is implemented in R using the auto.arima function from the 'forecast' package made by Hyndman and Khandakar (2008). This algorithm evaluates different ARIMA(p, d, q)specifications and selects the one with the lowest Akaike Information Criterion (AIC) as the best. The exact algorithm can be found in Section A.6 in the Appendix. The result of the Hyndman-Khandakar

Algorithm is an ARIMA$(3, 1, 3)$ model, where (as visible from $d = 1$) first-differencing is applied. Although the level data is stationary, differencing can lead to more parsimonious models by reducing the need for high-order AR or MA terms as would be needed in the levels model regarding the slowly decreasing ACF.

3.6 Benchmarks

This research considers two benchmarks. To begin with, an ARIMA(1,0,0) (which simplifies to an AR(1) model) is estimated for comparison with other models. It is described as

$$y_t = \mu + \alpha_1(y_{t-1} - \mu), \tag{19}$$

where μ is the constant mean. This approach is also taken by H. Wang (2019), who describes it as the no-predictability benchmark.

Moreover, we view the Heterogeneous Autoregressive (HAR) model of Realized Volatility (RV) by (Corsi, 2009) with the following specification:

$$RV_t = c + \alpha_1 RV_{t-1} + \alpha_2 RV_{t-1,5} + \alpha_3 RV_{t-1,22} + \epsilon_t, \tag{20}$$

where $RV_{t-1,L} = \frac{1}{L}\sum_{j=1}^{L} RV_{t-j} (L = 5, 22)$. It can be argued that this model for realized volatility is also applicable to model the VIX directly. Namely, the HAR model is designed to capture the heterogeneous nature of volatility over different time scales: the first lag represents the daily scale, where $L = 5$, and $L = 22$ represent weekly and monthly time periods respectively. The VIX, being a measure of expected market volatility, inherently reflects the market's view on volatility over different horizons. Therefore, we use the following variant of the HAR model for the VIX:

$$VIX_t = c + \beta_1 VIX_{t-1} + \beta_2 \frac{1}{5}\sum_{i=1}^{5} VIX_{t-i} + \beta_3 \frac{1}{22}\sum_{i=1}^{22} VIX_{t-i} + \epsilon_t. \tag{21}$$

3.7 Tree-based methods

3.7.1 Decision trees

A decision tree is a flowchart-like structure where each internal node represents a decision based on the value of a feature, each branch represents the outcome of the decision, and each leaf node represents a final prediction or outcome. Decision trees are used for both classification and regression tasks. They partition the data into subsets based on the feature values, making the decision-making process interpretable and easy to visualize. A visualization of a decision tree and its nodes can be found in Figure 5. However, one single decision tree can be prone to overfitting, which means the algorithm fits very closely to the training data but

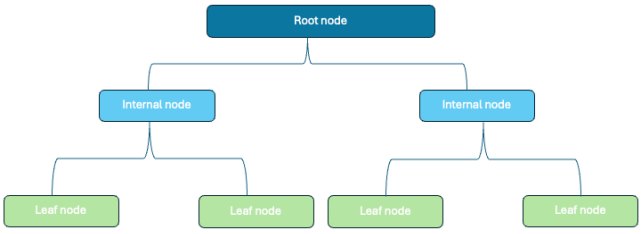


Figure 5: Visualization of a decision tree.

cannot generalize to a new dataset. Therefore, it may not always provide the best predictive performance. The solution will be explained next.

3.7.2 XGBoost

Extreme Gradient Boosting, abbreviated XGBoost (Chen & Guestrin, 2016), is a tree-based method which implements boosting, a machine learning technique for regression and classification problems. It builds models sequentially, where each new model attempts to correct the errors of the previous models. As opposed to other tree-based methods, XGBoost is efficient in the sense that it has lower running times.

Boosting combines weak learners, such as decision trees, sequentially to create one good model. Figure 6 visualizes the gradient boosting algorithm. First, the algorithm builds one decision tree based on the input, which in the case of this research will be the VIX index and possibly some macroeconomic time series. Then, the second tree when added to the first should minimize the loss, which we choose to be minimization of the squared differences between the predicted values and the actual values, effectively reducing the Mean Squared Error (MSE). This is done while finding the direction in which the loss function declines the fastest. Now, the first two models are combined into one model with a lower MSE than the initial model by taking the averages of the two trees. This can be done for a finite number of iterations until residuals have been minimized as much as possible. The resulting model will then be used to make predictions. XGBoost has no problem handling seasonality and trends in time-series data. The algorithm is implemented in R using the 'xgboost' package by Chen et al. (2024).

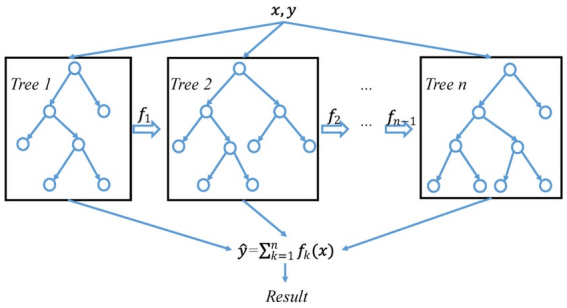


Figure 6: A general architecture of XGBoost (Y. Wang et al. (2019)).

