ECOM107 - Dissertation

Queen Mary University of London

School of Economics and Finance

Long-term Volatility Forecast with Time Series Models

Tuo Sa (ID: 210726231)

Date: August 21, 2023
Word Count: 6180
Abstract Word Count: 177
Supervisor: Konstantinos Theodoridis

Contents

Abstract

Most previous studies have focused on volatility forecasting for a single asset using a limited number of models. In contrast, this research offers a comprehensive comparison of 21 models, spanning GARCH-type models, HAR-type models, Machine Learning models, Neural Networks, and hybrid models, with a focus on long-term volatility forecasting. The dataset encompasses daily price data for 8 cryptocurrencies, ranging from 01-10-2015 to 15-08-2023. The primary feature considered is the historical volatility over a 30-day timeframe, and the forecast horizon is set at 240 days. To gauge performance, 3 out-of-sample metrics and 3 in-sample metrics have been employed. Additionally, the efficiency, orthogonality, and sensitivity of each model have been tested. Notably, the main findings indicate that simple classification models surpass other model categories in performance, delivering both accurate and robust outcomes. It underscores the need for financial practitioners to rethink over-relying on more complex models like deep learning when simpler models, particularly certain Machine Learning models, can provide more robust and accurate forecasts. The codes of this research have been uploaded to: https://github.com/Niacin233/Dissertation

1 Introduction

Research on volatility forecasting resonates with practical aspects of the financial sector, where accurate volatility forecasting is paramount for risk management, portfolio optimization, and trading strategies, etc (Poon and Granger, 2003). Volatility is a critical component in the evaluation of investment risk, though it is not the same as risk itself. An accurate forecast of asset price volatility enables investors and portfolio managers to make informed decisions regarding their risk tolerance and asset allocation. By incorporating volatility into their decision-making process, investors can better manage their exposure to risk and optimize their investment strategies. In the field of derivative valuation, volatility also plays a key role. Accurate volatility forecasting is essential for option pricing, such as with the Black-Scholes-Merton model, which incorporates volatility as a key input in computing the fair value of an option (Black and Scholes, 1973). From the perspective of financial institutions, a crucial component of risk management is volatility forecasting, which enables institutions to estimate potential losses and allocate reserve capital accordingly. Furthermore, central banks and policymakers utilize market estimates of volatility as an indicator of the vulnerability of financial markets and the economy. Volatility forecasts can inform monetary policy decisions and assist policymakers in assessing the stability of the macro-economy and financial system.

Statistically, volatility is a special case of time series. The complexity, non-linearity, and low signal-to-noise ratio of financial markets makes short-term asset return prediction nearly impossible (Gu et al., 2020). However, various stylized facts, such as the persistence, mean-reversion, and asymmetry, imply that volatility is largely predictable.

At early stage, the researchers adopted simple time series models, for instance EWMA (Morgan et al., 1996), to forecast volatility. Later research started to focusing on the stylized facts of volatility. Three representatives of these studies are (G)ARCH (Engle, 1982; Bollerslev, 1986), Stochastic Volatility (Hull and White, 1987), and HAR (Corsi, 2009). Following these studies, extensions of the models were developed to capture more stylized facts.

Another trend in volatility forecasting is machine learning models. Some research has suggested that machine learning methods have superiority over traditional models (Filipović and Khalilzadeh, 2021; Christensen et al., 2021), especially the deep learning models with ability to handle sequential data. However, their complexity and computational demands have sparked debates about their necessity and some studies argue that simpler models can achieve comparable performance. For instance, Elsayed et al. (2021) found that a simple Gradient Boosting Regression Tree (GBRT) model can outperform several deep neural network models. Meanwhile, deep learning models often fail to capture temporal dependencies, affecting their forecasting performance (Ughi et al., 2023). Therefore, while

machine learning models show promise in time series forecasting, it's crucial to consider their limitations and the potential benefits of simpler models.

This dissertation aims to conduct a comprehensive comparison of these traditional econometric models and machine learning models in the context of long-term volatility forecasting. The goal is to identify the strengths and weaknesses of each approach.

The introduction section explains the uniqueness of volatility data and defines the criteria of a good volatility model. The next section introduces the development of volatility models, followed by the mathematical definition of the models and performance measurements in methodology section. The Empirical Analysis demonstrates the performance metrics and results with detailed interpretation. Finally, all the results lead to conclusions in the last section.

1.1 Properties of volatility

The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots in figure1 can provide insights into the underlying structure of the data series, especially in the context of time series forecasting and determining the order of autoregressive (AR) processes (Hamilton, 2020). The ACF plot shows how a series is correlated with its past values. If the ACF tails off gradually and becomes non-significant (inside the blue shaded region) after a certain number of lags, it indicates that the series might have an autoregressive structure. On the other hand, the Partial Autocorrelation Function (PACF) measures the correlation between observations of a time series separated by k time units, removing the effect of any correlations due to the terms at shorter lags.

The ACF for Bitcoin's volatility shows a slow decay, indicating that there's a long memory in the series. It suggests shocks to volatility can have lasting effects. The PACF has a few significant spikes at the beginning and then tapers off, indicating potential autoregressive terms. Most other cryptocurrencies display a similar pattern in their ACF and PACF plots, suggesting that their volatilities might be driven by similar underlying processes. The long tail in the ACF plot and the significant lags in the PACF for most cryptocurrencies imply that past volatilities have an effect on current volatility. This is a common phenomenon in financial time series known as "volatility clustering" or "persistence", where periods of high volatility tend to be followed by periods of high volatility, and low volatility periods are followed by low volatility.

The other facts about volatility includes stationarity, distribution, and reaction to shocks. A Augmented Dickey-Fuller (ADF) test, a Jarque-Bera test, and a regression analysis were implemented to further dive into the facts (table 1). The results from the Augmented Dickey-Fuller (ADF) test indicate that the 30-day historical volatility for each cryptocur-

