NOVA IMS
INFORMATION MANAGEMENT SCHOOL

Master's Degree in
Data Science and Advanced Analytics

Specialization in
Data Science

Comparing Econometric and Machine Learning Approaches to Volatility Forecasting

Evidence from Cryptocurrency and Traditional Markets

Davide Montali

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade NOVA de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

COMPARING ECONOMETRIC AND MACHINE LEARNING APPROACHES TO VOLATILITY FORECASTING

Evidence from Cryptocurrency and Traditional Markets

by

Davide Montali

Master Thesis presented as partial requirement for obtaining the Master Degree in Data Science and Advanced Analytics with a specialization in Data Science

Supervised by

Professor Fernando Bacao, NOVA IMS

April, 2026

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism, any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Tokyo, Japan. 20th February 2026

Davide Montali

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof Bacao for his guidance and support throughout this thesis. His expertise and feedback were instrumental in shaping this thesis.

ABSTRACT

Can machine learning methods outperform traditional econometric models for volatility forecasting? The answer, this thesis finds, depends on the specific asset under consideration. This thesis compares GARCH, EGARCH, Random Forest, XGBoost, and LSTM models using daily data from 2019 to 2025 for Bitcoin, Ethereum, the S&P 500, and VIX. Performance is measured out-of-sample via RMSE, with Diebold-Mariano tests employed to assess whether observed differences are statistically significant. The results resist a simple "ML wins" or "stick with GARCH" conclusion. For the S&P 500, no machine learning method significantly improves on GARCH, and LSTM performs worse than GARCH outright. The only statistically significant improvement for equities comes from EGARCH ($p<0.01$), which captures the leverage effect that symmetric GARCH misses. Model structure, not model complexity, is what matters for equity volatility. For Bitcoin, the tree-based methods significantly outperform GARCH, with XGBoost achieving the lowest RMSE and an improvement of approximately 11 per cent relative to the GARCH baseline. But for Ethereum, machine learning offers no significant improvement; EGARCH performs best. Same asset class. Different outcomes. From a practical standpoint, the 11% RMSE reduction for Bitcoin could meaningfully affect risk management and position sizing decisions. For equity markets, EGARCH represents the appropriate upgrade from basic GARCH. For Ethereum, the additional model complexity appears to offer no benefit. The findings suggest that model selection needs to be asset-specific and validated through statistical testing rather than RMSE rankings alone. Deep learning neither "always works" nor "never works" for volatility. It depends on the asset.

KEYWORDS

Volatility Forecasting; GARCH; Machine Learning; LSTM; Cryptocurrency; Risk Management

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and Motivation

Financial markets do not sit still. The COVID-19 pandemic made this abundantly clear: volatility spiked across equities and cryptocurrencies alike, catching many risk models off guard. For practitioners managing portfolios or pricing options, these swings were not abstract. They carried real costs.

Predicting volatility has long been central to quantitative finance. Accurate forecasts enable effective hedging, fair derivative pricing, and sensible position sizing. Poor forecasts, by contrast, can lead to substantial losses. Since Engle (1982) introduced the ARCH model and Bollerslev (1986) extended it to GARCH, these econometric frameworks have been the go-to approach for modelling how volatility evolves over time. They remain popular for good reason: the parameters are interpretable, estimation is straightforward, and decades of research have refined their use.

Two developments over the past decade have begun to complicate this picture. First, cryptocurrencies arrived. Bitcoin and Ethereum now command market capitalisations in the hundreds of billions, yet they appear to behave quite differently from traditional equities. They trade around the clock, experience substantial price swings, and seem prone to sudden regime shifts that may leave standard models struggling. Second, machine learning has matured. Techniques such as LSTMs, Random Forests, and gradient boosting have demonstrated strong performance in other forecasting domains, prompting a natural question: might they perform better for volatility as well?

1.2. Research Problem

There appears to be a genuine trade-off between models that are easy to interpret and models that may be more accurate. GARCH parameters have clear economic meaning: they capture the persistence of shocks, the response to news, and asymmetric effects. Regulators often prefer this transparency. Traders, however, tend to focus on whether forecasts are correct. If a neural network produces better predictions, the opacity represents an acceptable cost.

Machine learning approaches can, in principle, capture nonlinear patterns without requiring the analyst to specify them in advance. This flexibility is appealing. It also comes with potential costs: these models can be harder to interpret, can require more data, and can overfit in ways that only become apparent out of sample. The question this thesis attempts to address is straightforward: **when, if ever, does the added complexity of machine learning pay off for volatility forecasting?**

1.3. Research Objectives

To address this question, this thesis compares forecasting performance across four assets (Bitcoin, Ethereum, the S&P 500, and VIX) using a consistent methodology throughout. This cross-asset design is important. Much of the existing literature examines either crypto or traditional markets, but rarely both together. By holding the methods fixed and varying the assets, it becomes possible to assess whether any machine learning advantage is confined to specific market conditions.

The analysis also examines a range of model complexity, from GARCH through Random Forest to LSTM. The goal is not to declare a single winner but to understand where on the interpretability-flexibility spectrum different approaches tend to excel. Rather than simply ranking models by RMSE, Diebold-Mariano tests are employed to assess whether observed differences are statistically meaningful or simply reflect sampling variation.

1.4. Research Questions

With that framing, this thesis addresses three questions:

- **RQ1**: Do machine learning methods actually outperform GARCH-family models for cryptocurrency volatility, or is the apparent advantage overstated?
- **RQ2**: If there is an ML advantage in crypto markets, does it carry over to traditional equities, or does it disappear?
- **RQ3**: Among the machine learning approaches, do the more complex models (like LSTM) genuinely beat simpler ones (like Random Forest), or is simpler sometimes better?

1.5. Contributions

A few things distinguish this work from existing studies. Most papers focus on either cryptocurrencies or equities; this thesis examines both under identical conditions, which permits direct comparison. The range of models (from interpretable GARCH to opaque LSTM) should help clarify when complexity adds value and when it merely adds computation time. The statistical tests move beyond raw accuracy metrics to assess whether differences are likely to be genuine.

The main finding is that model performance varies substantially across assets. Machine learning methods collectively outperform GARCH for Bitcoin but not for Ethereum, while for the S&P 500, LSTM is the worst-performing model and only EGARCH achieves a significant improvement. Blanket recommendations are probably too simple; asset-specific evaluation is necessary.

1.6. Thesis Structure

Chapter 2 reviews the relevant literature. Chapter 3 describes the data, model specifications, and evaluation procedure. Chapter 4 presents results and discusses their implications. Chapter 5 concludes.

2. LITERATURE REVIEW

The literature on volatility forecasting spans four decades of econometric refinement and a more recent wave of machine learning applications. This chapter covers both traditions, ending with the gaps this thesis attempts to fill.

2.1. Volatility in Financial Markets

At its simplest, volatility is the standard deviation of returns. This apparent simplicity, however, can be misleading. Volatility sits at the heart of option pricing, risk management, and portfolio construction (Hull, 2018). Underestimating it leads to excessive risk exposure; overestimating it results in suboptimal capital allocation.

What makes volatility tricky to forecast is that it does not behave like a well-mannered statistical object. Returns cluster: calm periods follow calm periods, and turbulent stretches tend to persist (Poon & Granger, 2003). The distribution of returns is fat-tailed, meaning extreme moves happen more often than a normal distribution would suggest. And there is asymmetry: bad news tends to rattle markets more than good news of equivalent magnitude. These three stylised facts (clustering, fat tails, asymmetry) are well established. Any model worth using needs to accommodate them, at least to some degree.