3.7.3 Local linear forests

Local Linear Forests (LLF), as described by Friedberg, Tibshirani, Athey and Wager (2020), combine the predictive power of decision trees and local regression. Essentially, they use a random forest, which combines the results of multiple decision trees into one prediction, to generate weights that can serve as a kernel for local linear regression, which results in an estimation of the form

$$\begin{pmatrix} \mu(x_0) \\ \hat{\theta}(x_0) \end{pmatrix} = \text{argmin}_{\mu,\theta} \left\{ \sum_{i=1}^{n} \alpha_i(x_0) \left(Y_i - \mu(x_0) - (x_i - x_0)\,\theta(x_0)\right)^2 + \lambda \left\|\theta(x_0)\right\|_2^2 \right\}, \tag{22}$$

where x_0 is the input location for prediction, x_i is a vector of features for point i, Y_i is the outcome, $\mu(x_0)$ is the local average, $\theta(x_0)$ is the slope, α_i is the weight obtained from the random forest, and λ is the ridge parameter used for regularization. Considering the promising literature on the (linear) HAR model, and the fact that the effect of the previous-day VIX value can be different in times of a recession and an expansion, the LLF model could be an appropriate

fit. In addition, Kleen and Tetereva (2022) recently applied the local linear forest in realized volatility forecasting for 186 S&P 500 stocks using HAR models at the leaves, which resulted in superior forecasting performance. The method is implemented in R using the 'grf' package by Tibshirani, Athey, Sverdrup and Wager (2024).

3.7.4 Tuning and fitting

The XGBoost and LLF model require several hyperparameters that need to be tuned. For training and tuning, the data from January 2, 1990, to August 10, 2009, will be used following Hao and Zhang (2013). Testing (or forecasting) will be done using the data from August 11, 2009, to August 10, 2010 as in the other models. The hyperparameters are tuned on the training set by time series cross-validation. This means creating several folds within the training set, where each fold is again split into a train set and a test set straight after each training window. A visualization of the procedure for 3 folds can be found in Figure 7, where the data in green would be the data from August 11, 2009 onwards. This paper uses a separation of the training data into 5 folds. Consequently, for each fold the model is trained using the data in the fold's training window, and its performance is evaluated using the data in the fold's testing window. Hence, for one set of hyperparameters we evaluate the performance on 5 the test sets indicated in light blue in Figure 7. This procedure is repeated for a grid of hyperparameters, for which we want to find the ones that yield the lowest Root Mean Squared Errors (RMSE) in the fold's test sets. Using this approach, the model can get accustomed to data it has not seen before, thus reducing the risk of overfitting. The parameters that need to be tuned, including a definition, their values in the tuning grid, and the optimal value are shown in Table 9 in Section A.4 in the Appendix.

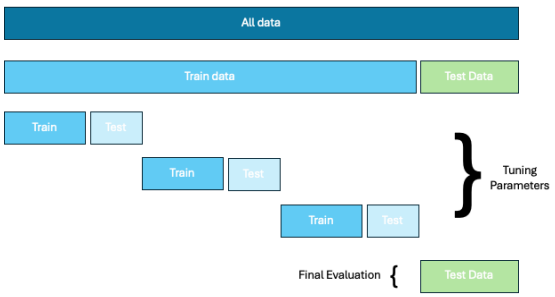


Figure 7: Cross validation on time series data.

After the hyperparameters have been tuned using the 5 folds of the data from January 1990 to August 2009, it is time to fit the models to this same part of the data, which corresponds to the entire training period. For the XGBoost model, this is done using two different approaches, and thus using two different sets of features the model has to use for making predictions. The first model is built upon the previous value of VIX itself. For the first point forecast, we use the trained model to make a one-step-ahead prediction. After that, the forecast is saved and the true value of the VIX for that day is appended to the training set. Thus, for the second day, a forecast is made using the true VIX closing price of the first day, without retraining the XGBoost model. This is repeated for one year consisting of 252 trading days. The other approach is inspired by Y. Zhang (2022). The author uses the 5-day, 15-day, and 30-day average stock price, modelled again using a moving window, as predictors for a set of stock prices. In contrast, where the author uses stock prices, this paper uses the VIX and the same average prices as in the HAR model, namely the 1-day, 5-day, and 22-day average VIX. Considering short, intermediate, and long trends altogether, these features summarize the

overall sentiment and trends in volatility that the model needs to capture. The phenomenon of volatility clustering, where periods of large volatility tend to also be followed by periods of large volatility, will also be captured using these lag effects by averaging past values, providing the model with a way to account for them without explicitly lagging the series multiple times. This method will be defined as the Rolling VIX Average Price (RVAP) for the remainder of this paper. The LLF model will only be estimated using this method.

3.8 Forecasting

To assess the model performance, the data is split into two subsets; a training set and a testing set as described in Section 3.7.4. To assess the predictive performances of the models, this research averages the performance measures of each model for each testing set. As performance measures, the Mean Error (ME), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Prediction Error (MAPE) will be used. All these metrics serve a different purpose. The mean error can identify whether the model systematically over- or underpredicts the VIX, while the mean absolute error looks at the average size of the errors. Furthermore, a high mean squared error indicates that the model is unsuited for handling outliers and the mean absolute percentage prediction error provides an intuitive measure of the average difference between the forecasts and the true values. The measures are calculated as follows.

$$ME = \frac{\sum_{i=1}^{n} y_i - \hat{y}_i}{n},$$

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n},$$

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n},$$

$$MAPE = \sum_{i=1}^{n} \frac{1}{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| * 100\%,$$

where y_i is the observed VIX, $\hat{y}_i$ is the prediction of the VIX and the VIX implied by the model in the case of GARCH. Finally, n is the size of the test set.

4 Results

4.1 GARCH models

This section discusses the fit of the GARCH models estimated on the VIX time series. Below, Table 2 shows the Maximum Likelihood estimates of the parameters, along with the standard errors in parentheses. This is a replication of the results from Hao and Zhang (2013). Note that these values depend on the optimization algorithm and the selected initial parameters, hence they differ (only very slightly) from the ones presented in Hao and Zhang (2013) even though the same method was implemented. The initial parameters for this research are presented in the Appendix. Also following their paper, the AGARCH model using VIX only was not estimated considering λ and θ play the same role. Hence, the model cannot be identified and comes down to the regular GARCH model.