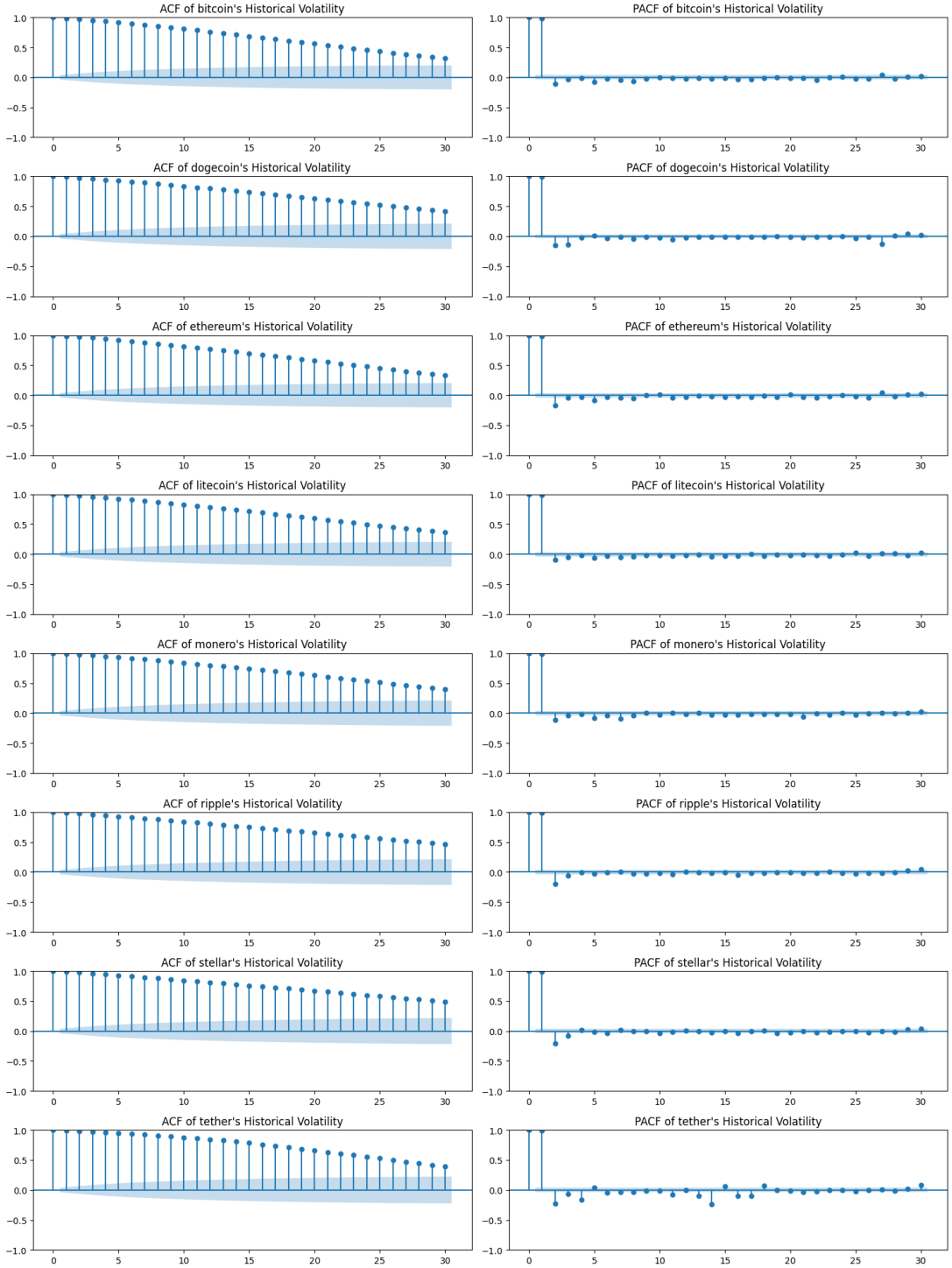

Figure 1: ACF and PACF of realized volatility

Table 1: Results of test for stylized facts

Cryptocurrency	ADF Test p-value	Asymmetry Coefficient	Asymmetry p-value	Jarque-Bera Test p-value
bitcoin	1.67604e-07	-0.00711353	0.34891	0
dogecoin	4.53668e-09	0.0553199	1.01244e-06	0
ethereum	2.65258e-09	0.00685229	0.353462	0
litecoin	1.17129e-06	0.011141	0.164065	0
monero	1.39444e-06	0.0086405	0.297857	0
ripple	1.61131e-05	0.0384326	0.000174259	0
stellar	2.36599e-05	0.0441391	2.89431e-06	0
tether	1.05401e-11	-0.0350414	0.039689	0

rency is stationary, as the p-values are very close to zero and below 0.05 significance level. This suggests that the volatilities exhibit mean reversion. The Jarque-Bera (JB) test checks if the returns are normally distributed. If the test rejects the null hypothesis of normality, it suggests that the returns might have heavy tails. The results from for Heavy Tails show that the p-values for all the cryptocurrencies are effectively zero, which strongly reject the null hypothesis for the log returns of all the cryptocurrencies. This means that extreme values (either very high or very low) are more likely to occur than what would be expected under a normal distribution. This heavy-tailed nature is a common characteristic of financial return data and indicates a higher likelihood of large price changes (jumps or crashes). To test the existence of leverage effect, namely the different impacts caused by positive and negative shock, the volatility will be regressed on lagged return. The coefficient of the lagged returns of Bitcoin, Litecoin, Monero, and Ethereum are of p-value greater than 0.05, indicating the presence of the leverage effect. Bitcoin and Tether show a negative coefficient, suggesting negative returns tend to increase volatility more than positive returns. Contrarily, Litecoin, Monero, and Ethereum show a positive coefficient. This means that positive returns might have a more pronounced effect on increasing volatility compared to negative returns. However, the effect is not typical of the traditional leverage effect observed in stock markets, where negative shocks have larger impacts for most of the securities (Engle and Patton, 2001).

1.2 Standard of a good volatility model

The discussion above is aligned to the research of Engle and Patton (2001), which highlights the stylized facts about volatility:

- persistence: Volatility tends to be auto-correlated, implying that periods of high volatility often lead to more high volatility days, while low volatility phases tend to persist..
- mean-reversion: Despite the fact that volatility appears to have quite long memory, it will eventually return to its mean over the long-term.
- Asymmetry: Negative returns (market downturns) tend to lead to more volatility than positive returns (market upturns) of the same magnitude.Patton and Sheppard, 2015
- Heavy Tails: Financial returns often exhibit "heavy tails" - that is, they are more likely to experience extreme outcomes than would be predicted by a normal distribution.

Engle and Patton (2001); Patton and Sheppard (2015) suggested a good volatility model should be able to capture the stylized facts and exhibit long memory to facilitate long-term forecasts. McAleer and Medeiros (2011) also attributed the failure of many latent volatility models to not adequately describing several stylized facts observed in financial time series. Given the widespread use of volatility in risk management, it is also crucial for a volatility model to anticipate volatility in high volatility period. (Bali, 2003;Chou, 2005;Ding et al., 2019). Most importantly, the model should perform robust forecast (Kristjanpoller et al., 2014; Poon and Granger (2003)).

In brief, the criteria for a good volatility model is:

- out-of-sample forecast performance
- incorporation of stylized facts
- long-term memory
- robust outcomes
- high volatility period forecasting performance

2 Literature Review

2.1 Econometrics Models

The discovery of ARCH effect was a milestone of volatility studies. The ARCH effect, which stands for "Autoregressive Conditional Heteroskedasticity," refers to a phenomenon in time series data where the variance of the residuals is not constant over time but depends on past values of the error terms. The term "heteroskedasticity" itself means "varying variance," and in the context of the ARCH effect, the variance is conditional on past information. The ARCH effect holds significant importance in financial econometrics. It is commonly observed that substantial fluctuations in asset returns are succeeded by similarly large fluctuations in either direction, while minor variations typically follow other minor variations. This pattern is referred to as "persistence".

The ARCH model, introduced by Engle (1982), can be represented as:

$$r_t = \sigma_t e_t \tag{1}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + ... + \alpha_q r_{t-q}^2 \tag{2}$$

Where r_t is the return at time t, σ_t^2 is the conditional variance of the error term at time t, e_t is a white noise error term with zero mean and unit variance, $\alpha_0, \alpha_1, ...$ are parameters to be estimated, q is the number of lags in the model.

More importantly, by the ARCH effect, volatility is proven to be predictable by its past values. It can be derived from equation 1, that the conditional expectation of r_t is

$$\begin{aligned} \mathbf{E}(r_t|r_{t-1}, r_{t-2}, ...) &= \mathbf{E}(\sigma_t e_t|r_{t-1}, r_{t-2}, ...) \\ &= \sigma_t \mathbf{E}(e_t|r_{t-1}, r_{t-2}, ...) \\ &= 0 \end{aligned} \tag{3}$$

Then, the variance of conditional return is defined as:

$$\begin{aligned} &\mathrm{Var}(r_t|r_{t-1}, r_{t-2}, ...) \\ &= \mathbf{E}[(r_t - \mathbf{E}(r_t|r_{t-1}, r_{t-2}, ...))^2|r_{t-1}, r_{t-2}, ...] \\ &= \mathbf{E}(r_t^2|r_{t-1}, r_{t-2}, ...) \end{aligned} \tag{4}$$

Similarly,

$$\begin{aligned}
&\text{Var}(e_t|r_{t-1}, r_{t-2}, ...) \\
&= \mathbf{E}[(e_t - \mathbf{E}(e_t|r_{t-1}, r_{t-2}, ...))^2|r_{t-1}, r_{t-2}, ...] \\
&= \mathbf{E}(e_t^2|r_{t-1}, r_{t-2}, ...) \\
&= 1
\end{aligned} \tag{5}$$

since $e_t \sim WhiteNoise(0, 1)$.

Then, from equation 4 and 5,

$$\begin{aligned}
&\text{Var}(r_t|r_{t-1}, r_{t-2}, ...) \\
&= \mathbf{E}(\sigma_t^2 e_t^2|r_{t-1}, r_{t-2}, ...) \\
&= \sigma_t^2 \mathbf{E}(e_t^2|r_{t-1}, r_{t-2}, ...) \\
&= \sigma_t^2
\end{aligned} \tag{6}$$

which suggests that the conditional variance is a function of past values due to equation 2.