2.2. Econometric Approaches: The GARCH Family

2.2.1. ARCH and GARCH Models

The modern treatment of volatility can be traced to Engle (1982), who proposed the Autoregressive Conditional Heteroskedasticity (ARCH) model. The core idea appears straightforward in retrospect: if volatility clusters, then current variance should depend on past shocks. Formally:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 \tag{2.1}$$

Here σ_t^2 is the conditional variance at time t, ω is a baseline level, and the α_i terms pick up the influence of past squared residuals.

Bollerslev (1986) extended this by letting variance depend on its own lagged values, yielding the Generalised ARCH (or GARCH) model:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{2.2}$$

The GARCH(1,1) version, with just one lag of each type, has proven surprisingly durable. It remains the workhorse model in both academia and industry (Engle, 2001). The sum $\alpha_1 + \beta_1$ measures persistence: values near one mean that volatility shocks take a long time to die out.

2.2.2. Extensions: EGARCH and Asymmetric Effects

Standard GARCH treats positive and negative shocks symmetrically. In practice, that is often wrong. A 3% drop tends to unsettle markets more than a 3% gain, the so-called leverage effect. Nelson (1991) addressed this with the Exponential GARCH (EGARCH) specification:

$$\ln(\sigma_t^2) = \omega + \sum_{i=1}^{q} \left[\alpha_i \left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right| + \gamma_i \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right] + \sum_{j=1}^{p} \beta_j \ln(\sigma_{t-j}^2) \tag{2.3}$$

The γ parameter captures asymmetry. When γ is negative, negative returns tend to raise volatility more than positive returns of equivalent magnitude, consistent with patterns typically observed in equity markets.

2.2.3. Strengths and Limitations

GARCH models have endured for sound reasons. The parameters carry economic interpretation: shock persistence, news impact, and asymmetric responses can all be discussed in terms that make economic sense. Estimation is relatively efficient, and forecasts can be computed analytically (Bollerslev et al., 1992). For risk managers who must explain their models to regulators, this transparency is valuable.

That said, GARCH is not without problems. The models impose a specific functional form on volatility dynamics, which can miss subtleties in the data. They are essentially linear in squared returns, so genuinely nonlinear patterns can slip through. And when markets undergo structural breaks (regulatory changes, financial crises, pandemics), GARCH models can be slow to adapt.

2.3. Machine Learning Approaches

2.3.1. Random Forests

Random Forests, introduced by Breiman (2001), take a different approach. Rather than specifying a model in advance, they construct multiple decision trees from the data and average their predictions. For volatility forecasting, this typically involves training trees on lagged returns, past volatility, and other potentially relevant features.

The appeal lies in their flexibility. Random Forests can capture nonlinear relationships without requiring the analyst to specify the functional form. They tend to handle outliers reasonably well

and provide built-in measures of feature importance. On the downside, the forecasts tend to be smoother than realised volatility; extreme spikes tend to be attenuated through averaging.

2.3.2. Gradient Boosting: XGBoost

XGBoost (Chen & Guestrin, 2016) builds on the ensemble idea but takes a sequential approach: each new tree focuses on correcting the mistakes of the previous ones. Regularisation helps prevent overfitting, and the algorithm handles missing data gracefully.

In volatility applications, XGBoost often does well. Some studies report R-squared values above 0.95, though such numbers should be treated with caution since they depend heavily on how the problem is set up. The model is sensitive to hyperparameter choices, and getting good performance typically requires some tuning.

2.3.3. Long Short-Term Memory Networks

LSTMs (Hochreiter & Schmidhuber, 1997) are a type of recurrent neural network designed to capture dependencies over long sequences. Their gating mechanisms let the network decide what information to remember and what to forget, which is useful for time series where patterns can persist over many periods.

For volatility forecasting, LSTMs process sequences of past returns or realised volatility to generate predictions. They do not require explicit specification of lag structures; the network learns these from the data. Fischer and Krauss (2018) found that LSTMs could outperform other machine learning methods in financial applications, and Liu (2019) reported similar results for equity index volatility, particularly during periods of market stress.

Whether LSTMs offer genuine improvements or have simply benefited from recent popularity remains an open question. They require more data and computation than simpler methods, and the reported improvements have not been consistent across assets or time periods.

2.3.4. Hybrid Approaches

Some researchers have tried combining the best of both worlds. Kim and Won (2018), for instance, proposed feeding GARCH residuals into an LSTM, letting the econometric model handle the well-understood linear dynamics while the neural network picks up whatever remains.

The logic is sensible: GARCH captures the basics, LSTM learns the rest. In practice, hybrid models sometimes outperform their components, but not always. The added complexity is only worthwhile if the performance gains are real and persistent. This study deliberately focuses on comparing pure econometric and pure machine learning approaches. Establishing how well each approach performs on its own is a necessary step before adding hybrid complexity on top.

2.4. Cryptocurrency Volatility

2.4.1. Characteristics of Crypto Markets

Cryptocurrencies present distinct modelling challenges. Bitcoin and Ethereum exhibit volatility levels several times higher than traditional equities, with daily moves of 5% or more occurring with some regularity (Baur et al., 2018). They trade continuously (no market close, no overnight gap), and the investor base tends to skew younger and more speculative.

Structural breaks are frequent. A tweet from a prominent figure, a regulatory crackdown in a major market, or a technical failure can send prices lurching. Dyhrberg (2016) found some parallels between Bitcoin and gold as hedging instruments, but the volatility profile is distinctly higher.

2.4.2. GARCH Models for Cryptocurrency

Researchers have applied GARCH-family models to crypto with mixed success. Katsiampa (2017) compared various specifications for Bitcoin and found that component GARCH models (which separate short-run and long-run volatility) fit best. This suggests that crypto volatility has both transient and persistent elements that need separate treatment.

Interestingly, the asymmetry in crypto markets sometimes runs opposite to equities. Rather than negative returns boosting volatility, some studies find that positive returns do, perhaps reflecting speculative exuberance. The leverage effect, in other words, might not hold in its usual form.

2.4.3. Machine Learning for Crypto Volatility

Machine learning has attracted considerable attention in this space. Aras (2021) compared deep learning approaches with GARCH for Bitcoin and reported that neural networks produced lower forecast errors, especially at longer horizons.

Zahid et al. (2022) experimented with hybrid GARCH-ML models and found that combining econometric structure with machine learning flexibility improved accuracy. But they also noted diminishing returns: fancier architectures did not necessarily beat simpler ones. This is a recurring theme. Complexity does not guarantee improvement.

2.5. Comparative Studies and Research Gap

2.5.1. GARCH versus Machine Learning

The literature comparing GARCH and machine learning presents a somewhat inconsistent picture. Bucci (2020) found that neural networks outperformed GARCH for realised volatility

on equity indices, particularly during turbulent periods. However, Christensen et al. (2023) demonstrated that simpler methods (including ridge regression) could match or exceed deep learning performance in certain settings.

Rahimikia and Poon (2021) conducted a broader comparison and concluded that gradient boosting often performed well, though the best-performing model varied by asset and horizon. The emerging consensus appears to be that no universal best model exists. Context matters: GARCH performs adequately in calm markets, while machine learning approaches might adapt better during regime transitions.

