From Traditional to Modern Techniques: Forecasting Financial Market Volatility with Econometric and Machine Learning Models

Anja Petric (598370), Dominik Baus (614414),
Iryna Matsiuk (598762) and Boris Rine (613746)

ERASMUS UNIVERSITEIT ROTTERDAM

Abstract

This study evaluates the predictive accuracy of various volatility forecasting models, including traditional econometric models (GARCH(1,1) and Beta-t-EGARCH), benchmark models (VIX and HAR-RV), and advanced machine learning techniques (LSTM, XGBoost and Random Forest). We assess each model's performance across multiple forecast horizons using daily data from the S&P 500 index and robust loss functions. We find that asymmetric models perform better than symmetric models in-sample. Our findings further reveal that hybrid models, which combine machine learning with traditional econometrics, significantly outperform standalone econometric models, particularly over longer time horizons. This integration allows for capturing complex, non-linear relationships within the data, enhancing predictive performance. The results highlight the potential of combining machine learning methods with traditional econometric approaches to improve financial volatility forecasts, offering valuable insights for both theoretical research and applied financial practices.

Date: May 24, 2024

1 Introduction

Volatility modelling has always been a critical aspect of financial markets, helping investors manage risk and optimize returns. Over time, various models have been developed to predict and manage volatility, but the key question remains: Are these complex models necessary, or are simple models sufficient?

This report aims to compare the performance of traditional econometric models, benchmark models, and state-of-the-art machine learning models in forecasting the volatility of the S&P 500 index. We examine classic models such as the widely-used GARCH(1,1) introduced by Bollerslev (1986) and the more advanced Beta-t-EGARCH model proposed by Harvey and Chakravarty (2008). Additionally, we evaluate benchmark models, including the Volatility Index (VIX) developed by the Chicago Board Options Exchange (CBOE) and the heterogeneous autoregressive realized volatility (HAR-RV) model by Corsi (2009).

To explore advanced methodologies, we incorporate machine learning approaches, specifically Long Short-Term Memory (LSTM) networks, Random Forest, and XGBoost. The motivation for using these models lies in their flexibility and ability to handle complex, non-linear relationships within the data, which traditional econometric models might not effectively capture. While GARCH models are favoured in econometrics for their tractability and interpretability—enabling analysts to understand the underlying mechanics and derive insights from model parameters—they rely on certain assumptions and constraints that may limit their adaptability. In contrast, machine learning models operate as 'black boxes,' meaning they may lack interpretability but can potentially outperform GARCH models because they are not constrained by the need for interpretability, allowing for superior predictive performance.

This report aims to determine which model provides the most accurate volatility predictions, and whether any of these models can outperform the well-established GARCH model. By evaluating these models using daily returns, realized variance measures across multiple horizons, and robust loss functions, we seek to provide insights into the most efficient methods for volatility forecasting.

The report's structure is as follows: Section 2 describes the data. Section 3 presents explains the methodology used. Section 4 presents the results and lastly, Section 5 summarizes the findings and concludes the report.

2 Data

The data used in our research is obtained from Yahoo Finance, covering the period from January 2000 to February 2024. The dataset includes daily information on the S&P500 index and related metrics, namely: *Opening Price of S&P500*: The price of the S&P500 index at market open each trading day, *Closing Price of S&P500*: The price of the S&P500 index at market close each trading day, *Close-to-Close Log Returns*: The natural logarithm of the ratio of consecutive closing prices, *Realized Variance (rv5)*: Computed from intra-day returns measured at 5-minute intervals, representing the sum of squared returns within each trading day. *Closing Level of the VIX*: The closing value of the VIX, which represents the market's expectation of volatility over the next 30 days, annualized.

3 Methodology

In this section, we outline the methodologies employed to forecast volatility and evaluate the performance of various models. The aim is to determine the effectiveness of different volatility forecasting models, including traditional econometric models and modern machine learning techniques in predicting volatility.

3.1 GARCH models

3.1.1 GARCH(1,1) model

We first employ the generalized autoregressive conditional heteroskedasticity (GARCH) model of order (1,1), as introduced by Bollerslev (1986). This model denotes the conditional variance of returns, assuming that the return series r_t is generated by:

$$r_t = \mu + \sigma_{t|t-1}\epsilon_t, \quad 1 \leq t \leq T, \tag{3.1.1}$$

where T denotes the sample size, and ϵ_t are independently and identically distributed (i.i.d.) 'shocks' with $E[\epsilon_t] = 0$ and $V[\epsilon_t] = 1$. We define volatility as $\sigma^2_{t|t-1}$ and assume the residuals to follow a Student's t-distribution. The implied residuals can be obtained by solving equation (3.1.1) for ϵ. The GARCH(1,1) model denotes the conditional variance at time $t+1$ as:

$$\sigma^2_{t+1|t} = \omega + \alpha(r_t - \mu)^2 + \beta\sigma^2_{t|t-1}, \tag{3.1.2}$$

where α, β, and ω are parameters. We impose $\omega > 0$, $\alpha \geq 0$ and $\beta \geq 0$ to guarantee that $\sigma^2_{t+1|t} \geq 0$ for all t. This standard GARCH(1,1) model assumes that positive and negative shocks ϵ_t affect the volatility at time $t+1$ similarly. To account for asymmetric effects, we consider the following asymmetric GARCH model:

$$\sigma^2_{t+1|t} = \omega + (\alpha_{\text{pos}}1_{\epsilon_t \geq 0} + \alpha_{\text{neg}}1_{\epsilon_t < 0})\epsilon_t^2 + \beta\sigma^2_{t|t-1}, \tag{3.1.3}$$

where the coefficients α_{pos} and α_{neg} describe the sensitivity of volatility to positive and negative shocks, respectively. We impose $\omega > 0$, $\alpha_{\text{pos}} \geq 0$, $\alpha_{\text{pos}} \geq 0$ and $\beta \geq 0$. We plot the news impact curve (NIC)