Engle (1982) also raised a Lagrange Multiplier (LM) test to distinguish whether a time series exhibits ARCH effect. Start by fitting a model to the data and extracting the residuals from the fitted model. These residuals should ideally represent white noise if the model is correctly specified. Followed by regressing the squared residuals on a constant and a number of its own lagged values.

$$\epsilon_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + ... + \alpha_q \epsilon_{t-q}^2 + u_t$$

The test statistic for the LM test is given by:

$$T \times R^2$$

Where T represent the sample size and R^2 denote the coefficient of determination obtained from regressing squared residuals against their lagged values.

The test statistic adheres to a chi-squared distribution with q degrees of freedom. Should the test statistic be notably large (in comparison to the chi-squared threshold values), we would dismiss the null hypothesis, which posits no presence of the ARCH effect, thereby indicating that the time series displays the ARCH effect.

Bollerslev (1986) generalized ARCH model by defining the variance of the error term at

time t as a function of both past squared error terms and past conditional variances.

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{7}$$

It captures both the persistence in volatility due to past shocks and the persistence due to past volatility. This makes GARCH models particularly useful for financial time series that exhibit long-lasting effects of shocks on volatility.

Furthermore, Engle and Patton (2001) proposed that a good volatility should be able to capture the stylized facts, persistence, mean-reversion, Asymmetry, and Heavy Tails, of volatility. Their research strongly underpinned GARCH-type models, which suggested that GARCH models capture the persistence in volatility by incorporating lagged volatility terms in the model, the mean-reverting behavior of volatility by including a constant term in the model, the asymmetric impact of negative and positive return innovations on volatility by including separate parameters for positive and negative return innovations, and the relationship between exogenous variables that may have a significant influence and volatility by including these variables in the model. There are plenty of extensions of the GARCH model, this research adopted some of the most commonly used of them, such as EGARCH (Nelson, 1991), GJR-GARCH (Glosten et al., 1993), FIGARCH (Baillie et al., 1996), and RSGARCH (Gray, 1996).

However, GARCH-type models, being short-memory models, cannot capture long-memory behavior effectively. When aggregated over extended periods, they tend to converge rapidly to their unconditional mean, rendering them inappropriate for longer forecasting horizons. However, by using fractional difference operators, FIGARCH captures long memory in a more succinct manner. Corsi (2009) argued that Fractional integration, while mathematically convenient, lacks a clear economic interpretation; using the fractional difference operator in FIGARCH models necessitates an extended build-up phase, potentially leading to the omission of numerous observations; FIGARCH might not capture the multiscaling behavior observed in empirical data and developed the Heterogeneous Autoregressive Model, which captures volatility dynamics by using lagged variances at different frequencies (daily, weekly, monthly). HAR has been a cornerstone in the financial econometrics literature due to its ability to capture the long memory properties of volatility. Over the years, various extensions and modifications of the HAR model have been proposed to capture different features of the volatility dynamics, such as volatility jumps (Andersen et al., 2007; Corsi and Renò, 2012), non-parametric components (Patton and Sheppard, 2015), and quadratic variation (Bollerslev et al., 2016).

2.2 Machine Learning Models

Machine learning algorithms are celebrated for their adeptness in deciphering intricate data relationships, leading to their extensive application across diverse domains. The research by Lu et al. (2022) adopted an exhaustive methodology, probing oil futures volatility through an array of machine learning models. This spectrum encompasses traditional models, mainstream machine learning models, and neural network configurations. Empirical data underscores the efficacy of machine learning models in recognizing non-linear inter-dependencies among variables. Further, they can adeptly perform variable selection via regularization and cross-validation, sidestepping potential over-fitting challenges.

Another study (Christensen et al., 2021) compared machine learning approaches to the Heterogeneous Autoregressive (HAR) model, it was found that off-the-shelf machine learning implementations consistently produced better one-day-ahead forecasts of realized variance. These machine learning algorithms, including regularization, tree-based algorithms, and neural networks, were able to extract more information from exogenous predictors of volatility. Additionally, the machine learning methods showed robustness even when a large number of irrelevant variables were added to the information set. Overall, machine learning algorithms have demonstrated their ability to handle the complex nature of financial markets and provide more accurate volatility forecasts compared to traditional models. Similar studies underpinned these conclusions (McAleer and Medeiros, 2011; Rahimikia and Poon, 2020; Shen et al., 2021)

Apart from purely deploying machine learning models, combining traditional models with machine learning models have been proven to provide more accurate forecasting results. Kristjanpoller et al. (2014), Kim and Won (2018), Hajizadeh et al. (2012), and Seo and Kim (2020) constructed hybrid models by using forecasts of GARCH-type models as predictors in machine learning models. By combining the strengths of different models, hybrid models can capture complex patterns in the data that might be missed by individual models. The flexibility of hybrid models enable them to be tailored to specific datasets or financial instruments. Moreover, the biases of one model can be offset by the other, leading to more robust predictions. However, the complexity of hybrid models raises the risk of of over-fitting and making it challenging to understand the underlying factors driving the forecasts.

Despite of the advantages of the machine learning methods, Elsayed et al. (2021) proved that, with proper feature engineering, Gradient Boosting Regression (GBR) model can surpass many state-of-the-art deep neural network (DNN) models. In another study on transformer model (Ughi et al., 2023), a deep learning model driven by self-attention mechanism, the authors demonstrated that a simplified model produced better outcomes. It raises the doubt on the robustness and effectiveness of time series forecast by deep

learning models.

3 Methodology

3.1 Data

The dataset includes the daily price of 8 cryptocurrencies from 01-10-2015 to 15-08-2023 from CoinGecko. The horizon is set to be 240-day. Since volatility is a latent variable, it is important to choose a proper indicator for it. Davidian and Carroll (1987) proved that variance of return is not robust when the distribution differs from normal distribution and a volatility forecast based on logarithmic return turns out to be more robust. Moreover, Figlewski (1997) showed that switching from monthly data to daily data with a horizon of 24-month increases the forecast error by twice, which suggested a relative decrease of sampling period in calculating volatility to forecast horizon degrades the forecast performance. Therefore, a 30-day history volatility is chosen as the indicator of latent volatility. The dataset is split into a training set (before 21-08-2022) and a testing set (from 21-08-2022 to 15-08-2023). Period with volatility 1.28 times higher than it standard deviation is recognised as high volatility period.

3.2 Metrics

This research adopts 3 out-of-sample metrics and 3 in-sample metrics to measure the ability to explain and forecast the data. Since the dataset contains the data of multiple cryptocurrencies, each model will be fitted to every cryptocurrency, and the average metrics over all the cryptocurrencies will be computed as the results of the model

3.2.1 Statistical Inference

The out-of-sample performance is measured by (Poon and Granger, 2003; Brooks, 2019):

1. Root Mean Square Error (RMSE): The Root Mean Square Error is the square root of the Mean Square Error. It measures the standard deviation of the residuals.

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

2. Mean Absolute Error (MAE): The Mean Absolute Error is the average of the absolute differences between the observed and predicted values.

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

3. Mean Absolute Percent Error (MAPE): The Mean Absolute Percent Error is the average of the absolute percent differences between the observed and predicted values.

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

In the above equations, y_i, $\hat{y}_i$, and n respectively denote the actual value, the predicted value, and the total number of observations. All these out-of-sample metrics measure the error of the model forecast. A lower value indicates a more accurate forecast.

The in-the-sample performance is measured by (Brooks, 2019):

1. Log-Likelihood: The log-likelihood is the logarithm of the likelihood function of a model. The likelihood function measures the goodness of fit of a statistical model to the data. A higher log-likelihood value indicates that the model explains the observed data better.

$$\ln(L) = \sum_{i=1}^{n} \ln(f(x_i;\theta))$$

where $f(x_i;\theta)$ is the probability density function for observation x_i and model parameters θ.