2.5.2. Identified Research Gap

For all the papers published, some gaps remain. First, most studies focus on either crypto or traditional assets, rarely both together. This makes it hard to know whether findings transfer across asset classes. Second, comparisons often pit GARCH against a single ML method (usually LSTM), leaving open where simpler alternatives like Random Forest might fit. Third, many papers emphasise in-sample fit rather than genuine out-of-sample forecasting, and when out-of-sample tests are used, the holdout periods are often short.

This thesis attempts to address these gaps by applying a consistent methodology (GARCH, EGARCH, Random Forest, XGBoost, and LSTM) to both cryptocurrencies (Bitcoin, Ethereum) and traditional assets (S&P 500, VIX). The focus is on out-of-sample performance, evaluated with standard metrics and Diebold-Mariano tests (Diebold & Mariano, 1995) to assess whether observed differences are statistically meaningful.

3. METHODOLOGY

This chapter describes the methodology: data sources, model specifications, and evaluation procedures. The aim is to provide enough detail that the analysis could be replicated.

3.1. Data

3.1.1. Data Sources

The analysis uses daily price data for four assets, selected to span both cryptocurrency and traditional markets:

- **Bitcoin (BTC)**: The original and still the largest cryptocurrency by market cap. It is the obvious starting point for any crypto analysis.
- **Ethereum (ETH)**: The second-largest cryptocurrency, but with a fundamentally different profile: its role as the primary platform for decentralised finance, its distinct protocol upgrade cycle, and its smart contract ecosystem all suggest potentially different volatility dynamics. Its inclusion permits assessment of whether Bitcoin results generalise within the crypto space.
- **S&P 500 Index (SPX)**: The standard U.S. equity benchmark. It provides a well-behaved comparison case.
- **CBOE Volatility Index (VIX)**: Sometimes called the "fear index." It measures implied volatility on S&P 500 options and behaves quite differently from the underlying index.

Cryptocurrency data comes from Polygon.io, which offers institutional-quality market data. Traditional asset data is from Yahoo Finance: convenient, widely used, and free. The sample runs from January 2019 through December 2025, giving roughly seven years of daily observations. For cryptocurrencies, that means data every day; for equities and VIX, trading days only.

3.1.2. Data Preprocessing

Log returns are computed in the standard manner:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{3.1}$$

where P_t is the closing price on day t. Log returns are convenient because they add across time and tend to be closer to normally distributed than simple returns, though "closer" is doing a lot of work here. These series still have fat tails.

Realised volatility over a rolling window of w days is defined as:

$$RV_t^{(w)} = \sqrt{\frac{252}{w} \sum_{i=0}^{w-1} r_{t-i}^2} \tag{3.2}$$

The 252 factor annualises the measure, assuming 252 trading days per year. The target variable is 5-day forward realised volatility (described in Section 3.3). The 21-day, 10-day, and 5-day rolling windows serve as features for the machine learning models.

3.1.3. Descriptive Statistics

Table 3.1 summarises the daily returns for each asset.

Table 3.1.: Descriptive Statistics of Daily Log Returns (2019–2025)

Asset	Obs.	Mean (%)	Std. (%)	Skewness	Kurtosis	Ann. Vol. (%)
BTC	2,556	0.10	3.38	-0.58	9.12	53.7
ETH	2,556	0.11	4.21	-0.28	7.45	66.8
SPX	1,758	0.05	1.15	-0.85	14.38	18.3
VIX	1,758	0.01	6.75	1.28	10.02	107.2

A few things jump out. Cryptocurrencies are volatile (no surprise there), with annualised volatility around three to four times that of the S&P 500. All four series show fat tails (kurtosis well above 3) and skewness. Equity returns are negatively skewed, consistent with the leverage effect; VIX is positively skewed, reflecting its tendency to spike during crises. The VIX itself has extremely high volatility, which is worth keeping in mind when interpreting the results for that asset.

3.2. Model Specifications

3.2.1. GARCH(1,1)

The workhorse GARCH(1,1) model specifies:

$$r_t = \mu + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \quad z_t \sim N(0,1) \tag{3.3}$$

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \tag{3.4}$$

The constraints $\omega > 0$, $\alpha \geq 0$, $\beta \geq 0$, and $\alpha + \beta < 1$ ensure positivity and stationarity. The α parameter captures how yesterday's shock feeds into today's variance; β measures how persistent volatility is on its own. When $\alpha + \beta$ is close to one, shocks decay slowly. Volatility clusters.

The model is estimated via quasi-maximum likelihood assuming normality. This approach yields consistent estimates even when the actual distribution has fatter tails, as is typically the case with financial returns.

3.2.2. EGARCH(1,1)

The Exponential GARCH model (Nelson, 1991) works with the log of variance:

$$\ln(\sigma_t^2) = \omega + \alpha \left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| + \gamma \frac{\epsilon_{t-1}}{\sigma_{t-1}} + \beta \ln(\sigma_{t-1}^2) \tag{3.5}$$

The γ term is the key addition. If γ is negative, bad news raises volatility more than good news of the same size, the leverage effect. This asymmetry is well documented in equity markets, though as noted earlier, it can run the other way in crypto.

3.2.3. Random Forest

Random Forest (Breiman, 2001) builds an ensemble of decision trees and averages their predictions:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x}) \tag{3.6}$$

where B is the number of trees and $T_b(\mathbf{x})$ is the prediction of tree b given features $\mathbf{x}$.

Each tree is trained on a bootstrap sample, and at each split only a random subset of features is considered. This randomisation tends to reduce overfitting. The analysis employs standard hyperparameters: 100 trees, maximum depth of 10, and a minimum of 5 samples per split. No extensive tuning was performed; the objective is to assess how a reasonable off-the-shelf configuration performs.

3.2.4. XGBoost

XGBoost (Chen & Guestrin, 2016) also builds an ensemble of trees, but sequentially: each new tree tries to correct the errors of the ones before it. The objective function includes a regularisation term:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.7}$$

where l is the loss and $\Omega(f_k)$ penalises complexity. This helps prevent overfitting, which matters because boosting can otherwise memorise training data.

Configuration: 100 estimators, max depth 6, learning rate 0.1. Again, nothing exotic.

3.2.5. LSTM Neural Network

LSTMs (Hochreiter & Schmidhuber, 1997) are recurrent networks with gating mechanisms that control information flow:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(forget gate)} \tag{3.8}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(input gate)} \tag{3.9}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(candidate state)} \tag{3.10}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \text{(cell update)} \tag{3.11}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(output gate)} \tag{3.12}$$

$$h_t = o_t \odot \tanh(C_t) \quad \text{(hidden state)} \tag{3.13}$$

The gates let the network decide what to remember and what to forget over potentially long sequences.

The architecture is deliberately kept moderate:

- Two LSTM layers with 64 and 32 units respectively
- Batch normalisation after each LSTM layer
- Dropout (rate 0.15) for regularisation
- A dense layer with 16 units and ReLU activation
- Final output layer with one unit (the volatility forecast)

The model uses a lookback window of 30 trading days, the Adam optimiser with learning rate 0.001, and early stopping with patience of 25 epochs. The final 10% of training data serves as a validation set. This configuration is fairly standard for financial time series: not extensively optimised, but reasonable.