$$NIC_t = \frac{\sigma^2_{t+1|t} - \sigma^2_{t|t-1} - E_{t-1}[\sigma^2_{t+1|t} - \sigma^2_{t|t-1}]}{\sigma^2_{t|t-1}} = \frac{\sigma^2_{t+1|t} - E_{t-1}[\sigma^2_{t+1|t}]}{\sigma^2_{t|t-1}}. \tag{3.1.4}$$

to assess how the conditional variance $\sigma^2_{t+1|t}$ changes in response to shocks ϵ_t. To obtain the predicted volatility (PVol) for the sum of days up to a horizon h, we compute:

$$PVol_t = \sum_{d=1}^{h} E_t[r^2_{t+d}] = h\mu^2 + \sum_{d=1}^{h} E_t[\sigma^2_{t+d|t+d-1}], \tag{3.1.5}$$

where r_{t+d} represents the returns and μ is the expected return. We construct predictions for $d = 1$, $d = 5$, and $d = 21$, corresponding to one day, one week, and one month ahead predictions to evaluate the volatility forecasting performance of the model.

3.1.2 Beta-t-EGARCH model

The Beta-t-EGARCH model, proposed by Harvey and Chakravarty (2008) and further developed by Harvey (2013), addresses some of the limitations of traditional GARCH models by incorporating both the size and sign of shocks to model volatility. This model ensures that the conditional variance remains positive by by setting $\sigma_{t|t-1} = \exp(\lambda_{t|t-1})$. The model is formulated as follows:

$$
\begin{aligned}
r_t &= \mu + \exp(\lambda_{t|t-1})\epsilon_t && (3.1.6)\\
\lambda_{t+1|t} &= \lambda(1-\phi) + \phi\lambda_{t|t-1} + \kappa u_t + \tilde{\kappa} v_t && (3.1.7)\\
u_t &= \sqrt{\frac{\nu+3}{2\nu}}\left(\frac{\nu+1}{\nu-2+\epsilon_t^2} - 1\right) && (3.1.8)\\
v_t &= \sqrt{\frac{(\nu-2)(\nu+3)}{\nu(\nu+1)}}\frac{\nu+1}{\nu-2+\epsilon_t^2}\epsilon_t && (3.1.9)
\end{aligned}
$$

where μ is the unconditional mean of the returns, $\lambda_{t|t-1}$ is the dynamic scale parameter and where u_t and v_t are robust variance and location measures, respectively. The parameter ϕ represents the persistence of volatility, determining how much past volatility influences current volatility. The parameters κ and $\tilde{\kappa}$ capture the impact of the magnitude and sign of shocks on volatility, respectively. Lastly, ν, represents the degrees of freedom of the Student's t-distribution, capturing the tail behavior of the distribution.

As with the GARCH models, we estimate both a symmetric and an asymmetric version of the Beta-t-EGARCH model. The symmetric version sets $\tilde{\kappa} = 0$, implying that the volatility only responds to the magnitude of the shocks and not their sign. The news impact curve (NIC) for the Beta-t-EGARCH model is defined as:

$$NIC_t = 2\kappa u_t + 2\tilde{\kappa} v_t, \tag{3.1.10}$$

As for the GARCH model, we generate h-day ahead volatility forecasts for $h = 1$, $h = 5$ and $h = 21$. The predicted volatility over an h-day horizon is given by:

$$PVol_t = h\mu^2 + \sum_{d=1}^{h} \exp\left(2\lambda + 2\phi^{d-1}(\lambda_{t+1|t} - \lambda) + 2(\kappa^2 + \tilde{\kappa}^2)\frac{1-\phi^{2(d-1)}}{1-\phi^2}\right) \tag{3.1.11}$$

where λ is the unconditional mean of $\lambda_{t|t-1}$.

3.2 Machine Learning models

To try and improve volatility forecasting, we train machine learning (ML) models using realized variance, VIX close data and point predictions from GARCH models. The VIX, a forward-looking indicator, reflects the market's view on future risk and uncertainty, while realized variance, obtained from high-frequency intraday returns, captures historical volatility as a backward-looking measure. GARCH models use past observations to predict future volatility by modeling the conditional variance of returns and effectively capture volatility clustering.

The structure of GARCH point predictions guides ML algorithms. Classical GARCH models impose structure by assuming how volatility develops, which makes them interpretable but limits forecasting flexibility. In contrast, machine learning allows for more flexible dynamics without requiring interpretability. Due to the lack of structure, ML models may not be able to fully learn the right dynamics, with the downside of possible overfitting. By incorporating GARCH point predictions as features in ML models, we aim to balance the strengths of both approaches, sacrificing some interpretability for improved forecasting ability.

We use GARCH-type models in combination with three ML models (LSTM, XGBoost and random forest) to enhance their predictive performance, as described below.

3.2.1 Long Short-Term Memory (LSTM)

As introduced by Hochreiter and Schmidhuber (1997), LSTM is a recurrent neural network designed to capture long-term dependencies in sequential data. LSTM uses memory cells (c_t) and gates (input gate (i_t), forget gate (g_t) and an output gate(o_t)) to store information for long periods of time, and forget unnecessary information, as described in equations below:

$$g_t = \sigma(U_g x_t + W_g h_{t-1} + b_f) \tag{3.2.1}$$

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{3.2.2}$$

$$\tilde{c}_t = \tanh(U_c x_t + W_c h_{t-1} + b_c) \tag{3.2.3}$$

$$c_t = g_t * c_{t-1} + i_t * \tilde{c}_t \tag{3.2.4}$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \tag{3.2.5}$$

$$h_t = o_t * \tanh(c_t) \tag{3.4.6}$$

Where σ is the sigmoid function, tanh is the hyperbolic tangent function, x_t is the input at time t, $\tilde{c}_t$ is an input modulate gate, h_t is the hidden state, U and W are weight matrices and b is a bias term.

We predict volatility at $t + 1$, $t + 5$, $t + 21$ using hybrid models that combine LSTM with an asymmetric GARCH(1,1) and LSTM with asymmetric beta-t-EGARCH, similar to Kim and Won (2018).