The residuals are the differences between the observed values and the predicted values:

$$r_i = y_i - \hat{y}_i$$

The variance of the residuals is given by:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} r_i^2$$

If we assume the residuals are normally distributed with mean 0 and variance σ^2, the probability density function for a normal distribution is:

$$f(r_i;\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r_i^2}{2\sigma^2}}$$

By plugging this into the log-likelihood expression, breaking down the logarithm,

and summing over all observations, the expression of Log-Likelihood becomes:

$$\ln(L) = (-n/2)\ln(2\pi) - (n/2)\ln(\sigma^2) - (1/2\sigma^2)\sum_{i=1}^{n} r_i^2$$

2. Akaike Information Criterion (AIC): The Akaike Information Criterion (AIC) evaluates the relative merit of a statistical model based on a specific dataset. While assessing the model's goodness of fit, it also considers the number of parameters needed to achieve this fit, imposing a penalty for each additional parameter. Among competing models, the model with the smallest AIC is favored, indicating an optimal trade-off between fitting accuracy and model intricacy.

$$AIC = 2k - 2\ln(L)$$

where k is the number of model parameters and L is the maximum value of the likelihood function for the model.

3. Schwarz Criterion (BIC): The Schwarz Criterion, commonly referred to as the Bayesian Information Criterion (BIC), serves as a yardstick for choosing the best model from a limited collection of models. Rooted in the likelihood function, it incorporates a penalty for the quantity of parameters present in the model. A lower BIC value indicates a better model.

$$BIC = \ln(n)k - 2\ln(L)$$

where n is the number of observations.

3.2.2 Sensitivity Analysis

Sensitivity analysis serves as a critical tool for assessing the robustness and consistency of a model by probing the extent to which alterations in input parameters influence the volatility forecast (Kristjanpoller et al., 2014). Essentially, this analysis seeks to gauge the reactivity of forecasts to variations in model parameters, input datasets, or foundational assumptions. By individually adjusting elements like the volatility time frame (7-day, 14-day, 30-day) and forecast horizon (120-day, 240-day, 360-day), the subsequent shifts in forecast performance can be monitored and interpreted, providing a clearer understanding of the model's robustness.

3.2.3 Forecast Efficiency and Orthogonality

Apart from the statistical metrics, another way of evaluation is the forecast efficiency (Hansen and Hodrick, 1980). The basic idea is to regress the actual values on the forecast values and test whether the slope of the regression line is 1 and the intercept is 0. If the slope is significantly different from 1 or the intercept is significantly different from 0, it suggests that the forecast is not efficient. The regression model can be represented as:

$$y_t = \alpha + \beta \hat{y}_t + \epsilon_t$$

where y_t is the actual value, $\hat{y}_t$ is the forecast value, α is the intercept, β is the slope, and ϵ_t is the error term.

Based on the method, Fair and Shiller (1990) further developed a orthogonality test, which is used to test whether the forecast errors are uncorrelated with the the forecast of the other model. If the errors are orthogonal, it suggests that the test model does not contain additional information that the other model captured. The orthogonality condition can be tested by regressing the forecast errors of one model on the prediction of the other model and testing whether the coefficients are statistically different from zero. The regression model can be represented as:

$$\epsilon_t = \gamma \hat{y}_t + u_t$$

By this method, information content for each fitted model can be quantified.

3.3 Models

This research compares GARCH-type models, HAR-type models, Machine Learning models, and Neural Networks. The forecasts are estimated by iterative one-day-ahead forecasting.

3.3.1 Baseline

If someone is required to guess the volatility in a long term, given the fact that volatility is stationary, the trivial answer would be the historical average. Because the distribution of variance is constant over all the period. Therefore, the baseline model adopts the in-sample mean as all the out-of-sample forecast.

3.3.2 Generalized Autoregressive Conditional Heteroskedasticity Models

1. **GARCH (Generalized Autoregressive Conditional Heteroskedasticity):**

   The GARCH(p, q) model generalizes the ARCH model by incorporating lagged

values of the conditional variance itself in addition to the squared returns. The specification of the model is (Brooks, 2019):

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

where σ_t^2 is the conditional variance of the return series, ϵ_t are the innovations, and the parameters α_0, α_i, and β_j need to be estimated.

2. **GJR-GARCH (Glosten-Jagannathan-Runkle Generalized Autoregressive Conditional Heteroskedasticity):**

   The GJR-GARCH model is an extension of the standard GARCH model that allows for different reactions to positive and negative innovations (asymmetry). The specification of the GJR-GARCH(p, q) model is:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \gamma \epsilon_{t-1}^2 I_{\{\epsilon_{t-1}<0\}} + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

   where I is an indicator function that equals 1 if the condition in the braces is true, and 0 otherwise.

3. **FIGARCH (Fractionally Integrated GARCH):**

   The FIGARCH model allows for a long memory process in the volatility. The traditional GARCH model assumes a short memory process, meaning that past shocks have an exponentially decreasing effect on current volatility. In the FIGARCH model, the effect declines at a hyperbolic rate, allowing for a more persistent impact.The FIGARCH(d, p, q) model can be represented as:

$$(1 - L)^d \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

   where L is the lag operator, and d is the fractional differencing parameter, allowing for long memory behavior.

3.3.3 Heterogeneous Autoregressive Models

The HAR framework is flexible, and researchers have introduced many extensions to capture various features of the volatility dynamics (Corsi, 2009; Corsi and Renò, 2012; Bollerslev et al., 2016). The coefficients (like $\beta_d, \beta_w, \beta_m$) are estimated using regression techniques, and they provide insights into the relative importance of daily, weekly, and monthly volatilities in predicting future volatility.

1. **HAR:**

   The basic HAR model captures the long memory property of volatility by incorporating lagged values of daily, weekly, and monthly historical volatilities.

$$V_t = c + \beta_d V_{t-1}^{(d)} + \beta_w V_{t-1}^{(w)} + \beta_m V_{t-1}^{(m)} + \epsilon_t$$

   Where V_t is the volatility at time t, $V_{t-1}^{(}d)$ is the daily historical volatility, $RV_{t-1}^{(w)}$ is the weekly historical volatility, $RV_{t-1}^{(m)}$ is the monthly historical volatility, ϵ_t is the error term.

2. **HAR-J (HAR with Jumps):**

   This model captures the jump components. The typical representation of the HAR-J model is:

$$V_t = c + \beta_d V_{t-1}^{(d)} + \beta_w V_{t-1}^{(w)} + \beta_m V_{t-1}^{(m)} + \gamma J_{t-1} + \epsilon_t J_t = V_t - BV_t BV_t = r_t \times r_{t-1}$$

   Where J_{t-1} is the jump component at time $t-1$, BV_t is the bipower variation at time t.

3. **HAR-Q (HAR with Quadratic Variation):**

   The HAR-Q model incorporates the Quadratic Variation (QV) to capture non-linear features in volatility. The typical representation of the HAR-Q model is:

$$V_t = c + \beta_d V_{t-1}^{(d)} + \beta_w V_{t-1}^{(w)} + \beta_m V_{t-1}^{(m)} + \gamma QV_{t-1} + \epsilon_t$$

   Where QV_{t-1} is the Quadratic Variation at time $t-1$.

3.3.4 Machine Learning Models

In addition to traditional statistical and econometric models, this research also explores machine learning methods for forecasting historical volatility. Given the non-linear and complex nature of financial time series data, machine learning models can capture intricate patterns in the data, making them a viable alternative for volatility prediction. Below, we delve into the mathematical formulations of some of these methods:

1. **K-Nearest Neighbors (KNN)**:

   KNN is a non-parametric algorithm that predicts the output of a new instance based on the outputs of its k nearest instances from the training set. The standard

formulation for KNN in the context of regression is:

$$\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

Where: - $N_k(x)$ is the set of k points in the training data closest to x, - y_i is the output of the i^{th} training instance.