3.2.6. Feature Engineering for ML Models

The machine learning models (Random Forest, XGBoost, LSTM) use the following features:

- Lagged returns: $r_{t-1}, r_{t-2}, \ldots, r_{t-22}$ (roughly a trading month of history)
- Realised volatility at different horizons: 5-day, 10-day, and 21-day rolling windows
- Lagged volatility: RV_{t-1} through RV_{t-5}

For LSTM, all features are standardised to have zero mean and unit variance, with the mean

and standard deviation computed on the training set only and then applied unchanged to the test set. This prevents information leakage and helps with training stability.

3.3. Experimental Design

3.3.1. Training and Testing

The out-of-sample evaluation is straightforward. For each asset:

1. Split the data 80/20 into training and test periods.
2. For GARCH models, use rolling one-step-ahead forecasts. At each test point, re-estimate the model on all data up to that point and forecast the next day.
3. For ML models, train once on the training set and generate forecasts for the entire test period.

This constitutes a deliberate asymmetric validation strategy. The protocols differ because the models differ. GARCH estimation is lightweight: re-fitting a two-parameter variance equation on each new day of data takes fractions of a second. Deep learning retraining is not lightweight. In any realistic deployment, a practitioner would re-estimate GARCH daily and retrain a neural network far less frequently, if at all. The design therefore reflects operational reality rather than laboratory symmetry. If anything, it sets a higher bar for the ML models: they must perform well without the benefit of continuous adaptation.

The target variable is 5-day forward realised volatility, consistent across all models.

3.3.2. Reproducibility

Random seeds are fixed throughout:

- NumPy: 42
- TensorFlow: 42
- Scikit-learn models use fixed random states

This allows exact replication of the results.

3.4. Evaluation Metrics

Forecasts are evaluated using three standard metrics:

Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{\sigma}_t - \sigma_t)^2} \tag{3.14}$$

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|\hat{\sigma}_t - \sigma_t| \tag{3.15}$$

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{\hat{\sigma}_t - \sigma_t}{\sigma_t}\right| \tag{3.16}$$

RMSE is the primary metric. It penalises large errors heavily, which matters in practice: underestimating a volatility spike can be far more costly than being slightly off during calm periods.

3.4.1. Statistical Testing

Raw error metrics can be misleading: what appears to be a meaningful difference may simply reflect sampling variation. To address this, the Diebold-Mariano test (Diebold & Mariano, 1995) is employed. The null hypothesis is that two forecasts have equal accuracy:

$$H_0 : E[d_t] = 0 \quad \text{where} \quad d_t = e_{1,t}^2 - e_{2,t}^2 \tag{3.17}$$

The test statistic is:

$$DM = \frac{\bar{d}}{\sqrt{\hat{V}(\bar{d})}} \sim N(0,1) \tag{3.18}$$

A significantly positive DM statistic means the second model beats the first; negative means the first is better. This provides some discipline beyond just eyeballing RMSE rankings.

4. RESULTS AND DISCUSSION

The results do not fit a simple "ML beats GARCH" narrative.

4.1. Forecasting Performance

Table 4.1 reports the out-of-sample root mean squared error (RMSE) for each model-asset combination. RMSE is the primary metric because it penalises large errors heavily, which matters when the concern is missing a volatility spike.

Table 4.1.: Out-of-Sample Forecasting Performance (RMSE)

Asset	GARCH(1,1)	EGARCH	Random Forest	XGBoost	LSTM
BTC	0.2076	0.2097	0.1855	**0.1848**	0.1972
ETH	0.3032	**0.2979**	0.3128	0.3291	0.3007
SPX	0.1392	0.1294	0.1362	**0.1286**	0.1430
VIX†	0.9758	0.9222	**0.7916**	0.8662	0.9174

Note: Bold indicates lowest RMSE for each asset.
† VIX results require separate interpretation; see Section 4.4.
RMSE expressed in annualised volatility (decimal form). A value of 0.10 corresponds to 10 percentage points.

For Bitcoin, tree-based methods perform best, with XGBoost achieving an RMSE of 0.1848, and both Random Forest and XGBoost significantly outperform GARCH; LSTM improves on GARCH by RMSE but not significantly. For Ethereum, EGARCH achieves the strongest performance while machine learning methods offer no significant improvement. For the S&P 500, XGBoost achieves the lowest raw RMSE (0.1286), though the Diebold-Mariano test does not confirm this as significant; EGARCH is the only method to significantly outperform GARCH for equities. LSTM performs worse than GARCH outright for SPX. The VIX results favour Random Forest.

From a practical standpoint, the Bitcoin improvement is the most actionable: an 11 per cent RMSE reduction from machine learning, confirmed as statistically significant, could meaningfully affect risk management decisions.

4.1.1. Statistical Significance: Diebold-Mariano Tests

Point estimates of forecast accuracy, while informative, do not establish whether observed differences reflect genuine performance disparities or sampling variation. The Diebold-Mariano test addresses this directly. Table 4.2 summarises the results, comparing each model against GARCH(1,1) as the benchmark.

The formal tests mostly confirm the rankings, but with some important qualifications:

Table 4.2.: Diebold-Mariano Test Results

Comparison	BTC	ETH	SPX	VIX
RF vs GARCH	***	ns	ns	***
XGBoost vs GARCH	***	ns	ns	**
EGARCH vs GARCH	*	**	***	***
LSTM vs GARCH	ns	ns	ns	ns

*** p<0.01, ** p<0.05, * p<0.10, ns = not significant.
Negative statistic indicates first model outperforms benchmark (lower squared errors than GARCH).

- **Bitcoin**: Random Forest and XGBoost achieve statistically significant improvements over GARCH(1,1) at the 1 per cent level. XGBoost achieves the lowest RMSE. LSTM produces a lower RMSE than GARCH but the difference is not statistically significant. EGARCH, by contrast, performs marginally worse than the symmetric baseline (DM=+1.65, p=0.10), with GARCH outperforming EGARCH for this asset. These results suggest that tree-based machine learning approaches capture patterns in Bitcoin volatility that GARCH misses, while asymmetric volatility specifications offer no advantage here.
- **Ethereum**: Neither the tree-based methods nor LSTM demonstrates statistically significant improvement over GARCH. EGARCH, however, achieves a significant improvement over the symmetric specification (p<0.05), consistent with the presence of asymmetric volatility dynamics.
- **S&P 500**: EGARCH achieves statistically significant improvement over GARCH(1,1) at the 1 per cent level. No machine learning method significantly outperforms GARCH, and LSTM produces higher forecast errors than the GARCH baseline. The asymmetric specification's ability to capture the leverage effect appears to be the dominant feature of equity volatility that symmetric GARCH misses.
- **VIX**: Random Forest and XGBoost both significantly outperform GARCH, as does EGARCH. The LSTM shows no significant difference from GARCH. As discussed in Section 4.4, these results require careful interpretation.

4.1.2. Cryptocurrency Assets

The Bitcoin results favour tree-based machine learning methods. XGBoost achieves the lowest RMSE (0.1848), with Random Forest close behind (0.1855) and LSTM further at 0.1972. The Diebold-Mariano tests confirm that Random Forest and XGBoost significantly outperform GARCH(1,1) at the 1 per cent level; LSTM improves on GARCH in RMSE terms but the difference is not statistically significant. The approximately 11 per cent improvement from XGBoost suggests that tree-based approaches capture patterns in Bitcoin volatility that traditional econometric methods do not.