3.2.2 Extreme Gradient Boosting (XGBoost)

Similarly, we use the hybrid of XGBoost with an asymmetric GARCH(1,1) and XGBoost with an asymmetric beta-t-EGARCH to create volatility forecasts. Our approach is to combine the original features of the dataset with the extracted GARCH features and use it as an input to train the XGBoost model, similar to Dai, Huang, Zeng, and Zhou (2022).

As Chen and Guestrin (2016) described, XGBoost is a gradient tree boosting algorithm: it builds a series of decision trees sequentially, each attempting to correct the errors of its predecessors, gradually improving predictive accuracy. The basic idea behind XGBoost involves starting with an initial prediction, typically the mean of the target variable, and sequentially adding new trees to the model. Each new tree is trained to predict the residual errors of the existing model, thereby refining the overall prediction. To control model complexity and prevent overfitting, XGBoost applies regularization techniques, incorporating both L1 (lasso) and L2 (ridge) penalties, which help ensure the model generalizes well to new data.

XGBoost can be expressed with the following equation:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{A.1}$$

where $\hat{y}_i$ is the predicted value, K is the total number of trees, f_k represents the k-th tree model, and x_i is the i-th input sample. The objective function is defined as:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{A.2}$$

The regularization term in XGBoost is given by:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{A.3}$$

where T denotes the number of leaf nodes per tree, w_j represents the weight of each leaf, and γ and λ are parameters used to control tree growth and prevent overfitting. λ is the L2 regularization factor and γ is the split threshold. Applying Taylor's expansion to the objective function, we get:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{A.4}$$

where $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ and $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)2}}$. Here, $l(y_i, \hat{y}_i^{(t-1)})$, g_i, and h_i are constants.

$$\omega^* = -\frac{\sum_{i \in Ij} g_i}{\sum_{i \in Ij} h_i + \lambda} \tag{A.5}$$

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \tag{A.6}$$

The contribution and information gain from a split in the objective function are computed with equations (A.5) and (A.6), respectively. Where G_L and G_R are the first-order gradient statistics for the left and right leaves, and H_L and H_R are the second-order gradient statistics sums for the left and right leaves at the split point, respectively.

For further information about XGBoost, please refer to Appendix A. We implement the model with the help of the xgboost package in Python.

3.2.3 Random chent

The random forest technique, introduced by Breiman (2001), is a robust ensemble learning strategy that constructs multiple decision trees and aggregates their predictions for regression tasks. This method incorporates randomness by selecting random subsets of data points (bootstrap sampling) and random subsets of features at each split. This approach reduces correlation among trees, minimizing overfitting and enhancing prediction stability. Each tree grows to maximum depth without pruning, overfitting the bootstrap sample. The model then averages the predictions from all trees to form the final prediction.

We combined Random Forest with asymmetric GARCH(1,1) and asymmetric Beta-t-EGARCH models to forecast volatility. By incorporating GARCH parameters such as conditional variance and lagged returns as features, we blend the interpretability and insights of GARCH models with the non-linear predictive power of Random Forests.

3.3 In-sample analysis

To estimate the parameters of both the GARCH and Beta-t-EGARCH models, we use the maximum likelihood (ML) estimation. The estimation process involves maximizing the log-likelihood function based on the observed return series r_t. For both models, the log-likelihood function $L(\theta)$ for the parameter vector θ is given by:

$$L(\theta) = \sum_{t=1}^{T} \left[-\log(\sigma_{t|t-1}) + \log p\left(\epsilon_t\right) \right] \tag{3.3.1}$$

where $p(\epsilon_t)$ is the probability density function of the Student's t-distribution. The initial values for the parameters are chosen based on preliminary analysis and common practices in volatility modeling. Appropriate constraints are set to ensure parameter values are within plausible ranges (e.g., $\omega > 0$, $\alpha \geq 0$, $\beta \geq 0$, $|\phi| < 1$).

After estimating both the symmetric and asymmetric versions of the GARCH and Beta-t-EGARCH models, we use the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) to select the best-fitting version of each model. The reason why we use information criteria instead of the log-likelihood is because models with more parameters have higher log-likelihoods due to the better fit. The model with the lower AIC and BIC values is chosen for subsequent analysis.

We will check the efficiency of our forecast by considering the Mincer-Zarnowitz regression of the following form:

$$y_{t+1} = \alpha + \beta \hat{y}_{t+1|t} + \eta_{t+1} \tag{3.3.2}$$

for which it should hold that $\alpha = 0$ and $\beta = 1$. As a dependent variable we will use a target variance based on daily return data. In particular, the target variance is defined as sum of squared close-to-close log returns over all future days up to the prediction horizon.

3.4 Out-of-sample analysis

To evaluate the performance of our models, we compare them against two benchmark models: the VIX and the HAR-RV model of Corsi (2009). The first benchmark model is the VIX, which predicts the (annualised) standard deviation of the S&P500 index returns over the next month. The predicted variance at time t over a d-day horizon is given by:

$$PV_{t,d} = \frac{d}{250} \times \text{VIX}_t^2 \tag{3.4.1}$$

where the factor $\frac{d}{250}$ adjusts for the d-day horizon based on 250 trading days per year.

The second benchmark model is the HAR-RV model, which directly models realised variance using an autoregressive structure with different frequencies:

$$\text{TV}_{t,d} = c_d + \beta_d^{(d)} \text{RV}_t^{(d)} + \beta_d^{(w)} \text{RV}_t^{(w)} + \beta_d^{(m)} \text{RV}_t^{(m)} + e_t \tag{3.4.2}$$

where $\text{RV}_t^{(d)}$, $\text{RV}_t^{(w)}$, and $\text{RV}_t^{(m)}$ are the daily, weekly, and monthly realised variances, respectively. The model parameters are estimated using ordinary least squares (OLS). The prediction is obtained by filling in the estimated parameters in the right-hand side of the equation.