2. **Support Vector Regression (SVR)**:

SVR is a regression method that aims to find a hyperplane that best fits the data, while ensuring that deviations beyond a certain margin are minimized. The standard formulation of SVR is:

$$y(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \phi(x_i) \cdot x + b$$

Subject to:

$$y_i - \sum_{j=1}^{n} (\alpha_j - \alpha_j^*) \phi(x_j) \cdot x_i - b \leq \epsilon + \xi_i$$

$$\sum_{j=1}^{n} (\alpha_j - \alpha_j^*) \phi(x_j) \cdot x_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\alpha_i, \alpha_i^* \geq 0$$

Where: - ϕ is a kernel function, - α_i and α_i^* are Lagrange multipliers, - ξ_i and ξ_i^* are slack variables, - ϵ is the margin of tolerance.

3. **Random Forest**:

A decision tree is a flowchart-like classification model in which each internal node represents a feature (or attribute), each branch represents a decision rule, and each leaf node represents an outcome. Creating a decision tree involves selecting attributes that return the highest information gain, which is defined as:

$$IG(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v) E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

Where p_+ is the proportion of positive examples in S, p_- is the proportion of negative examples in S, $IG(S, A)$ is the information gain of attribute A for dataset S, $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset

of S for which attribute A has value v.

The Random Forest is an ensemble learning method by creating multiple decision trees during training and outputs the mean prediction of the individual trees for regression tasks, and mode of the individual trees for classification tasks. The formulation for the output of the Random Forest is:

$$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$

Where: - B is the number of trees in the forest, - $T_b(x)$ is the prediction of the b^{th} tree.

4. **Gradient Boosting Regression**:

   Gradient Boosting Regression builds an additive model in a forward stage-wise fashion; it generalizes by allowing an arbitrary differentiable loss function. The prediction model in the form of an ensemble of weak prediction models, typically decision trees, is formulated as:

$$\hat{y}(x) = \sum_{b=1}^{B} \rho_b T_b(x)$$

   Where: - B is the number of boosting stages, - $T_b(x)$ is the prediction of the b^{th} weak learner, - ρ_b is the learning rate.

   At each boosting stage, the algorithm fits the negative gradient (called pseudo-residuals) of the given loss function to the current model.

3.3.5 Deep Learning Models

Deep learning, a branch of machine learning, employs algorithms with many layers (hence "deep") to analyze various factors of data. The inherent capacity of deep learning models to capture intricate patterns in large datasets makes them especially suitable for forecasting tasks like volatility prediction. In this research, 3 commonly used neural networks are deployed and the number of layer, number of neuron, and activation function are decided by cross validation (Ketkar and Santana, 2017).

1. **Fully Connected Neural Network (FCNN)**:

   An FCNN consists of multiple layers of neurons, where every neuron in a layer is connected to every neuron in the subsequent layer. Given an input vector x, the

output of the l^{th} layer can be defined as:

$$h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)})$$

Where $h^{(l)}$ is the output of the l^{th} layer, $W^{(l)}$ is the weight matrix for the l^{th} layer, $b^{(l)}$ is the bias vector for the l^{th} layer, σ is an activation function (e.g., ReLU, Sigmoid, Tanh).

2. **Convolution Neural Network (CNN)**:

   CNN is specifically designed to process grid-structured data (like images). A typical CNN comprises alternating convolutional and pooling layers, followed by fully connected layers. The convolutional layers aim to learn local patterns in the input through the convolution operation. In contrast, pooling layers condense the data, retaining only essential information. The final fully connected layers interpret these features to make a decision (in classification) or a prediction (in regression or forecasting).

   The convolution operation for layer l on input x can be represented as:

$$h_i^{(l)} = \sigma\left(\sum_j W_j^{(l)} \cdot x_{i-j} + b^{(l)}\right)$$

   Where $h_i^{(l)}$ is the output at position i of the l^{th} layer, $W_j^{(l)}$ is the weight at position j of the filter in layer l, $b^{(l)}$ is the bias for the l^{th} layer.

   Pooling layers are introduced to reduce spatial dimensions while retaining significant information. The most common form is max-pooling:

$$p_i^{(l)} = \max\left(h_i^{(l)}, h_{i+1}^{(l)}, ..., h_{i+k-1}^{(l)}\right)$$

   Where$p_i^{(l)}$ is the output of the pooling operation in the l^{th} layer at position i, k is the pooling window size.

   The final fully connected layers are identical to a FCNN.

3. **Long Short-Term Memory (LSTM)**:

   LSTM is a type of recurrent neural network (RNN) architecture that are well-suited for time series forecasting due to their capability to capture long-term dependencies. For a given input sequence x, the LSTM updates its cell state C_t and hidden state h_t as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(Forget Gate)} \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Input Gate)} \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Output Gate)} \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned}
$$

Where: - f_t, i_t, o_t are the forget, input, and output gates respectively, - $\tilde{C}_t$ is the candidate cell state, - W matrices and b vectors are learnable parameters, - $\odot$ denotes element-wise multiplication.

3.3.6 Hybrid Models

The hybrid models are designed as using the estimation of one model as one of the in put variables of another model (Kristjanpoller et al., 2014). In this research, the in-sample forecasts of basic GARCH and HAR models are deployed as explanatory variables of Machine Learning models and Neural Networks.

4 Empirical Analysis

4.1 Model Performance

The results of the performance metrics presented in Table 2 indicate that the Random Forest (RF) model was the standout, consistently ranking as the top performer across all out-of-sample metrics. This model displayed the smallest discrepancies between its forecasted values and the actual outcomes. Notably, during high volatility periods, the K-Nearest Neighbors (KNN) model showcased superior performance, especially in terms of RMSE and MAE. Additionally, the Fully Connected Neural Network (FCNN) model demonstrated remarkable fitting accuracy, boasting the best AIC and BIC scores among all contenders. Not to be overlooked, the HAR-FCNN model reported the highest log-likelihood, designating it as the best-fitting model in its cohort.

Interestingly, only a third of the models (KNN, RF, GBR, FCNN, LSTM, HAR-FCNN, GARCH-RF) surpassed the baseline across out-of-sample metrics, and merely 5 out of 21 models (KNN, RF, GBR, HAR-FCNN, GARCH-RF) did so under extreme conditions. Generally, Machine Learning models, with the exception of SVR, demonstrated superior forecasting prowess. Conversely, neural networks did not fare as well in out-of-sample metrics compared to in-sample metrics, particularly during tumultuous periods.

Of the three neural networks, the LSTM, despite its lackluster in-sample performance, delivered the most promising out-of-sample results. This highlights potential over-fitting concerns with neural networks. Hybrid models, which integrate features from various modeling paradigms, showed intermediate performance. They improved upon the econometric models they encompassed and could be a compelling choice for those seeking a balanced blend of model capabilities.

In line with the studies of Lu et al. (2022) and Christensen et al. (2021), traditional models were overshadowed by machine learning models. Although HAR models did exhibit enhanced results, they struggled to surpass the baseline. Furthermore, integrating asymmetry, long memory, jumps, and non-linear terms into GARCH and HAR models didn't result in substantial enhancements in outcomes.

4.2 Model Efficiency and Orthogonality

Table 3 provides a detailed representation of the results from the efficiency tests conducted across various models. A close inspection of the table reveals that for several models, including the HAR variants, CNN, LSTM, HAR-RF, and HAR-GBR, their intercepts display a significant deviation from 0. This suggests a potential bias inherent in their forecasting processes. Intriguingly, the HAR models, as well as the HAR-RF and HAR-GBR models, exhibit slopes nearing 0, highlighting their limited explanatory capabilities. In contrast, the RF model stands out for its remarkable efficiency. It achieves this by having a slope that closely approaches 1, while its intercept remains statistically indistinct from 0, suggesting minimal bias.

In Table 4 and Table 5, each row denotes the residual of the corresponding model. On the other hand, each column pertains to a model, particularly its forecasts serving as predictors. The intersecting cells, designated by (i, j), carry p-values derived from the regression of forecast errors from model i against the forecasts generated by model j. Typically, a $p_{<0.5}$ (p-value smaller than 0.05) is an indication that model j possesses additional forecasting information not present in model i. Namely, the forecasts from model j can explain some variability in the forecast errors of the model i, and thus,model i be missing some information captured by the model j.The presence of NaN values is a indication of colinearity between the dependent and independent variables. This mean that the forecast errors of model i can be perfectly explained or predicted by the forecasts of model j, indicating no additional information can be obtained from the errors of model i when one has the forecasts of model j.