Model & data	α_0	α_1	β_1	θ	λ
GARCH					
Returns	7.0633e-7	0.0635	0.9313	-	0.0529
	(1.7032e-7)	(0.0070)	(0.0074)		(0.0135)
VIX	1.7189e-6	0.0367	0.9388	-	0.7888
	(0.03823e-6)	(0.0008)	(0.0015)		(0.0297)
Both	1.6871e-6	0.0471	0.9499	-	0.2071
	(0.0402e-6)	(0.0010)	(0.0011)		(0.0122)
TGARCH					
Returns	1.0899e-6	0.0016	0.9322	0.1094	0.0234
	(0.1783e-6)	(0.0053)	(0.0071)	(0.0116)	(0.0142)
VIX	1.6118e-6	0.0022	0.9530	0.0456	0.4204
	(0.0447e-6)	(0.0025)	(0.0018)	(0.0024)	(0.0380)
Both	1.5225e-6	0.0038	0.9599	0.0613	0.0777
	(0.0416e-6)	(0.0016)	(0.0010)	(0.0023)	(0.0128)
AGARCH					
Returns	1.1338e-6	0.0554	0.8799	1.0127	0.0154
	(0.1725e-6)	(0.0055)	(0.0105)	(0.0937)	(0.0143)
Both	1.7203e-6	0.0393	0.9345	0.7738	0.0162
	(0.0469e-6)	(0.0010)	(0.0016)	(0.0308)	(0.0142)

Table 2: Maximum Likelihood estimates of GARCH models using returns, VIX or both. In parentheses are standard errors.

In line with the findings of Hao and Zhang (2013), the most interesting aspect of Table 2 is that the equity risk premium parameter λ increases when VIX is included (either alone or with returns) in the estimation procedure. To illustrate the impact, for the TGARCH model the equity risk premium increases from 0.0234 to 0.4204 when only VIX is used, which is almost 18 times as large. This result suggests that incorporating VIX, which shows the market's expectation of volatility, leads to investors requiring a higher compensation for taking on the risk of investing. Another notable finding is the rather large persistence parameter β_1, and the corresponding low values of α_0 induced by the stationary condition constraints presented in Section 3.3. On one hand, the high value suggests that shocks in volatility have a long-lasting effect while the increasing value of β_1 when VIX is also considered raises the long-run variance of the GARCH processes under the risk-neutral measure.

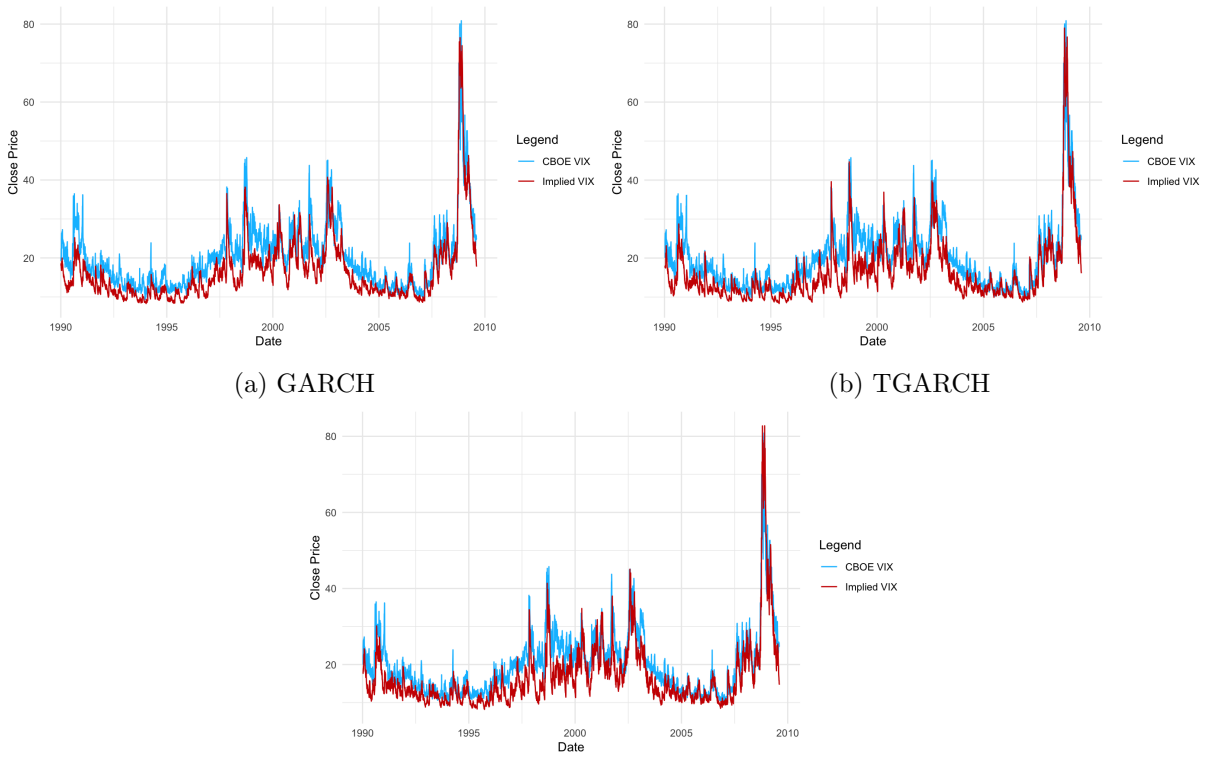
Now that the parameters are estimated, the implied VIX of the GARCH models can be calculated using Equation 17. Table 3 shows how the implied VIX fits the CBOE VIX in levels for the three GARCH models investigated from January 2, 1990, to August 8, 2009.

Table 3: GARCH model fit of the VIX levels.