The Ethereum results are a different story. EGARCH achieves the lowest RMSE (0.2979), whilst machine learning methods offer no significant improvement over the GARCH baseline. The tree-based methods and LSTM all produce higher forecast errors than the asymmetric econometric specification.

This divergence within the same asset class raises questions about the transferability of modelling approaches. Bitcoin and Ethereum share certain structural characteristics: both trade continuously, both exhibit high volatility relative to traditional assets, and both attract broadly similar investor constituencies. Yet the patterns that machine learning exploits in Bitcoin volatility do not appear to exist in Ethereum, or perhaps exist in a form that these methods cannot capture effectively.

One potential explanation relates to market microstructure. Bitcoin, as the largest cryptocurrency, might exhibit more systematic trading patterns amenable to machine learning approaches. Ethereum's volatility dynamics might be driven more by idiosyncratic factors (protocol upgrades, decentralised finance activity, network congestion) that do not produce consistent patterns. The significance of EGARCH's improvement over symmetric GARCH ($p<0.05$) suggests that asymmetric volatility responses are the dominant feature in Ethereum markets. This interpretation is tentative; the data here do not settle it.

4.1.3. Traditional Assets: S&P 500

The S&P 500 results tell a different story from the cryptocurrency assets. The only statistically significant improvement over GARCH comes from EGARCH ($p<0.01$), consistent with decades of evidence on equity volatility: negative returns raise subsequent volatility more than positive returns of equivalent magnitude, and symmetric GARCH cannot capture this. XGBoost achieves the lowest raw RMSE (0.1286 versus GARCH's 0.1392), but the Diebold-Mariano test does not confirm this as significant ($p=0.45$). LSTM performs worse than GARCH outright, with an RMSE of 0.1430 against GARCH's 0.1392.

The implication is that, for mature equity markets, getting the model structure right matters more than adding machine learning complexity. The leverage effect is well understood, and EGARCH captures it directly. Machine learning methods do not appear to exploit additional structure in the data beyond what the asymmetric specification already accounts for.

For practitioners, these results suggest that EGARCH, not neural networks, is the appropriate upgrade from basic GARCH for equity volatility forecasting.

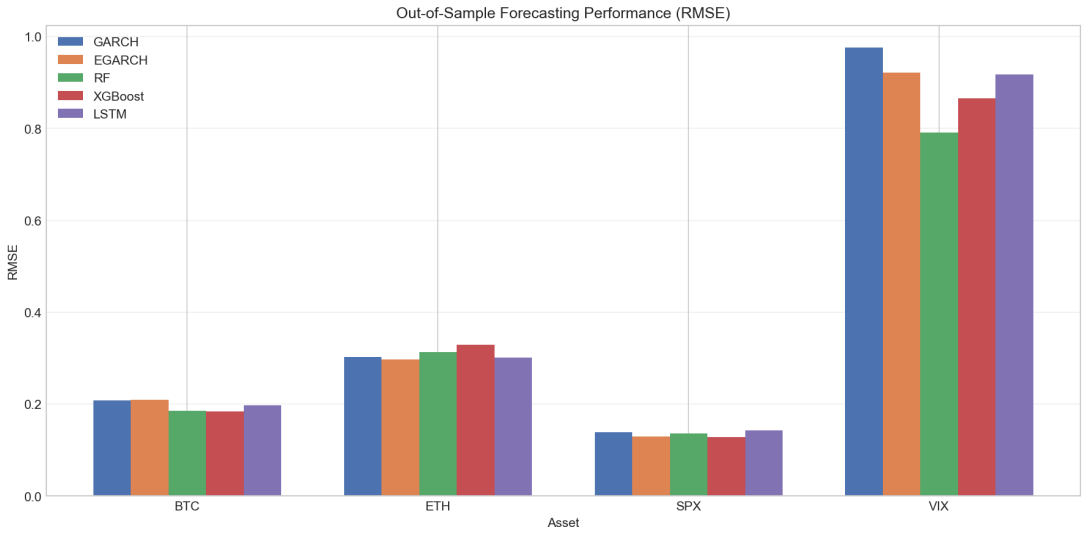

Figure 4.1.: Model performance comparison across assets (RMSE, lower values indicate superior performance)

4.2. Comparative Analysis

4.2.1. Cryptocurrency versus Traditional Markets

Table 4.3 presents average performance metrics by asset class. These aggregates, however, obscure important heterogeneity.

Table 4.3.: Average RMSE by Asset Class (excluding VIX)

Asset Class	GARCH	EGARCH	RF	XGBoost	LSTM
Crypto (BTC, ETH)	0.2554	0.2538	0.2492	0.2570	**0.2475**
Traditional (SPX)	0.1392	0.1294	0.1362	**0.1286**	0.1430

Note: Bold indicates lowest average RMSE for each asset class.

The cryptocurrency average shows LSTM performing marginally best, with RF close behind. This average, however, masks important heterogeneity: machine learning methods significantly outperform GARCH for Bitcoin but not for Ethereum. Aggregation at the asset-class level thus obscures rather than illuminates the key pattern.

This observation carries methodological implications. Studies that report average performance across cryptocurrency assets miss important within-class variation. Model selection needs to proceed at the level of individual assets rather than asset categories.

4.2.2. Model Complexity and Performance

The relationship between model complexity and forecast accuracy does not follow a monotonic pattern. For Bitcoin, tree-based methods perform best and are the only ML approaches that

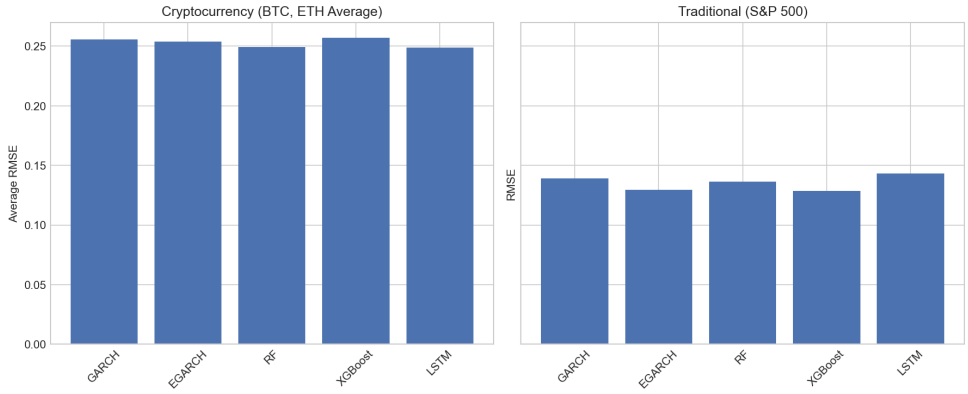

Figure 4.2.: Average model performance: Cryptocurrency versus traditional assets

significantly outperform GARCH. For Ethereum and VIX, simpler methods match or exceed the neural network. For the S&P 500, the most complex model (LSTM) performs worst, and only EGARCH, which adds targeted asymmetry rather than general complexity, achieves a meaningful improvement. Complexity helps only when the data contain exploitable patterns of the right kind.