The HAR models exhibit p-values of 0.00 when compared amongst themselves, implying that each distinct HAR model captures unique volatility features. This underscores the extensibility of HAR models. Notably, no orthogonality tests between HAR models and

Table 2: Performance Metrics (time frame = 30day, horizon = 240day).

	Out-of-Sample						In-Sample		
Model	RMSE	MAE	MAPE	Extreme RMSE	Extreme MAE	Extreme MAPE	AIC	BIC	LogLikelihood
GARCH	0.763653	0.727739	2.566556	0.694961	0.657995	1.856004	6987.788288	7011.267606	-3489.894144
GJR-GARCH	0.998787	0.949712	3.078236	0.925130	0.873908	2.403954	6989.027181	7018.376329	-3489.513590
FIGARCH	0.683895	0.649275	2.244421	0.613380	0.575281	1.709230	6989.702213	7019.051361	-3489.851106
HAR	0.462385	0.415635	1.071890	0.457626	0.408585	0.860504	-6994.449370	-6970.971581	3501.224685
HAR-J	0.463614	0.416744	1.075127	0.459735	0.410423	0.864196	-6994.019871	-6964.674547	3502.009936
HAR-Q	0.461480	0.414266	1.067400	0.460274	0.410145	0.860624	-7000.347716	-6971.002391	3505.173858
KNN	0.227361	0.184797	0.493421	0.284090	0.232061	0.412710	-8.881741	5.040815	8.440870
SVR	0.590188	0.553980	3.160660	0.554242	0.516850	2.303734	1757.559953	1771.482509	-874.779977
RF	0.219762	0.155122	0.363699	0.302898	0.252150	0.402678	-51.805929	-37.883373	29.902964
GBR	0.235939	0.190118	0.510648	0.287784	0.256340	0.443468	-9.097675	4.824881	8.548837
FCNN	0.386488	0.343865	1.145509	0.404068	0.351696	0.901899	-9379.246983	-8823.439896	4784.373492
CNN	0.427059	0.365086	1.657634	0.466559	0.419364	1.292262	-9350.237346	-8794.113554	4769.868673
LSTM	0.346322	0.260109	0.715427	0.420559	0.337458	0.713535	-2556.028255	17095.637430	4626.514128
GARCH-FCNN	0.995147	0.933771	4.033025	1.039333	0.980778	3.264084	-9244.498105	-8357.966670	4773.249052
HAR-FCNN	0.393628	0.357526	0.954573	0.390458	0.350685	0.748320	-9289.162111	-8447.094061	4788.081055
GARCH-KNN	0.756573	0.727158	2.431871	0.689509	0.661043	1.781959	3510.565248	3524.487804	-1751.282624
GARCH-RF	0.270804	0.200978	0.451119	0.353499	0.296857	0.490805	173.586433	187.508989	-82.793217
GARCH-GBR	0.473974	0.449592	1.732821	0.424891	0.405166	1.275369	2014.501531	2028.424086	-1003.250765
HAR-KNN	0.439215	0.405713	1.058411	0.439589	0.395986	0.849600	778.808339	792.730895	-385.404170
HAR-RF	0.453031	0.409443	1.065134	0.449830	0.402447	0.853242	547.615296	561.537851	-269.807648
HAR-GBR	0.455889	0.413379	1.081972	0.451096	0.404773	0.862082	586.712510	600.635066	-289.356255
Baseline	0.402458	0.382930	1.327082	0.399249	0.366783	1.010118	941.188909	947.058739	-469.594455

other models yielded a $p_{<0.05}$. This might explain why hybrid models that integrate HAR models with other models exhibit performance comparable to the standalone HAR models. In contrast, the introduction of new terms into the GARCH model didn't markedly enhance the information it captured. All GARCH models displayed five or more instances of $p_{<0.05}$. Among the Machine Learning models, KNN stood out with the least number of $p_{<0.05}$ occurrences—only one—while other Machine Learning models showed three such instances. The GARCH model contributes to one $p_{<0.05}$ in the RF model, specifically 0.02, suggesting that the GARCH model accounts for some variability in the RF's residuals. This might explain why the GARCH-RF model outperformss other GARCH hybrid models. Both GARCH-KNN and HAR-KNN demonstrate $p_{<0.05}$ against the combined traditional model, hinting that the KNN might not be fully harnessing the predictive power of both GARCH and ARCH. Except for GARCH-KNN and GARCH-RF, all Neural Network and hybrid models show more than three $p_{<0.05}$ occurrences. Thus, Machine Learning models seem to possess the strongest capability to extract information across all model categories.

4.3 Sensitivity Analysis

Table 6 illustrates how the models respond to changes in the forecast horizon. Machine learning models tend to yield robust out-of-sample forecasts across different horizons, whereas the error metrics of traditional models fluctuate significantly and reach a peak at the 240-day horizon. The HAR models surpass the Neural Networks at horizons of 120 days and 360 days. However, this ranking is inverted at the 240-day horizon. The forecast performance of neural networks declines as the horizon extends. Among all the hybrid models, only GARCH-RF consistently produces robust outcomes.

Comparing the deviation in performance metrics due to changes in the volatility time frame, Table 7 indicates that most models perform better as the time frame increases. Conversely, the performance of HAR, hybrid HAR, and SVR diminishes with an extended time frame. The results from CNN and LSTM display considerable variation, suggesting a lack of robustness to changes in the time frame.

5 Conclusions

This research conducted a comprehensive comparison of the long-term volatility forecasting performance of GARCH models, HAR models, Machine Learning models, Neural Networks, and hybrid models. The GARCH models exhibited the poorest out-of-sample performance, potentially due to their intrinsic computation of conditional variance, while this study adopted historical volatility as the feature. On the other hand, HAR models surpassed most other models, including all GARCH models, SCR, and neural networks.

Table 3: Efficiency Test (time frame = 30day, horizon = 240day).

	Average Intercept	Average Slope	Average Intercept p-value	Average Slope p-value
GARCH	0.148932	0.236039	0.192581	0.073063
GJR-GARCH	0.123070	0.275378	0.126685	0.117143
FIGARCH	0.206944	0.237909	0.114752	0.063924
HAR	0.430389	0.045895	0.000005	0.566254
HAR-J	0.431817	0.047768	0.000009	0.574175
HAR-Q	0.436494	0.043826	0.000014	0.601442
KNN	0.499758	-0.143637	0.184351	0.182784
SVR	0.145108	0.307907	0.141885	0.090752
RF	-0.324256	1.102051	0.301199	0.280562
GBR	0.933946	-2.331422	0.466122	0.297359
FCNN	0.649140	-0.191505	0.235010	0.113288
CNN	-0.181235	1.580594	0.006397	0.100468
LSTM	-0.211214	1.555045	0.015292	0.015636
GARCH-FCNN	0.246717	0.164312	0.269831	0.086629
HAR-FCNN	0.397109	0.115620	0.093932	0.499785
GARCH-KNN	0.277599	0.148415	0.187728	0.182785
GARCH-RF	10.763607	-19.432765	0.236197	0.268837
GARCH-GBR	0.245972	0.201411	0.314428	0.115218
HAR-KNN	0.454604	0.105629	0.121003	0.472244
HAR-RF	0.415240	0.083624	0.002104	0.440760
HAR-GBR	0.433527	0.074851	0.027581	0.532069

Table 4: Orthogonality Test 1 (time frame = 30day, horizon = 240day).