Model & data	ME	Std.Err.	MAE	MSE	RMSE	P-value
GARCH						
Returns	3.58	3.32	4.00	23.85	4.88	0.0000
VIX	0.11	3.08	2.36	9.49	3.08	0.4904
Both	0.2625	3.22	2.39	10.45	3.23	0.1162
TGARCH						
Returns	3.76	3.25	4.06	24.71	4.97	0.0000
VIX	0.10	3.05	2.33	9.33	3.05	0.5332
Both	0.27	3.08	2.31	9.56	3.09	0.1060
AGARCH						
Returns	3.46	3.22	3.79	22.33	4.73	0.0000
Both	0.26	3.08	2.34	9.54	3.09	0.1141

This table shows how the implied VIX fits the CBOE VIX in levels for the three GARCH models investigated during the period from January 2, 1990 to August 8, 2009. The error is calculated as the CBOE VIX minus the implied VIX. The mean error (ME) calculates the daily average error between the implied VIX and the CBOE VIX. The standard error (Std.Err.) calculates the standard deviation of the error. The mean absolute error (MAE) calculates the daily average absolute error between the implied VIX and the CBOE VIX. The mean squared error (MSE) calculates the daily average squared error between the implied VIX and the CBOE VIX. The root mean squared error (RMSE) calculates the square root of the mean squared error. The p-value is for the null hypothesis that the means of the implied VIX and the CBOE VIX are equal.

When only returns are regarded, the mean error calculated as the difference between the CBOE VIX and the implied VIX is high, but decreases when VIX is included too. Also taking into account the p-values, we can thus conclude that the implied VIX is significantly lower than the CBOE VIX for all three GARCH models when only returns are considered. Hao and Zhang (2013) acknowledge this to the GARCH models being unable to capture the variance premium. In contrast, when VIX data is considered the performance becomes significantly better, but Hao and Zhang (2013) argue that the parameters are distorted to match the levels of the VIX. Their more elaborate results show that the statistical properties of the VIX are still not captured well by the GARCH models. Furthermore, the AGARCH model demonstrates superior performance across all evaluation metrics while also showing a small standard error. This shows that the AGARCH model accurately captures the leverage effect as opposed to the GARCH model, and the improvement over the TGARCH model could be attributed to its flexibility as opposed to the regime-switching nature of the TGARCH model.



(a) GARCH

(b) TGARCH

(c) AGARCH

Figure 8: Comparison between the implied VIX and the CBOE VIX (estimated with returns).

4.2 XGBoost, ARIMA, and HAR models

This section discusses the fit of the XGBoost, ARIMA, and HAR models on the VIX time series. Table 4 shows the parameter estimates for the AR(1), ARIMA(3,1,3), and HAR models.

Model	μ	α_1	α_2	α_3	θ_1	θ_2	θ_3
AR(1)	20.2157	0.9843	-	-	-	-	-
	(1.3238)	(0.0025)	-	-	-	-	-
ARIMA(3,1,3)	-	-1.0997	0.1734	0.5391	0.9896	-0.4265	-0.6759
	-	(0.0643)	(0.0706)	(0.0491)	(0.0595)	(0.0588)	(0.0480)
	c	β_1	β_2	β_3			
HAR	0.2066	0.8387	0.1283	0.0229	-	-	-
	(0.0558)	(0.0146)	(0.0190)	(0.0102)	-	-	-

Table 4: Estimates of the AR(1), ARIMA(3,1,3), and HAR models. In parentheses are standard errors.

We discuss a few noteworthy findings. To begin with, for the ARIMA models, we see that the α_1 coefficient is very close to one, indicating a high persistence in the VIX time series as confirmed by the slowly decreasing autocorrelation function shown in Figure 3. Specifically, the current value of the series is almost the same as the previous value, with only a small amount of random variation. Second, the absence of an intercept for the ARIMA(3,1,3) model can be explained by the differencing of the data indicated by $d = 1$ in the ARIMA(p, d, q) specification. Differencing

the data removes trends and makes the series stationary, which often results in a mean closer to zero thereby reducing the need for a high intercept to fit the model. It is also interesting that the α_1 coefficient for this model is negative. This indicates that a positive change in one period might be followed by a negative change in the next for the differenced time series. This typically relates to the concept of mean reversion, which suggests that asset prices will eventually return to their long-term mean or average. While the VIX is not a stock, this phenomenon is also believed to occur for volatility as shown by Fouque, Papanicolaou and Sircar (2000). The results of the hyperparameter tuning for the XGBoost model and LLF model can be found in Table 9 in the Appendix. Table 5 shows the fit of the models to the CBOE VIX with the same performance measures as Table 3.

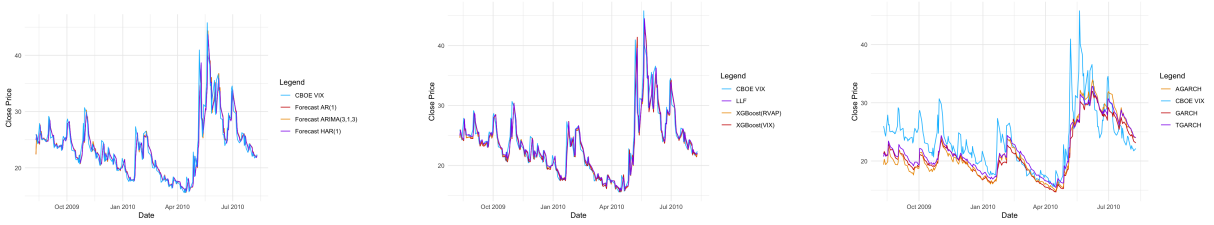
Model & data	ME	Std.Err.	MAE	MSE	RMSE	P-value
AR(1)	0.0014	0.02	0.92	2.18	1.48	0.9931
ARIMA (3,1,3)	0.0024	0.02	0.91	2.11	1.45	0.9886
HAR	-2.35e-16	0.02	0.91	2.12	1.46	1
XGBoost (RVAP)	0.10	0.01	0.28	0.34	0.58	0.5322
XGBoost (VIX)	0.12	0.007	0.23	0.24	0.49	0.4678
LLF	0.007	0.02	0.88	1.98	1.41	0.9681

Table 5: Autoregressive and XGBoost model fit of the VIX levels.