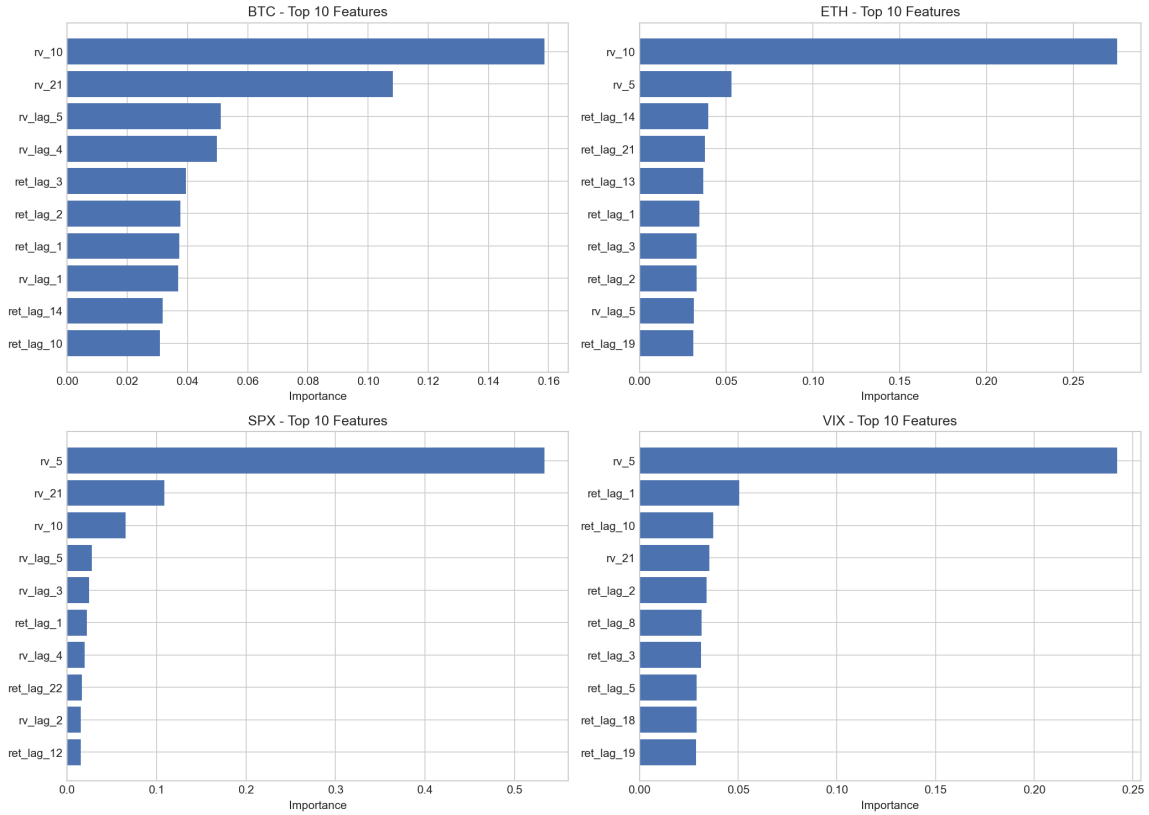

Figure 4.3.: Random Forest feature importance by asset

Figure 4.3 displays feature importance measures from the Random Forest models. Across all assets, lagged volatility measures dominate, with recent returns contributing to a lesser degree. This pattern is consistent with the well-documented persistence of volatility: historical volatility

remains the most informative predictor of future volatility. The relative homogeneity of feature importance across assets suggests that the differential performance of LSTM likely arises from temporal pattern exploitation rather than differences in relevant information.

4.3. Discussion

4.3.1. Research Questions

The findings permit responses to the research questions posed in Chapter 1.

RQ1: Do machine learning methods achieve statistically significant improvements over GARCH-family models for volatility forecasting in cryptocurrency markets?

The answer is asset-dependent. For Bitcoin, Random Forest and XGBoost significantly outperform GARCH(1,1), with XGBoost achieving the lowest RMSE (0.1848); LSTM produces a lower RMSE than GARCH but the difference is not statistically significant. The approximately 11 per cent improvement from XGBoost is statistically significant and potentially economically meaningful. For Ethereum, no machine learning method significantly outperforms GARCH; EGARCH achieves the best performance. A categorical answer regarding cryptocurrency markets is not supported by the evidence.

RQ2: Do any performance advantages observed in cryptocurrency markets extend to traditional equity markets?

Not through machine learning. No ML method achieves a statistically significant improvement over GARCH for the S&P 500. EGARCH does, and substantially so ($p<0.01$), by capturing the leverage effect that symmetric GARCH misses. LSTM performs worse than GARCH outright. The ML advantage observed in Bitcoin does not extend to equities; if anything, the equity results favour simpler, structurally motivated specifications.

RQ3: Among machine learning approaches, do more complex architectures such as LSTM provide meaningful improvements over simpler ensemble methods?

The answer varies by asset. For Bitcoin, tree-based methods achieve lower RMSE than LSTM and are the only ML approaches that significantly outperform GARCH. For Ethereum and VIX, LSTM offers no significant improvement over simpler methods. For the S&P 500, LSTM is the worst-performing model. Across assets, greater complexity does not translate to better forecasts; in equities, it actively hurts.

4.3.2. Implications for Practice

A few observations for practitioners:

1. **Asset-specific validation is essential**. The divergence between Bitcoin and Ethereum results demonstrates that performance within one asset does not predict performance

in another, even within the same asset class. Model selection should proceed through asset-by-asset testing rather than categorical assumptions.

2. **Statistical significance testing provides essential information**. Some apparent RMSE differences do not survive formal testing. Reliance on point estimates alone risks adopting methods that offer no genuine improvement.

For equity markets, EGARCH is the appropriate upgrade: it captures the leverage effect directly, and machine learning methods including LSTM do not improve on it. LSTM performs worse than GARCH outright for the S&P 500, and the computational investment in neural network implementation is not justified for this asset class. In crypto, the statistically significant gains come from tree-based methods on Bitcoin; LSTM does not reliably add value. Deployment should follow per-asset validation, not categorical assumptions about where neural networks help.

4.3.3. Economic Significance

Statistical significance does not automatically establish economic significance. The 11 per cent RMSE improvement for Bitcoin is statistically confirmed and could affect risk management decisions and position sizing for cryptocurrency portfolios. For equities, EGARCH's statistically significant improvement over GARCH could affect Value-at-Risk calculations, though the magnitude is modest and the practical effect unquantified here. Whether any of these improvements translate to realised gains in risk-adjusted returns remains an empirical question that the present analysis does not address directly.

4.4. VIX: A Special Case

4.4.1. Distinctive Properties

The VIX differs fundamentally from the other assets examined. Whilst Bitcoin, Ethereum, and the S&P 500 are price-based instruments whose volatility is derived from return calculations, VIX already represents a volatility measure: specifically, the 30-day implied volatility of S&P 500 options as calculated from options prices. Forecasting VIX volatility therefore involves predicting the volatility of volatility, a second-order quantity with distinct statistical properties.

VIX exhibits several characteristics relevant to forecasting:

- Mean reversion towards a long-run average of approximately 20.
- Asymmetric dynamics: rapid increases during periods of market stress followed by gradual declines.
- Strong negative correlation with S&P 500 returns (Figure 4.4).
- Bounded behaviour: practical floors and ceilings constrain the range of outcomes.