To evaluate the models, we use the mean squared error (MSE), the QLIKE loss function (QLIKE), and the mean absolute error(MAE):

$$\text{MSE} = \frac{1}{N}\sum_{t=1}^{N}(\text{TV}_t - \text{PV}_t)^2 \tag{3.4.3}$$

$$\text{QLIKE} = \frac{1}{N}\sum_{t=1}^{N}\left(\log(\text{PV}_t) + \frac{\text{TV}_t}{\text{PV}_t}\right) \tag{3.4.4}$$

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|\text{TV}_t - \text{PV}_t| \tag{3.4.5}$$

where PV is the volatility prediction by the model and TV is the target variance. The target variance is either based on daily returns, as described above, or a target variance based on the sum of intraday returns up to the prediction horizon multiplied by the factor 1.4. The MAE is included because it can be interesting to include a loss function which puts less weight on outlier predictions.

Building on the original framework proposed by Diebold and Mariano (1995), the Diebold-Mariano (DM) test remains a widely used statistical tool for comparing the predictive accuracy of competing forecast models, as reflected upon in Diebold's retrospective analysis two decades later (Diebold, 2015). The DM test evaluates the null hypothesis that two forecasts have the same predictive accuracy by comparing their forecast errors. The test statistic is given by:

$$DM = \frac{\bar{d}}{\sqrt{V(\hat{d}_{t+1})/P}} \tag{3.4.6}$$

where d_t is the difference in the loss values of the two models at time t, and $\bar{d}$ is the mean of these differences.

We perform econometric out-of-sample analysis using a rolling window approach. We use only the first 70% of the data to estimate the model parameters and generate forecasts for the remaining data. This process is repeated, moving the window each time new data becomes available. We construct volatility predictions for $d = 1$, $d = 5$, and $d = 21$ days ahead, for each model: HAR-RV, VIX, GARCH, Beta-t-EGARCH. Similarly, for hybrid models, we do a 70:30 split, where the first group trains the model and the second tests it. While we acknowledge the benefits of maintaining consistency across all models, we do not employ a rolling window approach for hybrid models due to computational limitations as retraining requires significant computational resources.

4 Results

We implemented the econometric models in MATLAB. The hybrid machine learning models were implemented using Python. Output for the econometric models was equal for both programming languages.

4.1 In-sample analysis

As can be seen in Table 1, for both GARCH and Beta-t-EGARCH, the asymmetric models exhibit lower AIC and BIC; moreover, the $H_0 : \alpha_{pos} = \alpha_{neg}$ is rejected by the

Table 1: Parameter Estimates for GARCH and Beta-t-EGARCH Models

Parameter	**Symm GARCH**	**Asymm GARCH**	**Symm Beta-t-EGARCH**	**Asymm Beta-t-EGARCH**
μ	0.076	0.046	0.076	0.036
ω	0.014	0.017	–	–
α	0.121	–	–	–
α_{pos}	–	0	–	–
α_{neg}	–	0.208	–	–
β	0.876	0.884	–	–
ν	6.470	6.892	7.178	8.500
λ	–	–	0.000	-0.029
ϕ	–	–	0.983	0.974
κ	–	–	0.080	0.061
$\tilde{\kappa}$	–	–	–	-0.091
Log-Likelihood	-8211.3	-8111.4	-8228.8	-8088.5
AIC	16433	16235	16468	16189
BIC	16466	16275	16501	16229
LR stat	199.8		280.6	
LR P-value	0.000		0.000	

Note: The LR test statistics provided are results of a Likelihood Ratio test comparing $H_0 : \alpha_{pos} = \alpha_{neg}$ versus $H_1 : \alpha_{pos} \neq \alpha_{neg}$.

LR test. Therefore the asymmetric models are a better fit. From Table 1 we see that according to the asymmetric GARCH model, negative shocks have a stronger effect on volatility compared to positive shocks, which can likely be attributed to the leverage effect. It can also be observed that $\beta + \frac{\alpha_{\text{pos}}}{2} + \frac{\alpha_{\text{neg}}}{2}$ is close to 1, which suggests that shocks to the conditional variance are persistent over time.

From the estimated coefficients of the asymmetric Beta-t-EGARCH model (Table 1), we again observe that negative shocks have a stronger effect on volatility than positive shocks due to the negative $\tilde{\kappa}$, indicating the leverage effect. Furthermore, the high value of $\phi = 0.974$ shows that volatility shocks are highly persistent over time. The degrees of freedom parameter ($\nu = 8.5$) suggests heavy tails, implying increased tail risk in the returns distribution. Figure 1 plots the implied residuals against the news impact curves for the asymmetric GARCH and the asymmetric Beta-t-EGARCH models. The first thing to note is that both news impact curves exhibit asymmetry. Roughly speaking negative implied residuals seem to have a higher impact on volatility than positive implied residuals. This is especially extreme in the case of the asymmetric GARCH model, where the impact of positive shocks on volatility is close to zero.

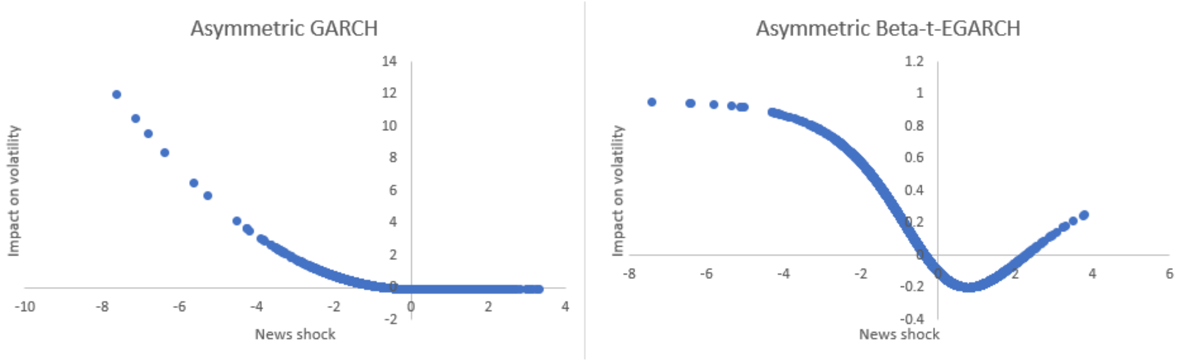

Figure 1: News impact curves for asymmetric GARCH and EGARCH models

One possible explanation for this phenomenon could be that the market responds more extreme in the case of bad news which increases volatility more than good news. Another observation is that the more extreme in either direction the shock is, the higher the magnitude of the impact on volatility. Table 2 shows the output from the Mincer-Zarnowitz regressions for both asymmetric models.