	GARCH	GJR-GARCH	FIGARCH	HAR	HAR-J	HAR-Q	KNN	SVR	RF
GJR-GARCH	0.073	NaN	0.039	0.157	0.170	0.181	0.095	0.082	0.027
FIGARCH	0.104	0.120	NaN	0.180	0.166	0.188	0.179	0.157	0.074
HAR	0.323	0.404	0.328	NaN	0.000	0.000	0.548	0.410	0.589
HAR-J	0.322	0.383	0.332	0.000	NaN	0.000	0.554	0.394	0.594
HAR-Q	0.310	0.361	0.321	0.000	0.000	NaN	0.580	0.374	0.625
KNN	0.199	0.151	0.189	0.221	0.216	0.223	NaN	0.112	0.181
SVR	0.082	0.117	0.032	0.153	0.141	0.143	0.093	NaN	0.165
RF	0.020	0.032	0.063	0.277	0.276	0.285	0.425	0.003	NaN
GBR	0.060	0.063	0.210	0.190	0.195	0.190	0.309	0.052	0.236
FCNN	0.071	0.191	0.072	0.284	0.269	0.286	0.079	0.057	0.034
CNN	0.180	0.180	0.099	0.240	0.236	0.225	0.180	0.108	0.113
LSTM	0.105	0.117	0.004	0.168	0.172	0.168	0.114	0.184	0.001
GARCH-FCNN	0.020	0.023	0.024	0.180	0.203	0.205	0.149	0.017	0.155
HAR-FCNN	0.462	0.495	0.488	0.097	0.094	0.085	0.510	0.515	0.489
GARCH-KNN	0.001	0.026	0.021	0.109	0.120	0.115	0.376	0.000	0.129
GARCH-RF	0.093	0.173	0.071	0.321	0.329	0.314	0.272	0.007	0.212
GARCH-GBR	0.012	0.012	0.017	0.099	0.109	0.103	0.213	0.008	0.098
HAR-KNN	0.041	0.052	0.100	0.008	0.008	0.010	0.416	0.145	0.484
HAR-RF	0.432	0.471	0.497	0.031	0.025	0.022	0.481	0.530	0.448
HAR-GBR	0.443	0.480	0.499	0.074	0.065	0.060	0.465	0.453	0.529
GARCH	NaN	0.083	0.054	0.092	0.092	0.093	0.048	0.081	0.002

Table 5: Orthogonality Test 2 (time frame = 30day, horizon = 240day).

	GBR	FCNN	CNN	LSTM	GARCH-FCNN	HAR-FCNN	GARCH-KNN	GARCH-RF	GARCH-GBR	HAR-KNN	HAR-RF	HAR-GBR
GJR-GARCH	0.330	0.101	0.010	0.000	0.000	0.302	0.058	0.117	0.134	0.224	0.231	0.282
FIGARCH	0.000	0.110	0.003	0.000	0.075	0.125	0.001	0.000	0.007	0.004	0.124	0.130
HAR	0.495	0.548	0.528	0.515	0.249	0.212	0.203	0.527	0.296	0.284	0.021	0.119
HAR-J	0.503	0.560	0.539	0.525	0.225	0.207	0.206	0.535	0.310	0.288	0.026	0.119
HAR-Q	0.526	0.550	0.542	0.494	0.209	0.226	0.219	0.564	0.314	0.265	0.029	0.107
KNN	0.231	0.110	0.186	0.181	0.148	0.264	0.011	0.231	0.114	0.144	0.231	0.160
SVR	0.285	0.040	0.113	0.000	0.000	0.169	0.045	0.091	0.184	0.024	0.049	0.023
RF	0.212	0.299	0.314	0.004	0.001	0.254	0.040	0.283	0.008	0.213	0.212	0.228
GBR	NaN	0.277	0.194	0.123	0.181	0.236	0.086	0.315	0.141	0.322	0.143	0.113
FCNN	0.154	NaN	0.125	0.000	0.007	0.250	0.119	0.074	0.146	0.189	0.171	0.213
CNN	0.221	0.201	NaN	0.001	0.009	0.135	0.022	0.085	0.129	0.143	0.079	0.060
LSTM	0.023	0.268	0.003	NaN	0.078	0.177	0.000	0.016	0.111	0.045	0.196	0.165
GARCH-FCNN	0.167	0.069	0.033	0.000	NaN	0.332	0.117	0.087	0.154	0.294	0.148	0.122
HAR-FCNN	0.526	0.667	0.463	0.367	0.360	NaN	0.580	0.441	0.589	0.095	0.022	0.104
GARCH-KNN	0.228	0.324	0.107	0.000	0.000	0.313	NaN	0.195	0.132	0.299	0.241	0.232
GARCH-RF	0.233	0.218	0.349	0.014	0.002	0.118	0.035	NaN	0.127	0.082	0.198	0.267
GARCH-GBR	0.244	0.123	0.065	0.120	0.000	0.398	0.116	0.119	NaN	0.225	0.189	0.147
HAR-KNN	0.257	0.430	0.441	0.207	0.013	0.034	0.052	0.327	0.102	NaN	0.007	0.001
HAR-RF	0.453	0.586	0.546	0.432	0.465	0.031	0.476	0.465	0.506	0.080	NaN	0.027
HAR-GBR	0.544	0.582	0.585	0.443	0.433	0.021	0.435	0.522	0.488	0.034	0.000	NaN
GARCH	0.243	0.069	0.004	0.000	0.000	0.218	0.096	0.073	0.067	0.149	0.199	0.226

Hybrid models built on the foundation of HAR generally showed better performance than those based on GARCH models. However, all HAR models displayed bias and lacked robustness, with a notably diminished performance at the 240-day horizon.

The findings regarding neural networks differ from previous studies by Lu et al. (2022) and Christensen et al., 2021. This research challenges the notion that neural networks can consistently sidestep over-fitting issues and maintain strong robustness. While they demonstrated impressive in-sample performance, their forecasting efficacy was outperformed by simpler machine learning models and HAR models. Notably, the results from CNN and LSTM models were susceptible to fluctuations based on the time frame. These observations align with Elsayed et al. (2021), arguing against the superiority of deep learning methods in time series forecasting. Compared to research supporting deep learning methods (Christensen et al., 2021; McAleer and Medeiros, 2011; Rahimikia and Poon, 2020; Shen et al., 2021), it's worth noting that their methodologies encompass the integration of additional exogenous variables, utilization of large datasets, and refined feature engineering. Such conditions aren't always attainable. For instance, the relatively recent emergence of cryptocurrencies limits available data, thus hampering the training efficacy of intricate deep learning models. Furthermore, exogenous variables can induce instability in forecasts across different assets.

Regarding hybrid models, the conclusions drawn in this research diverge from studies like Kristjanpoller et al. (2014), Kim and Won (2018), Hajizadeh et al. (2012), and Seo and Kim (2020). While many hybrid models enhance forecast accuracy relative to one of their foundational models, they might not surpass the performance of the other contributing model. This research posits that standalone machine learning models and FCNN can outperform hybrid variants that incorporate traditional models.

The empirical analysis conducted in this research strongly underpinned KNN and RF models in volatility forecasting. These models delivered exemplary out-of-sample and high-volatility period performance, exhibiting resilience against variations in forecast horizon and volatility time frame. Models like RF and KNN, despite being simple non-parametric classification models, consistently outperformed linear regression, autoregressive, and complex neural network models in volatility forecasting.

While this research provides a detailed comparative analysis, it's crucial to acknowledge potential limitations. The findings might be sensitive to specific asset types or market conditions. Future research could explore the impact of diverse economic climates, such as recessions and financial crisis, on these models. There's also room to investigate other features engineering techniques and different proxies for latent volatility that might refine the models' forecasting capabilities further. Additionally, it would be beneficial to explore the generalizability of these results across different financial markets or asset types to

ensure broader applicability and it is not neglectable that deep learning models have exceptional potential in data-rich environments.

References

Andersen, T. G., Bollerslev, T., & Diebold, F. X. (2007). Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility. *The review of economics and statistics*, *89*(4), 701–720.

Baillie, R. T., Bollerslev, T., & Mikkelsen, H. O. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *74*(1), 3–30.

Bali, T. G. (2003). An extreme value approach to estimating volatility and value at risk. *The Journal of Business*, *76*(1), 83–108.

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

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

Bollerslev, T., Patton, A. J., & Quaedvlieg, R. (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics*, *192*(1), 1–18.

Brooks, C. (2019). *Introductory econometrics for finance*. Cambridge university press.