We see that all models show no significant difference in mean from the CBOE VIX as shown by the p-value column. Furthermore, the (partially) linear models show a MAE from 0.88 to 0.92, while it is much lower for the XGBoost models.

4.3 Forecasting

This section discusses the predictive performance of the models. Table 6 shows the out-of-sample performance for the period 11 August 2009 to 10 August 2010, which is equal to 252 trading days. In contrast, Figure 9 shows the VIX forecasts for the new period.



(a) Autoregressive models (b) XGBoost models (c) GARCH models (both returns and VIX)

Figure 9: Forecasts of the models from August 11, 2009, to August 10, 2010.

Model & data	ME	MSE	MAE	MAPE	St.Err.
GARCH					
Returns	5.59	40.79	5.67	24.47%	3.10
VIX	1.97	14.99	2.98	13.78%	3.34
Both	2.35	15.46	2.98	14.04%	3.16
TGARCH					
Returns	4.89	33.66	4.98	29.98%	3.13
VIX	1.61	13.26	2.74	12.29%	3.27
Both	1.58	12.56	2.63	11.77%	3.18
AGARCH					
Returns	5.02	38.32	5.36	33.92%	3.63
Both	2.10	15.61	3.09	14.54%	3.36
XGBoost					
RVAP	0.15	4.25	1.24	4.77%	0.13
VIX	0.18	4.29	1.25	4.83%	0.13
LLF					
RVAP	0.01	4.14	1.24	4.84%	0.13
ARIMA(3,1,3)					
VIX	-0.02	4.16	1.26	4.92%	0.13
AR(1)					
VIX	0.04	4.17	1.25	4.85%	0.13
HAR					
VIX	0.02	4.12	1.25	4.85%	0.13

Table 6: Performance measures for 252 days forecasts.

In line with Table 3, even in terms of forecasting the GARCH class models perform the worst when only returns are included, with mean errors of 5.59, 4.89 and 5.02 for the GARCH, TGARCH, and AGARCH models respectively. However, when VIX is included the performance improves, and the mean errors halve. In contrast, the forecasting performance of the TGARCH model when both returns and VIX are considered is better than the model when only VIX is considered, but only slightly: the MSE differ by only 0.7, which is not much. It can be concluded that the GARCH models have little predictive power, which was to be expected considering their fit. Again, the true VIX is undervalued, and the variance premium is thus not accurately captured. This can also be seen in Figure 9 for the GARCH models with both returns and VIX.

Moving towards the other models, we observe that the ARIMA(3,1,3) model and the AR model show similar performance. This can be attributed to the high autocorrelation in the time series, as established in Figure 3. Even simple models that rely heavily on past values can perform quite well when future values are strongly correlated with past values. This also reduces the performance gap between simple autoregressive models and complex machine learning models: the XGBoost models' MAE and MAPE are lower or equal to the MAE of the autoregressive models, especially the XGBoost RVAP model yields the lowest MAPE of 4.77%. However, they

are worse at handling outliers as can be seen from the high MSE: the squaring of the errors emphasizes the larger discrepancies, so a lower MSE suggests fewer large errors. However, the LLF model also yields the lowest MAE and its MSE is lower than the other tree-based models, suggesting the linearity in the leaf nodes of the trees could be an improvement. The HAR model, originally meant to model realized volatility, is certainly promising too considering it has the lowest MSE of 4.12.

To assess if there is a significant difference in predictive performance between the (of the benchmarks best performing) HAR model and the other models, we perform Diebold-Mariano tests for the HAR model against all other models. The null hypothesis is that both predictions have the same accuracy, while the alternative hypothesis specifies that the HAR model is more accurate than the other. Table 8 in the Appendix shows that the HAR model makes significantly better predictions than the three GARCH models while the p-values for the other models do not suggest rejection of the null hypothesis.

In conclusion, models using VIX data including both short-, intermediate-, and long-term averages tend to perform best in forecasting the VIX, although there is no significant difference between the benchmark HAR model and the machine learning models. The GARCH models are not suited for forecasting and are significantly worse than the HAR model. The XGBoost models have lower or equal MAE compared to autoregressive models but struggle with outliers. In contrast, the LLF model also yields the lowest MAE among tree-based models while its ability to handle outliers is better.

4.4 Directional Forecasting

Considering the similar predictive performance of the models, it is also interesting to analyze their performance in forecasting the direction of the VIX. The GARCH models will not be considered in this section as they have little predictive performance. The evaluation is done by creating the confusion matrix consisting of true positives (down,down), true negatives (up,up), false positives (down,up), and false negatives (up,down) and calculating several performance measures. The outcomes can be seen below in Figure 10 and Table 7.

Prediction	**Down**	**Up**
Down	96	62
Up	53	40

(a) XGBoost (VIX)

Prediction	**Down**	**Up**
Down	97	60
Up	52	42

(b) XGBoost (RVAP)

Prediction	**Down**	**Up**
Down	94	59
Up	55	43

(c) HAR

Prediction	**Down**	**Up**
Down	92	57
Up	57	45

(d) AR(1)

Prediction	**Down**	**Up**
Down	88	56
Up	61	46

(e) ARIMA(3,1,3)

Prediction	**Down**	**Up**
Down	93	58
Up	56	44

(f) Local Linear Forest

Figure 10: Confusion matrices for different models.

Table 7: Performance measures for the confusion matrices.