Table 2: Output from Mincer-Zarnowitz regressions

Parameter	**Asymmetric GARCH**			**Asymmetric Beta-t-EGARCH**		
	1 day	5 days	21 days	1 day	5 days	21 days
α	0.129	1.075	10.597	-0.236	-1.237	-4.697
	(0.064)	(0.186)	(0.726)	(0.065)	(0.189)	(0.831)
β	0.843	0.793	0.621	1.163	1.178	1.174
	(0.017)	(0.010)	(0.010)	(0.022)	(0.014)	(0.018)
RMSE	4.4	12.9	49.4	4.4	12.3	47.8
R^2	0.277	0.488	0.370	0.307	0.537	0.410
Wald stat	4007	10703	11420	2646	6638	5247
Wald P-value	0.000	0.000	0.000	0.000	0.000	0.000

Note: The Wald test statistics provided are results of a Wald test comparing $H_0 : \beta_0 = 0$ and $\beta_1 = 1$ versus $H_1 : \beta_0 \neq 0$ and $\beta_1 \neq 1$

From Table 2 we observe that predictions extending further into the future show higher root mean squared errors for both models, indicative of decreasing predictive accuracy over time. In the asymmetric GARCH model, the increasing positivity of α correlates with an underprediction of volatility, highlighting a potential limitation in the model's responsiveness to market shocks. Conversely, the asymmetric Beta-t-EGARCH model overpredicts volatility under similar conditions, suggesting different sensitivities in how each model processes market information. The rejection of $H_0 : \beta_0 = 0$ and $\beta_1 = 1$ in the Wald tests for both models across all forecast horizons further confirms that our forecasts systematically deviate from the observed variances, pointing to non-optimal model specifications or the need for additional model enhancements.

4.2 Out-of-sample analysis

Table 3: Models predictions evaluation table using MSE

Model	1 day	5 days	21 days
Asymm GARCH	27.8	366.8	6332.8
Asymm Beta-t-EGARCH	26.1	313.4	4511.5
VIX	29.9	355.6	4645.4
HAR-RV	25.6	321.3	4517.8
XGBoost + Asymm Garch	28.2	231.8	3814.1
XGBoost + Asymm Beta-t-EGarch	26.5	248.2	3881.5
Random Forest + Asymm Garch	25.1	258.3	4327.2
Random Forest + Asymm Beta-t-EGarch	24.8	260.0	4360.0
LSTM + Asymm Garch	19.4	261.1	4338.1
LSTM + Asymm Beta-t-EGarch	19.8	244.2	4499.4

Note: Here target variance computed from daily returns is used as a target for the MSE.

In Table 3, we present mean squared errors (MSEs) computed from forecast errors derived from target variances calculated from daily returns for 10 models across 3 forecast horizons. Among these models, we observe that LSTM combined with asymmetric GARCH yields the lowest MSE for a one-day horizon, while XGBoost combined with asymmetric GARCH performs best for 5 and 21 days. Comparison of Hybrid models between each other based on MAE and QLIKE leads to similar results: LSTM and XGBoost combined with asymmetric GARCH or Asymmetric Beta-t-EGARCH tend to have the lowest evaluation metrics values, as detailed in Table 8 and Table 9 in Appendix C. Our analysis also compares 4 econometric models using forecast errors based on intra-day target variance. We can observe that asymmetric Beta-t-EGARCH and HAR-RV consistently demonstrate the lowest values for the 3 above-mentioned evaluation metrics and two different target variances. Further details exhibiting that the four econometric models perform similarly in comparison for both target variances can be found in Table 5, Table 6, and Table 7 in Appendix B.

We have also conducted a sensitivity analysis on the MSE of the econometric models over time. Our goal was to see how MSE behaves over time and whether certain moments in time have a larger impact on one model than another. When wee plotted MSE over time, we saw a spike at the beginning of the COVID-19 pandemic. Some plots for various time horizons and target variances can be found in Appendix F. We noticed that these spikes are of different magnitude for the different econometric models. Interestingly, it seems like the main driver of the differences in MSE at the end is mainly caused by that spike at the beginning of 2020.

Table 4: Diebold-Mariano tests results for MSE comparison with econometric models 1 day

	VIX	HAR-RV	Asymm GARCH	Asymm Beta-t-EGARCH
VIX	-	-	-	-
HAR-RV	-1.25	-	-	-
Asymm GARCH	-0.43	0.85	-	-
Asymm Beta-t-EGARCH	-1.34	0.36	-0.67	-
XGBoost + Asymm Garch	-1.07	0.89	0.14	0.95
XGBoost + Asymm Beta-t-EGarch	-0.97	0.38	-0.33	0.28
Random Forest + Asymm Garch	-0.51	-0.06	-0.54	-0.14
Random Forest + Asymm Beta-t-EGarch	-0.66	-0.14	-0.83	-0.24
LSTM + Asymm Garch	-1.23	-1.07	-1.88*	-1.12
LSTM + Asymm Beta-t-EGarch	-1.36	-1.22	-2.12**	-1.28

Note: The results displayed in the table are DM test statistics

To assess the significance of differences in MSEs among models, we conduct the Diebold-Mariano test. The results for the 1-day forecast horizon are presented in Table 4. It's apparent that for the most part, there are no significant differences in performance among the models. However, we see that the LSTM model significantly outperforms the asymmetric GARCH(1,1) model at the 5% significance level. For longer forecast horizons the differences in prediction accuracy between more models become significant (see Appendix D for details). In such cases, hybrid models generally yield better forecasts compared to econometric ones, and HAR-RV and asymmetric Beta-t-EGARCH typically outperform VIX and asymmetric GARCH. Among Machine Learning models, we see that both variations of LSTM give better forecasts than both variations of Random forest (for details refer to Appendix E).