Chou, R. Y. (2005). Forecasting financial volatilities with extreme values: The conditional autoregressive range (carr) model. *Journal of Money, Credit and Banking*, 561–582.

Christensen, K., Siggaard, M., & Veliyev, B. (2021). A machine learning approach to volatility forecasting. *Available at SSRN*.

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

Corsi, F., & Renò, R. (2012). Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling. *Journal of Business & Economic Statistics*, *30*(3), 368–380.

Davidian, M., & Carroll, R. J. (1987). Variance function estimation. *Journal of the American statistical association*, *82*(400), 1079–1091.

Ding, D., Zhang, M., Pan, X., Yang, M., & He, X. (2019). Modeling extreme events in time series prediction. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1114–1122.

Elsayed, S., Thyssens, D., Rashed, A., Jomaa, H. S., & Schmidt-Thieme, L. (2021). Do we really need deep learning models for time series forecasting? *arXiv preprint arXiv:2101.02118*.

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

Engle, R. F., & Patton, A. J. (2001). What good is a volatility model? *Quantitative finance*, *1*(2), 237.

Fair, R. C., & Shiller, R. J. (1990). Comparing information in forecasts from econometric models. *The American Economic Review*, 375–389.

Figlewski, S. (1997). Forecasting volatility. *Financial markets, institutions & instruments*, *6*(1), 1–88.

Filipović, D., & Khalilzadeh, A. (2021). Machine learning for predicting stock return volatility. *Swiss Finance Institute Research Paper*, (21-95).

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

Gray, S. F. (1996). Modeling the conditional distribution of interest rates as a regime-switching process. *Journal of Financial Economics*, *42*(1), 27–62.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Hajizadeh, E., Seifi, A., Zarandi, M. F., & Turksen, I. (2012). A hybrid modeling approach for forecasting the volatility of s&p 500 index return. *Expert Systems with Applications*, *39*(1), 431–436.

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

Hansen, L. P., & Hodrick, R. J. (1980). Forward exchange rates as optimal predictors of future spot rates: An econometric analysis. *Journal of political economy*, *88*(5), 829–853.

Hull, J., & White, A. (1987). The pricing of options on assets with stochastic volatilities. *The journal of finance*, *42*(2), 281–300.

Ketkar, N., & Santana, E. (2017). *Deep learning with python* (Vol. 1). Springer.

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

Kristjanpoller, W., Fadic, A., & Minutolo, M. C. (2014). Volatility forecast using hybrid neural network models. *Expert Systems with Applications*, *41*(5), 2437–2442.

Lu, X., Ma, F., Xu, J., & Zhang, Z. (2022). Oil futures volatility predictability: New evidence based on machine learning models. *International Review of Financial Analysis*, *83*, 102299.

McAleer, M., & Medeiros, M. C. (2011). Forecasting realized volatility with linear and nonlinear univariate models. *Journal of Economic Surveys*, *25*(1), 6–18.

Morgan, J. P., et al. (1996). Riskmetrics technical document.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the econometric society*, 347–370.

Patton, A. J., & Sheppard, K. (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and Statistics*, *97*(3), 683–697.

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

Rahimikia, E., & Poon, S.-H. (2020). Machine learning for realised volatility forecasting. *Available at SSRN*, *3707796*.

Seo, M., & Kim, G. (2020). Hybrid forecasting models based on the neural networks for the volatility of bitcoin. *Applied Sciences*, *10*(14), 4768.

Shen, Z., Wan, Q., & Leatham, D. J. (2021). Bitcoin return volatility forecasting: A comparative study between garch and rnn. *Journal of Risk and Financial Management*, *14*(7), 337.

Ughi, R., Lomurno, E., & Matteucci, M. (2023). Two steps forward and one behind: Rethinking time series forecasting with deep learning. *arXiv preprint arXiv:2304.04553*.

Table 6: Sensitivity Analysis with Horizon

	120day		240day		360day	
	RMSE	Extreme_RMSE	RMSE	Extreme_RMSE	RMSE	Extreme_RMSE
GARCH	0.6287	0.4972	0.7637	0.6950	0.8190	0.7186
GJR-GARCH	0.6164	0.4788	0.9988	0.9251	0.7269	0.6419
FIGARCH	0.5008	0.4391	0.6839	0.6134	0.7762	0.6807
HAR	0.2883	0.3614	0.4624	0.4576	0.2730	0.2993
HAR-J	0.2889	0.3624	0.4636	0.4597	0.2744	0.3006
HAR-Q	0.2887	0.3632	0.4615	0.4603	0.2736	0.3011
KNN	0.2637	0.3651	0.2274	0.2841	0.2591	0.2913
SVR	0.5999	0.5500	0.5902	0.5542	0.6024	0.5352
RF	0.2458	0.3166	0.2198	0.3029	0.2496	0.2864
GBR	0.2474	0.3117	0.2360	0.2878	0.2571	0.2930
FCNN	0.3027	0.4264	0.3865	0.4041	0.4854	0.5021
CNN	0.3216	0.4443	0.4271	0.4666	0.5434	0.5351
LSTM	0.3493	0.3060	0.3463	0.4206	0.5917	0.6138
GARCH-FCNN	0.6595	0.5208	0.9951	1.0393	0.7521	0.6839
HAR-FCNN	0.3143	0.3845	0.3936	0.3905	0.2695	0.2979
GARCH-KNN	0.6845	0.5716	0.7566	0.6895	0.7820	0.7116
GARCH-RF	0.2526	0.3094	0.2708	0.3535	0.2740	0.3118
GARCH-GBR	0.6659	0.6364	0.4740	0.4249	0.4550	0.4005
HAR-KNN	0.2747	0.3454	0.4392	0.4396	0.2672	0.2956
HAR-RF	0.3038	0.3747	0.4530	0.4498	0.2707	0.2970
HAR-GBR	0.2883	0.3502	0.4559	0.4511	0.2734	0.2968
Baseline	0.4272	0.4549	0.4025	0.3992	0.3810	0.3545

Table 7: Sensitivity Analysis with Time Frame

	7day		14day		30day	
	RMSE	Extreme_RMSE	RMSE	Extreme_RMSE	RMSE	Extreme_RMSE
GARCH	0.8384	0.6855	0.8031	0.6599	0.7637	0.6950
GJR-GARCH	1.0721	0.9310	1.0377	0.9036	0.9988	0.9251
FIGARCH	0.7618	0.6874	0.7256	0.6334	0.6839	0.6134
HAR	0.3460	0.4423	0.4217	0.4731	0.4624	0.4576
HAR-J	0.3624	0.4632	0.4287	0.4821	0.4636	0.4597
HAR-Q	0.3376	0.4702	0.4262	0.4854	0.4615	0.4603
KNN	0.3292	0.5002	0.3155	0.4758	0.2274	0.2841
SVR	0.4626	0.5465	0.4973	0.5409	0.5902	0.5542
RF	0.3066	0.4706	0.2451	0.3958	0.2198	0.3029
GBR	0.3261	0.4884	0.2525	0.3919	0.2360	0.2878
FCNN	0.3611	0.4926	0.4809	0.5529	0.3865	0.4041
CNN	0.3505	0.5011	0.9808	1.3042	0.4271	0.4666
LSTM	0.4823	0.4639	0.8513	0.8034	0.3463	0.4206
GARCH-FCNN	0.8973	0.7363	0.8477	0.7183	0.9951	1.0393
HAR-FCNN	0.3356	0.4496	0.4947	0.5632	0.3936	0.3905
GARCH-KNN	0.7943	0.6427	0.9077	0.7561	0.7566	0.6895
GARCH-RF	0.7472	0.6267	0.3489	0.3380	0.2708	0.3535
GARCH-GBR	0.8027	0.6665	0.6701	0.5874	0.4740	0.4249
HAR-KNN	0.3229	0.4512	0.3884	0.4420	0.4392	0.4396
HAR-RF	0.3548	0.4544	0.2516	0.3525	0.4530	0.4498
HAR-GBR	0.3377	0.4488	0.3800	0.4455	0.4559	0.4511
Baseline	0.4327	0.5203	0.4204	0.4749	0.4025	0.3992