Model	Accuracy	P-value [Acc >NIR]	Sensitivity	Specificity
AR(1)	0.5458	0.9453	0.6174	0.4412
ARIMA(3,1,3)	0.5339	0.9763	0.5906	0.4510
HAR	0.5458	0.9453	0.6309	0.4216
XGBoost (VIX)	0.5498	0.9297	0.6242	0.4412
XGBoost (RVAP)	0.5538	0.9109	0.6510	0.4118
LLF	0.5458	0.9453	0.6242	0.4314

The accuracy is measured as the sum of true positives (down,down) and true negatives (up,up) divided by the total number of forecasts. The p-value is of the null hypothesis that the accuracy and the No Information Rate (NIR), taken to be the largest class percentage in the data, are equal. The sensitivity is calculated as $\frac{TruePositive}{TruePositive+FalseNegative}$ and the specificity as $\frac{TrueNegative}{TrueNegative+FalsePositive}$.

As seen from Table 7, the XGBoost model with the moving averages of short-. intermediate-, and long-term VIX values has the highest accuracy and predicts the direction accurately more than 55% of the time. However, the p-value suggests that the model is not significantly better than the "kitchen sink" approach of always predicting the largest percentage class, which, in this case, is the VIX going down. After the XGBoost model with only the lagged VIX, The AR(1), HAR, and LLF models come in third and achieve an accuracy of 54.58%. Moreover, the ability to correctly predict downward movements of the VIX (measured by sensitivity) is better than the ability to predict upward movements, which is indicated by the higher sensitivity for all models. This could be attributed to the fact that there are more downward movements than upward movements in the time series considering "down" is the largest percentage class.

5 Conclusion

This paper investigated various models for estimating and forecasting the CBOE VIX index, focusing on the performance of GARCH models under the Locally Risk-Neutral Valuation Relationship (LRNVR) and alternative autoregressive models and tree-based machine learning models. The inclusion of short-, intermediate-, and long-term average levels of the VIX was also examined by implementing a HAR model and including the terms as regressors for the XGBoost and LLF models instead of the lagged VIX. After estimation and hyperparameter tuning based on data from January 2, 1990, to August 10, 2009, we used the models to make both point and directional forecasts for 1 step ahead for 252 consecutive trading days.

We observed that the GARCH models have different specifications under the physical and risk-neutral measure, and accounting for this difference is crucial in the estimation procedure with S&P 500 returns, VIX, or both time series. By utilizing Maximum Likelihood estimation, we derived formulas for the VIX implied by the GARCH models.

Although the VIX time series is non-stationary, we estimated an AR(1) model and obtained an ARIMA(3,1,3) model using the Hyndman-Khandakar algorithm as benchmarks along with the HAR model with moving average VIX levels for 1, 5, and 22 trading days.

As mentioned before, this research also considers two tree-based methods that can capture

non-linear relationships. The XGBoost model is estimated for both the lagged VIX time series and employs the same moving average approach as the HAR model, while the LLF model will only use the latter.

In line with the findings of Hao and Zhang (2013), numerical results for the three GARCH specifications reveal that including VIX in the estimation process significantly increases the equity risk premium parameter λ. Furthermore, the long-lasting effects of volatility shocks are highlighted by the magnitude persistence parameter and it is found that the inclusion of the VIX time series in the estimation process results in the means of the implied VIX and CBOE being equal as opposed to when only returns are considered.

Out-of-sample forecasts made for the period August 11, 2009, to August 10, 2010 show that GARCH models perform poorly when only returns are considered, but their accuracy improves substantially with the inclusion of VIX data. On the other hand, the autoregressive models perform much better due to the high autocorrelation in the VIX series, which also reduces the performance gap with the more complex machine learning models. Among the models evaluated, the HAR model exhibits the lowest mean squared error, suggesting superior handling of outliers and large discrepancies. The LLF model comes close in terms of MSE and obtains the lowest MAE, while the XGBoost model with short-, intermediate-, and long-term moving averages yields the lowest mean absolute percentage prediction error. However, a Diebold-Mariano test indicated the forecasts of the HAR model and these other models are not significantly different.

Finally, in forecasting VIX direction, the XGBoost model including VIX moving averages achieves the highest accuracy of 55.38% though its performance is not statistically significantly better than a naive prediction model that always predicts downward movement. The AR(1), HAR, and LLF models show an accuracy of 54.58%, while all models have higher sensitivity for predicting VIX declines, indicating a general proficiency in forecasting downward movements.

A limitation of this research lies in the parameter tuning of the machine learning models. This was achieved using a 5-fold time series cross-validation. However, despite this method, there is still a considerable amount of randomness involved in estimating the models, and thus its performance is difficult to evaluate. In addition, a larger grid of hyperparameters could have been considered. Moreover, in addition to directional and 1-day ahead forecasting, we could have also evaluated different time periods or multi-day ahead forecasts to make the results more robust.

Along the aforementioned points, further research can explore several areas to enhance the understanding and forecasting of the VIX index. Including more explanatory variables such as macroeconomic indicators, S&P 500 stock data, and global financial indices might improve model accuracy and would lay the ground for interesting variable importance analyses. Furthermore, exploring more sophisticated machine learning techniques can capture more complex patterns in the VIX time series. For example, the approach by Kleen and Tetereva (2022) of employing a panel of HAR models instead of individual HAR models resulting in pooled estimators would have already been a great idea for improvement.

References

Ahoniemi, K. (2006, 07). Modeling and forecasting implied volatility - an econometric analysis of the vix index.

Ahoniemi, K. (2008). Modeling and forecasting the vix index. *Available at SSRN 1033812*.