5 Conclusion

This report has explored the performance of various models in forecasting volatility. Our evaluation compared the econometric GARCH(1,1) and Beta-t-EGARCH models, used the VIX and the HAR-RV model as benchmarks and lastly compared these with advanced machine learning models including LSTM networks, Random Forest, and XGBoost.

From the in-sample analysis, we found that the asymmetric versions of GARCH and Beta-t-EGARCH models provided a better fit than their symmetric counterparts. These models more effectively captured the asymmetric impact of positive and negative shocks on volatility, reflecting the leverage effect observed in financial markets. Out-of-sample testing showed that the Beta-t-EGARCH and the HAR-RV perform best within the standard econometrics models. When introducing machine learning models, the LSTM and XGBoost models, when combined with the asymmetric GARCH, perform best overall and thus outperform the standard econometric models, as well as the benchmark models.

So, while traditional models like GARCH(1,1) are favoured for their clarity and ability to provide insightful analysis through interpretable parameters, their integration with flexible, albeit less interpretable, machine learning models such as LSTM and XGBoost significantly improves predictive accuracy. This combination effectively captures both linear relationships and complex dynamics, enhancing forecasting capabilities across extended horizons. Consequently, this advanced approach surpasses traditional models in performance, confirming that integrating these methodologies yields the most accurate volatility predictions. Thus, we see that adding more complex models can improve results significantly and that it is, in fact, possible to improve on the widely accepted GARCH(1,1) model.

Future research could focus on assessing the scalability of these integrated models under different market conditions, their responsiveness to sudden economic changes, and the potential benefits of incorporating real-time data for more timely predictions. Exploring the effects of alternative data sources, such as market sentiment analytics, could also enrich the models' forecasting power. Implementing the rolling window approach to train the machine learning models could enhance the results, too, as it would consider more recent trends and patterns in the data. Additionally, further analysing the MSE over time could provide more insights into the behaviour during certain periods and enhance understanding of the models.

In conclusion, this report demonstrates that integrating machine learning techniques with traditional econometric models improves the accuracy of volatility forecasts.

References

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

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

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).

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

Dai, H., Huang, G., Zeng, H., & Zhou, F. (2022). Pm2. 5 volatility prediction by xgboost-mlp based on garch models. *Journal of cleaner production*, *356*, 131898.

Diebold, F. X. (2015). Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of diebold–mariano tests. *Journal of Business & Economic Statistics*, *33*(1), 1-.

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

Harvey, A. C. (2013). *Dynamic models for volatility and heavy tails: With applications to financial and economic time series* (Vol. 52). Cambridge University Press.

Harvey, A. C., & Chakravarty, T. (2008). *Beta-t-(e)garch.* (Unpublished manuscript)

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Kim, H. Y., & Won, C. H. (2018, August). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications*, *103*, 25–37.

Appendix

A Additional explanation for XGBoost

XGBoost can be expressed with the following equation:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{A.1}$$

where $\hat{y}_i$ is the predicted value, K is the total number of trees, f_k represents the k-th tree model, and x_i is the i-th input sample. The objective function is defined as:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{A.2}$$

The regularization term in XGBoost is given by:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{A.3}$$

where T denotes the number of leaf nodes per tree, w_j represents the weight of each leaf, and γ and λ are parameters used to control tree growth and prevent overfitting. λ is the L2 regularization factor and γ is the split threshold. Applying Taylor's expansion to the objective function, we get:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{A.4}$$

where $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ and $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)2}}$. Here, $l(y_i, \hat{y}_i^{(t-1)})$, g_i, and h_i are constants.

$$\omega^* = -\frac{\sum_{i \in Ij} g_i}{\sum_{i \in Ij} h_i + \lambda} \tag{A.5}$$

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \tag{A.6}$$

The contribution and information gain from a split in the objective function are computed with equations (A.5) and (A.6), respectively. Where G_L and G_R are the first-order gradient statistics for the left and right leaves, and H_L and H_R are the second-order gradient statistics sums for the left and right leaves at the split point, respectively.

B Comparison of the two target variances

Table 5: Econometric models predictions evaluation table using MSE

Model	**Target variance daily**			**Target variance intra-day**		
	1 day	5 days	21 days	1 day	5 days	21 days
Asymm GARCH	27.8	366.8	6332.8	7.3	187.7	4680.4
Asymm Beta-t-EGARCH	26.1	313.4	4511.5	4.0	85.0	2007.0
VIX	29.9	355.6	4645.4	5.1	100.6	2324.3
HAR-RV	25.6	321.3	4517.8	4.1	82.9	2122.5

Table 6: Predictions evaluation table using MAE

Model	**Target variance daily**			**Target variance intra-day**		
	1 day	5 days	21 days	1 day	5 days	21 days
Asymm GARCH	1.49	5.25	23.67	0.84	4.14	20.79
Asymm Beta-t-EGARCH	1.40	4.63	20.68	0.70	3.54	17.70
VIX	1.64	5.85	24.69	1.00	4.85	22.96
HAR-RV	1.27	4.25	19.48	0.58	2.94	16.23

Table 7: Predictions evaluation table using QLIKE

Model	**Target variance daily**			**Target variance intra-day**		
	1 day	5 days	21 days	1 day	5 days	21 days
Asymm GARCH	0.771	2.477	4.207	0.579	2.291	3.972
Asymm Beta-t-EGARCH	0.745	2.474	4.251	0.578	2.296	4.006
VIX	0.847	2.493	4.143	0.747	2.395	3.983
HAR-RV	0.684	2.401	4.205	0.516	2.235	3.963

C Other loss functions for machine learning models

Table 8: Hybrid models predictions evaluation table using MAE

Model	1 day	5 days	21 days
XGBoost + Asymm Garch	1.30	4.06	18.21
XGBoost + Asymm Beta-t-EGarch	1.30	3.87	18.24
Random Forest + Asymm Garch	1.37	4.31	20.62
Random Forest + Asymm Beta-t-EGarch	1.34	4.16	20.09
LSTM + Asymm Garch	1.26	4.00	18.61
LSTM + Asymm Beta-t-EGarch	1.22	3.75	18.82

Note: Here target variance computed from daily returns is used as a target for the MAE.