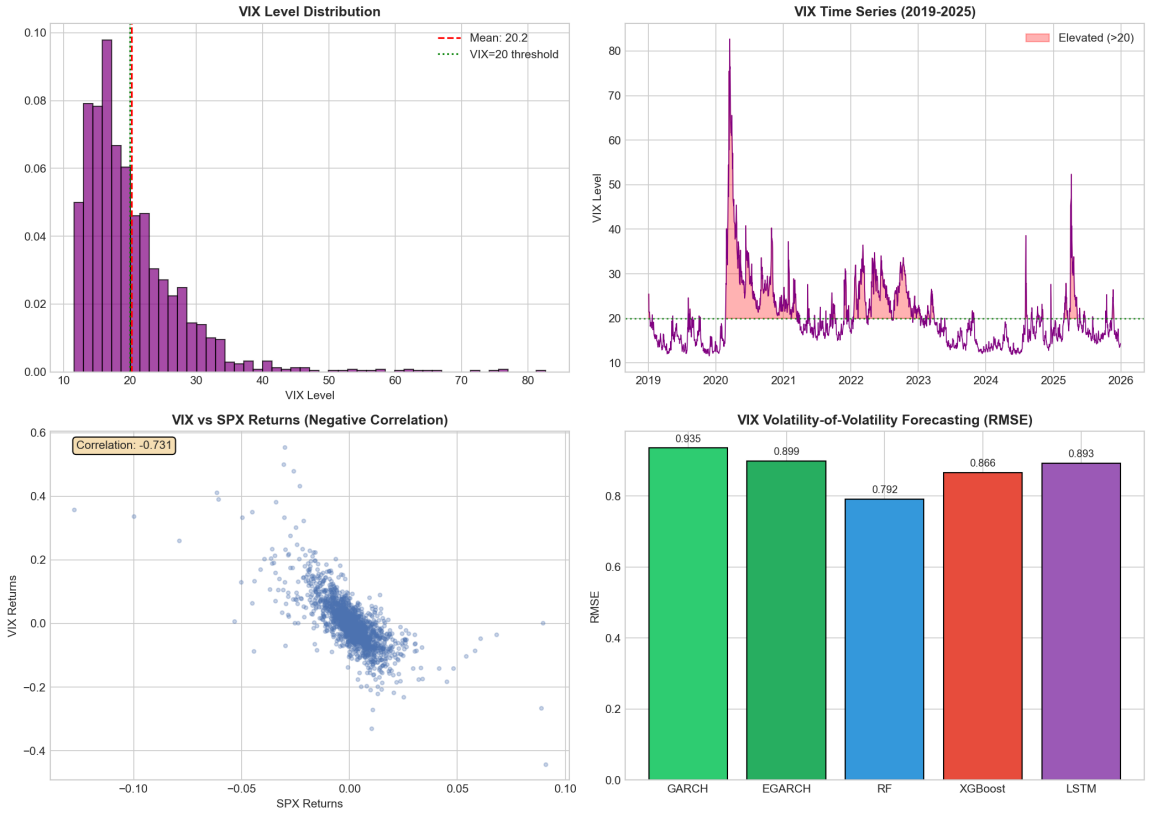

Figure 4.4.: VIX characteristics (2019–2025)

4.4.2. Interpretation of Results

Given these distinctive properties, the VIX results warrant separate consideration. Random Forest achieves the lowest RMSE (0.7916) and significantly outperforms GARCH(1,1). EGARCH also demonstrates significant improvement. LSTM, however, shows no significant difference from GARCH.

The elevated RMSE values across all methods (ranging from 0.79 to 0.98) reflect the inherent difficulty of forecasting volatility-of-volatility. The finding that tree-based methods outperform sequence models likely relates to the bounded, mean-reverting nature of VIX, which differs from the dynamics of return-based volatility.

4.5. Robustness Considerations

4.5.1. Training Methodology

One asymmetry in the experimental design deserves attention. GARCH models were re-estimated at each forecasting step, incorporating all available data up to the forecast origin. Machine learning models, by contrast, were trained once on the initial training set and subsequently held fixed. This approach reflects typical deployment practice (rolling neural network re-estimation would

entail substantial computational cost) but confers an adaptability advantage to econometric methods.

The finding that EGARCH outperforms machine learning for the S&P 500 might partly reflect this asymmetry. Whether periodic machine learning retraining would alter the conclusions is a question for future investigation.

4.5.2. Additional Considerations

A few additional factors affect generalisability:

1. **Hyperparameter selection**: The machine learning models employ relatively standard configurations. Alternative hyperparameter choices might yield different results.
2. **Sample period**: The 2019–2025 period encompasses the COVID-19 market disruption and multiple cryptocurrency market cycles. Performance in other periods may differ.
3. **Neural network architecture**: More sophisticated designs, such as attention mechanisms or Transformer architectures, remain untested.

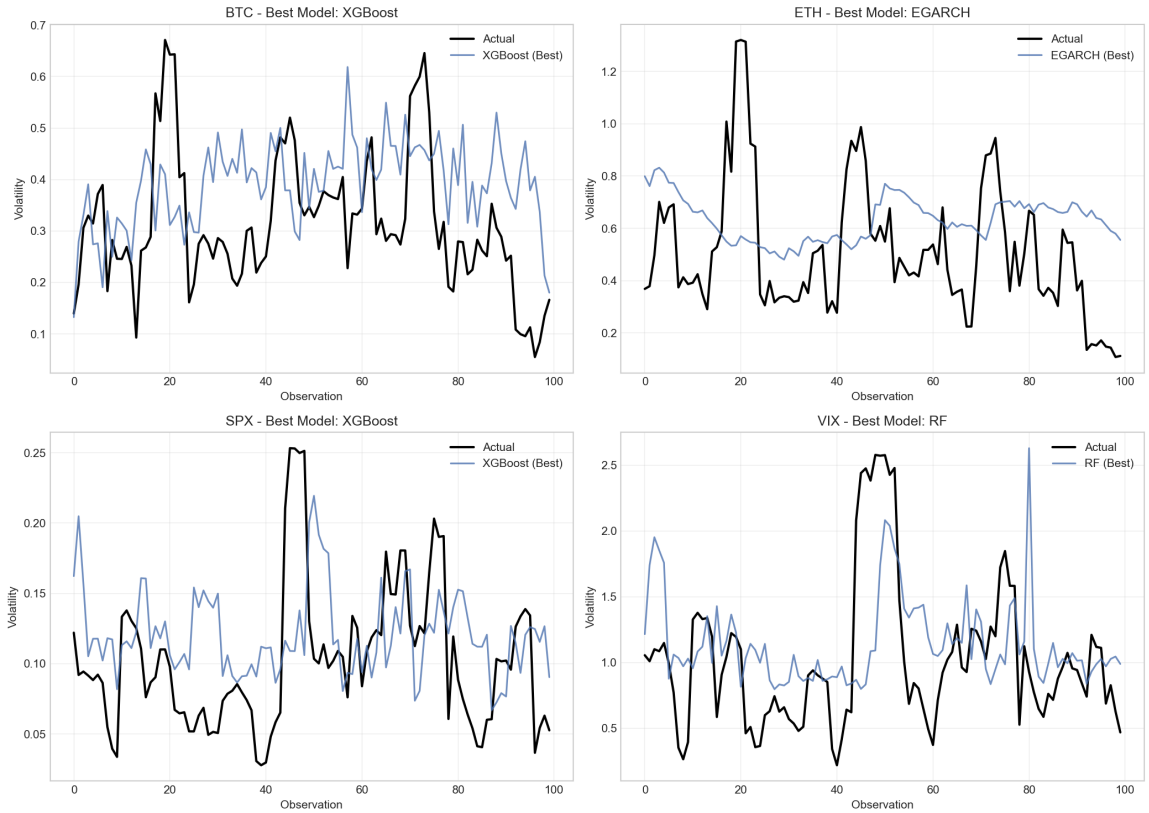

Figure 4.5.: Best model forecasts versus realised volatility (final 100 observations)

5. CONCLUSION

5.1. Summary of Findings

This thesis compared econometric and machine learning approaches to volatility forecasting across cryptocurrency and traditional assets, using daily data from 2019 to 2025 and Diebold-Mariano tests to distinguish real performance differences from noise. The results do not support any single model as universally better.