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

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

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... Yuan, J. (2024). xgboost: Extreme gradient boosting [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=xgboost` (R package version 1.7.7.1)

Christoffersen, P., Feunou, B., Jacobs, K. & Meddahi, N. (2014). The economic value of realized volatility: Using high-frequency returns for option valuation. *Journal of Financial and Quantitative Analysis*, *49*(3), 663–697.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, *7*(2), 174–196.

Degiannakis, S., Filis, G. & Hassani, H. (2018). Forecasting global stock market implied volatility indices. *Journal of Empirical Finance*, *46*, 111–129.

Degiannakis, S. A. (2008). Forecasting vix. *Journal of Money, Investment and Banking*(4).

Duan, J.-C. (1995). The garch option pricing model. *Mathematical finance*, *5*(1), 13–32.

Engle, R. F. & Bollerslev, T. (1986). Modelling the persistence of conditional variances. *Econometric reviews*, *5*(1), 1–50.

Espejo, L. (2024). *Garch, egarch, nagarch, gjr models and implicit vix.* `https://www.mathworks.com/matlabcentral/fileexchange/44113-garch-egarch-nagarch-gjr-models-and-implicit-vix`. (MATLAB Central File Exchange. Retrieved June 24, 2024)

Fouque, J.-P., Papanicolaou, G. & Sircar, K. R. (2000). Mean-reverting stochastic volatility. *International Journal of theoretical and applied finance*, *3*(01), 101–142.

Friedberg, R., Tibshirani, J., Athey, S. & Wager, S. (2020). Local linear forests. *Journal of Computational and Graphical Statistics*, *30*(2), 503–517.

Hao, J. & Zhang, J. E. (2013). Garch option pricing models, the cboe vix, and variance risk premium. *Journal of Financial Econometrics*, *11*(3), 556–580.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, *6*(2), 327–343.

Heston, S. L. & Nandi, S. (2000). A closed-form garch option valuation model. *The review of financial studies*, *13*(3), 585–625.

Hyndman, R. J. & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J. & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, *27*, 1–22.

Kleen, O. & Tetereva, A. (2022). A forest full of risk forecasts for managing volatility. *Available at SSRN 4161957*.

Kotu, V. & Deshpande, B. (2018). *Data science: concepts and practice*. Morgan Kaufmann.

Liu, Q., Guo, S. & Qiao, G. (2015). Vix forecasting and variance risk premium: A new garch approach. *The North American Journal of Economics and Finance*, *34*, 314–322.

Meddahi, N. & Renault, E. (2004). Temporal aggregation of volatility models. *Journal of Econometrics*, *119*(2), 355–379.

Poon, S.-H. & Granger, C. W. J. (2003). Forecasting volatility in financial markets: A review. *Journal of economic literature*, *41*(2), 478–539.

Prasad, A., Bakhshi, P. & Guha, D. (2023). Forecasting the direction of daily changes in the india vix index using deep learning. *IIMB Management Review*, *35*(2), 149–163.

Saha, A., Malkiel, B. G. & Rinaudo, A. (2019). Has the vix index been manipulated? *Journal of Asset Management*, *20*(1), 1–14.

Stein, E. M. & Stein, J. C. (1991). Stock price distributions with stochastic volatility: an analytic approach. *The review of financial studies*, *4*(4), 727–752.

Tibshirani, J., Athey, S., Sverdrup, E. & Wager, S. (2024). grf: Generalized random forests [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=grf` (R package version 2.3.2)

Wang, H. (2019). Vix and volatility forecasting: A new insight. *Physica A: Statistical Mechanics and its Applications*, *533*, 121951.

Wang, S., Li, K., Liu, Y., Chen, Y. & Tang, X. (2024). Vix constant maturity futures trading strategy: A walk-forward machine learning study. *Plos one*, *19*(4).

Wang, Y., Pan, Z., Zheng, J., Qian, L. & Mingtao, L. (2019, 08). A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, *364*.

Wiggins, J. B. (1987). Option values under stochastic volatility: Theory and empirical estimates. *Journal of financial economics*, *19*(2), 351–372.

Wu, X., He, Q. & Xie, H. (2023). Forecasting vix with time-varying risk aversion. *International Review of Economics & Finance*, *88*, 458–475.

Zhang, W. & Zhang, J. E. (2020). Garch option pricing models and the variance risk premium. *Journal of Risk and Financial Management*, *13*(3), 51.

Zhang, Y. (2022). Stock price prediction method based on xgboost algorithm. In *Proceedings of the 2022 international conference on bigdata blockchain and economy management (icbbem 2022)* (p. 595-603). Atlantis Press.

A Appendix

A.1 Code

The code is attached in the zip file. To make the code fully understandable, I will explain a few things from the code in this Appendix. First, the code for the historical estimation of the GARCH model is done with a modified version of code by Espejo (2024). The AGARCH and TGARCH models were then based on the modified GARCH model but written by myself, along with files for transforming the data and running. The Matlab code consists of the following files:

- **DateConverter**: load the data and make sure the dates are transformed into the right format. The dataset has already been cleaned to get the right time period. Run this first.
- **DateConverterNew**: same as the above, but will load the data for the forecasting period. One should run this after DateConverter to also obtain the total period.
- **Modelsneeded**: a class with methods that all GARCH models use. It has a constructor, a function for plotting the implicit VIX, a function for estimating model parameters and a function to convert the T-Bill rates to the format used in Hao and Zhang (2013).
- **garch, agarch, and tgarch**: these classes specify the likelihoods for the different models and the structure of the forecasts. They inherit Modelsneeded.
- **RunningScript**: this is the file we run the code in, as to be seen in Section A.2.