Table 9: Hybrid models predictions evaluation table using QLIKE

Model	1 day	5 days	21 days
XGBoost + Asymm Garch	0.541	2.331	4.116
XGBoost + Asymm Beta-t-EGarch	0.580	2.324	4.119
Random Forest + Asymm Garch	0.888	2.357	4.165
Random Forest + Asymm Beta-t-EGarch	0.846	2.344	4.133
LSTM + Asymm Garch	405039.889	2.367	4.260
LSTM + Asymm Beta-t-EGarch	35112.344	2.304	4.418

Note: Here target variance computed from daily returns is used as a target for the QLIKE.

D Diebold-Mariano test results for longer prediction horizons

Table 10: Diebold-Mariano tests results for MSE comparison with econometric models 5 days

	VIX	HAR-RV	Asymm GARCH	Asymm Beta-t-EGARCH
VIX	-	-	-	-
HAR-RV	-1.89*	-	-	-
Asymm GARCH	0.19	0.90	-	-
Asymm Beta-t-EGARCH	-2.11**	-0.53	-1.22	-
XGBoost + Asymm Garch	-2.41**	-1.86*	-2.51**	-2.09**
XGBoost + Asymm Beta-t-EGarch	-2.31**	-1.65*	-2.19**	-1.83*
Random Forest + Asymm Garch	-2.36**	-1.61	-2.05**	-1.78*
Random Forest + Asymm Beta-t-EGarch	-2.39**	-1.58	-2.07**	-1.86*
LSTM + Asymm Garch	-2.14**	-1.46	-2.25**	-1.75*
LSTM + Asymm Beta-t-EGarch	-2.32**	-1.41	-1.90*	-1.69*

Note: The results displayed in the table are DM test statistics

Table 11: Diebold-Mariano tests results for MSE comparison with econometric models 21 days

	VIX	HAR-RV	Asymm GARCH	Asymm Beta-t-EGARCH
VIX	-	-	-	-
HAR-RV	-1.17	-	-	-
Asymm GARCH	2.26**	2.47**	-	-
Asymm Beta-t-EGARCH	-2.00**	-0.06	-2.37**	-
XGBoost + Asymm Garch	-3.41***	-2.40**	-3.11***	-2.70***
XGBoost + Asymm Beta-t-EGarch	-3.26***	-2.25**	-3.06***	-2.54**
Random Forest + Asymm Garch	-1.03	-0.54	-2.45**	-0.53
Random Forest + Asymm Beta-t-EGarch	-0.89	-0.41	-2.45**	-0.41
LSTM + Asymm Garch	-1.01	-0.13	-2.27**	-0.16
LSTM + Asymm Beta-t-EGarch	0.39	0.95	-1.99**	1.49

Note: The results displayed in the table are DM test statistics

E Diebold-Mariano test results for machine learning models

Table 12: Diebold-Mariano tests results for MSE comparison with combined ML + econometric models 1 day

	XGBoost + Asymm Garch	XGBoost + Asymm Beta-t-EGarch	Random Forest + Asymm Garch	Random Forest + Asymm Beta-t-EGarch	LSTM + Asymm Garch	LSTM + Asymm Beta-t-EGarch
XGBoost + Asymm Garch	-	-	-	-	-	-
XGBoost + Asymm Beta-t-EGarch	-0.68	-	-	-	-	-
Random 2016t + Asymm Garch	-0.39	-0.22	-	-	-	-
Random chent + Asymm Beta-t-EGarch	-0.53	-0.37	-0.11	-	-	-
LSTM + Asymm Garch	-1.17	-1.13	-1.83*	-1.94*	-	-
LSTM + Asymm Beta-t-EGarch	-1.31	-1,29	-1.67*	-2.14**	-0.02	-