The rest of the models were implemented in R, with the corresponding packages mentioned in this paper. The code in the R-markdown files can be run sequentially, and there are instructive comments placed where needed. The R-code consists of the following files:

- **DataCleaning**: obtain the right time period from the VIX, S&P 500, and T-Bill dataset. The final clean datasets will also be attached in the zip folder such that one does not have to run this file.
- **AR(1) model**: fits the AR(1) model as described in the text. Includes additional code for the confusion matrix.
- **ARIMA(3,1,3) model**: fits the ARIMA(3,1,3) model using the auto.arima function. Includes additional code for the confusion matrix.
- **HAR(1) model**: fits the HAR(1) model as described in the text. Includes additional code for the confusion matrix.
- **XGBoost (VIX)**: fits the XGBoost model using the lagged VIX time series. A seed is set at the beginning to make the results reproducible. Includes additional code for the confusion matrix.
- **XGBoost (RVAP)**: fits the XGBoost model using the moving averages of the VIX time series. A seed is set at the beginning to make the results reproducible. Includes additional code for the confusion matrix.
- **PlottingGarch**: used to make the plots in Figure 9. The data of the predictions that should be loaded will be provided in the zip folder.
- **Diebold-Mariano**: used to perform the Diebold-Mariano tests as seen in Table 8. The required data is provided in the zip file.

A.2 Start parameters of the GARCH models

This code was run to estimate the GARCH models. The initial parameters used in the optimization algorithm are $\alpha_0, \alpha_1, \beta_1, \theta$, and λ respectively. Each model is run by changing the *garch_model* specification in line 2 to either garch, agarch, or tgarch, and removing the %-sign from the corresponding line of code.

```
1  %Estimate the garch model
2  garch_model = garch(SP500_total, TrBill_total, VIX_total, 1990,
       1, 2, 2009, 8, 10, 252);
3
4  %GARCH
5  %modelestimates(garch_model,"Returns",[10^-7,0.1,0.3,0.001]);
6  %modelestimates(garch_model,"VIX",[10^-7,0.1,0.3,0.7]);
7  %modelestimates(garch_model,"ReturnsVIX",[10^-7,0.01,0.3,0.001])
       ;
8
9  %AGARCH
10 %modelestimates(garch_model,"Returns
       ",[10^-7,0.05,0.88,1.01,0.01]);
11 %modelestimates(garch_model,"ReturnsVIX
       ",[10^-7,0.03,0.9,0.01,0.77]);
12
13 %TGARCH
14 %modelestimates(garch_model,"Returns",[10^-7,0.1,0.9,0.1,0.02]);
15 %modelestimates(garch_model,"VIX",[10^-7,0.2,0.3,0.02,0.03]);
16 %modelestimates(garch_model,"ReturnsVIX
       ",[10^-7,0.004,0.9,0.04,0.4]);
```

A.3 Diebold-Mariano test

	GARCH	TGARCH	AGARCH	XGBOOST(RVAP)	XGBOOST(VIX)	LLF	ARIMA(3,1,3)	AR(1)
P-value	4.608e-12	8.714e-10	2.883e-15	0.1171	0.2590	0.4047	0.3363	0.4036

Table 8: P-values of the Diebold-Mariano tests.

A.4 Parameter tuning of the XGBoost models

Parameter	Description	Values checked	Returns	RVAP
nrounds	Number of trees to build	50, 100, 150	100	50
max_depth	Maximum depth of one tree	4, 6, 8	4	4
colsample_bytree	Fraction of features sampled for each tree	0.5, 0.75, 1	0.75	1
eta	The learning rate, which controls the step size at each iteration	0.01, 0.05, 0.1	0.05	0.1
gamma	Minimum loss reduction required at every split	0, 0.1, 0.2	0.2	0.2
min_child_weigth	Minimum sum of instance weights required in each child node during the tree building process	1, 3, 5	5	5
subsample	Fraction of training data sampled for each tree	0.6, 0.8, 1	0.6	0.6

Table 9: Hyperparameter tuning for the XGBoost model.

A.5 Parameter tuning of Local Linear Forest model

The following parameters were tuned for the Local Linear Forest model. The description is taken from the documentation of the 'll_regression_forest' function in the R-package 'grf' (Tibshirani et al., 2024). All other parameters are set to the default values indicated in the aforementioned package.

Parameter	Description	Values checked	RVAP
num.trees	Number of decision trees in the forest	2000	2000
mtry	Number of variables tried for each split of a tree	1, 2, 3	1
alpha	Controls the maximum imbalance of a split	0.01, 0.05, 0.1	0.1
min.node.size	A target for the minimum number of observations in each tree leaf	5, 10, 15	15
imbalance.penalty	Controls how harshly imbalanced splits are penalized	0, 0.25, 0.5	0
honesty.fraction	The fraction of data that will be used for determining splits	0.5	0.5
honesty.prune.leaves	Prunes the estimation sample tree such that no leaves are empty	TRUE	TRUE

Table 10: Hyperparameter tuning for the LLF model.

A.6 Hyndman-Khandakar Algorithm

Algorithm 1 Hyndman-Khandakar Algorithm for Automatic ARIMA Modelling

Step 1: Determine the number of differences
Set $0 \leq d \leq 2$ using repeated KPSS tests.
Step 2: Choose the values of p and q by minimizing AICc
(a) Initial Model Fitting
Fit the following four initial models:
    ARIMA$(0, d, 0)$
    ARIMA$(2, d, 2)$
    ARIMA$(1, d, 0)$
    ARIMA$(0, d, 1)$
if $d \leq 1$ **then**
    Fit ARIMA$(0, d, 0)$ without a constant.
end if
Select the model with the smallest AIC as the current model.
(b) Stepwise Model Selection
repeat
    Consider variations of the current model by:
        Varying p and/or q by ± 1 , ensuring $0 \leq p \leq 5$ and $0 \leq q \leq 5$ manually set to limit the amount of lags.
        Including/excluding a constant.
    Evaluate these variations and select the model with the smallest AIC.
    Update the current model to the best model found in this step.
until No model with a lower AIC is found