Note: The results displayed in the table are DM test statistics

F MSE over time for all econometric models for different prediction horizons

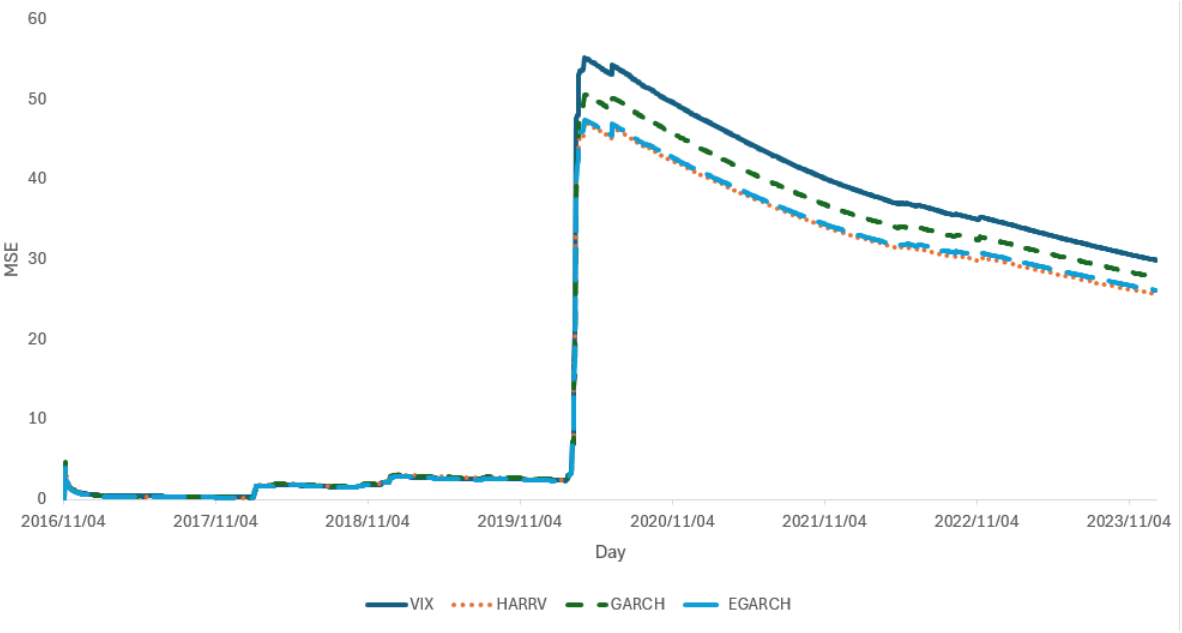

Figure 2: MSE over time for 1-day predictions using daily returns

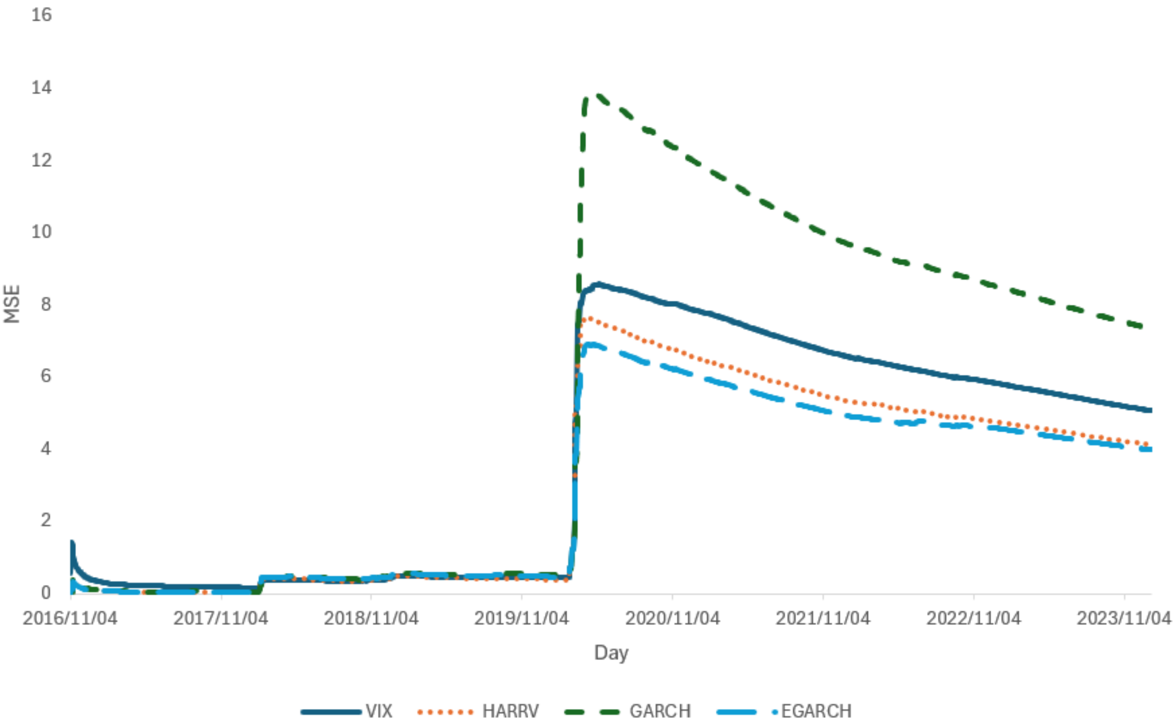

Figure 3: MSE over time for 1-day predictions using intraday returns

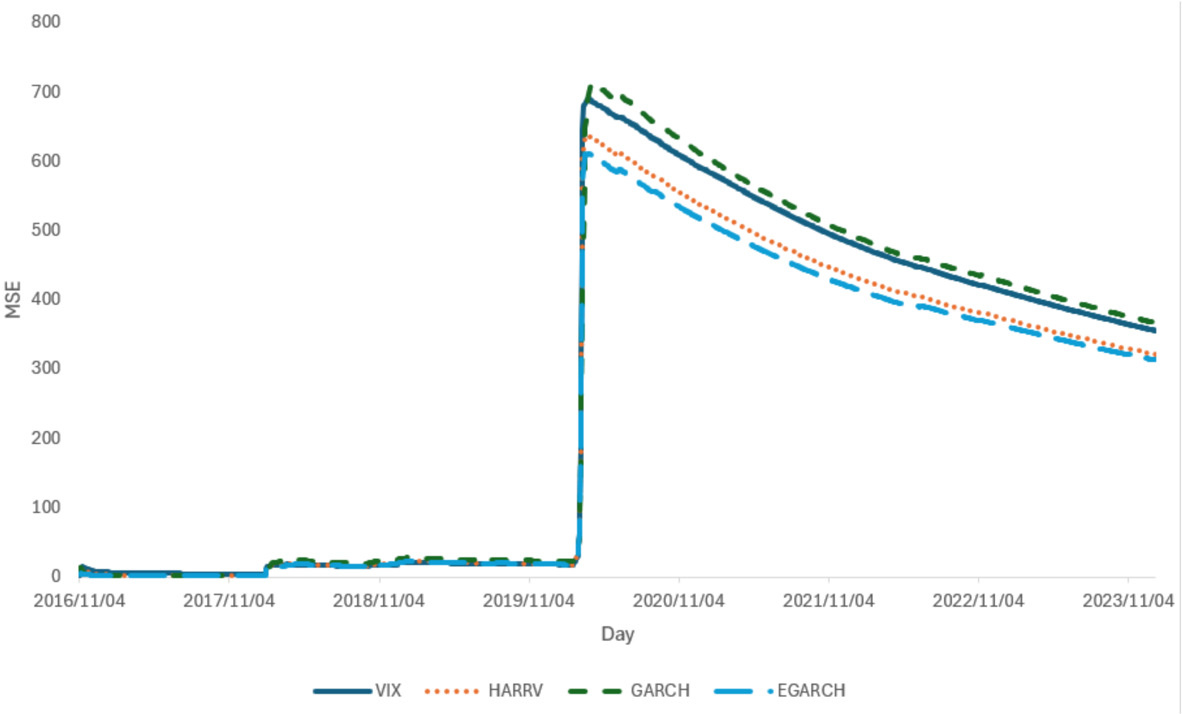

Figure 4: MSE over time for 5-day predictions using daily returns