For the S&P 500, the headline result is negative: no machine learning method significantly improves on GARCH. LSTM performs worse than GARCH outright. Only EGARCH, which captures the leverage effect directly, achieves a statistically significant improvement ($p<0.01$). In mature equity markets, model structure matters more than model complexity.

For Bitcoin, the tree-based machine learning methods significantly outperform GARCH(1,1), with XGBoost achieving the lowest RMSE (0.1848) and Random Forest close behind (0.1855). LSTM produces a lower RMSE than GARCH (0.1972) but the difference is not statistically significant. The approximately 11 per cent improvement from XGBoost is statistically significant. For practitioners concerned with Bitcoin volatility, these findings suggest that tree-based machine learning approaches offer measurable benefits.

These findings do not generalise within the cryptocurrency asset class. For Ethereum, no machine learning method significantly improves upon GARCH, and EGARCH achieves the lowest errors. The divergence between Bitcoin and Ethereum results (machine learning methods significantly improving forecasts for one cryptocurrency but not the other) constitutes one of the more striking findings of this study.

Model selection, these findings suggest, needs to proceed at the level of individual assets rather than asset categories. The temptation to generalise from one cryptocurrency to another, or to assume that methods ineffective for one asset will prove similarly ineffective for others, is not supported by the evidence presented here.

5.2. Contributions

This thesis makes several contributions to the volatility forecasting literature:

1. **Cross-asset comparison with consistent methodology**. By applying identical methods across both cryptocurrency and traditional assets, the analysis permits direct comparison where most existing studies focus on one domain or the other. This design controls for methodological differences that complicate interpretation of findings across studies.
2. **Documentation of within-class variation**. The finding that machine learning methods significantly improve Bitcoin forecasts but not Ethereum forecasts challenges categorical

assumptions about cryptocurrency volatility modelling. Performance appears to depend on asset-specific characteristics that do not align neatly with asset-class boundaries.

3. **Evidence that model structure outperforms model complexity for equities**. For the S&P 500, EGARCH is the only method to significantly outperform GARCH. Machine learning methods, including LSTM, do not add value over the asymmetric econometric specification. The leverage effect appears to be the dominant feature of equity volatility, and EGARCH captures it directly.
4. **Reproducible analysis**. The analysis employs fixed random seeds and documented procedures, permitting replication and extension.

5.3. Limitations

The main limitations on interpretation are:

1. **LSTM architecture**. The neural network specification employed (two layers of 64 and 32 units with 30-day lookback) represents one of many possible configurations. Attention mechanisms, Transformer architectures, or alternative hyperparameter selections might yield different results. Whether architectural modifications would alter the Bitcoin-Ethereum divergence remains an open question.
2. **Training asymmetry**. GARCH models were re-estimated at each forecasting step, whilst machine learning models were trained once on the initial training set. This approach reflects typical deployment practice but gives econometric methods more capacity to adapt to recent market developments.
3. **Forecast horizon**. The analysis focuses exclusively on 5-day ahead volatility forecasts. Different horizons (shorter or longer) might favour different models. LSTM's capacity to capture extended temporal dependencies might prove more advantageous at longer forecast horizons.
4. **GARCH-ML horizon mismatch**. The ML models target 5-day forward realised volatility directly, while the GARCH and EGARCH implementations here produce rolling one-step-ahead conditional variance forecasts reported on the same annualised scale. For a mean-reverting GARCH process the two quantities are close but not identical. Aggregating GARCH variance over a 5-day horizon would remove the asymmetry and is a natural refinement.
5. **Annualisation convention**. All realised volatility measures are annualised with a factor of $\sqrt{252}$, treating each asset as if it traded 252 days per year. This understates crypto annualised volatility in absolute terms, since BTC and ETH trade roughly 365 days per year. Because the target and all forecasts share the same scaling, this does not affect RMSE, MAE, MAPE, or Diebold-Mariano comparisons; it only affects the absolute level of

reported volatilities.

6. **Feature specification**. Machine learning models employed only price-based features: lagged returns and realised volatility measures. Alternative feature sets (incorporating sentiment indicators, on-chain metrics for cryptocurrencies, or options-implied volatility) might improve machine learning performance.
7. **Sample period**. The 2019–2025 period includes several unusual market episodes, including the COVID-19 market disruption and multiple cryptocurrency boom-bust cycles. Whether the findings generalise to other sample periods remains uncertain.

5.4. Implications for Practice

A few points for practitioners:

1. **Asset-specific validation**. The divergence between Bitcoin and Ethereum results demonstrates that performance for one asset does not predict performance for another, even within the same asset class. Model selection should proceed through testing on each asset of interest rather than reliance on categorical assumptions.
2. **Statistical significance testing**. Some apparent improvements in RMSE do not achieve statistical significance under formal testing. Adopting methods based solely on point estimates risks implementing approaches that offer no genuine advantage.
3. **For equity markets, use EGARCH not LSTM**. EGARCH is the only method to significantly outperform GARCH for the S&P 500. LSTM performs worse than GARCH outright. The computational investment in neural network implementation is not supported by the evidence for this asset class.
4. **Tree-based ML is worth testing for crypto, but validate per asset**. Random Forest and XGBoost significantly outperform GARCH for Bitcoin, but no ML method significantly improves on GARCH for Ethereum. LSTM produces a lower RMSE than GARCH for Bitcoin but the difference is not statistically significant. Deployment decisions should be based on per-asset validation, not on categorical assumptions.

5.5. Directions for Future Research

The clearest next step is testing more sophisticated neural network architectures. Transformers and attention mechanisms have shown strong results in sequence modelling elsewhere, and whether they improve on LSTM for assets where it currently falls short is a natural empirical question. Periodic retraining of machine learning models would also address the training asymmetry described above: updating models monthly or quarterly would allow fairer comparison with rolling GARCH estimation, at the cost of considerable computation. Intraday data might favour

sequence models more consistently than daily frequency does, since the temporal structure available to the models would be richer. Hybrid approaches, such as using GARCH forecasts as features for machine learning models or feeding GARCH residuals into an LSTM, represent another avenue with some theoretical appeal. The most practically valuable extension, though, would translate the statistical improvements found here into economic terms: actual trading performance, option pricing accuracy, or risk management outcomes. That is ultimately what motivates volatility forecasting.

5.6. Concluding Remarks

Machine learning does not straightforwardly beat econometric methods for volatility forecasting. For Bitcoin, the tree-based machine learning methods significantly outperform GARCH, while LSTM does not. For the S&P 500, no ML method significantly beats GARCH and LSTM is the worst-performing model. For Ethereum, the picture is similarly mixed. The findings point in different directions depending on the asset.

These findings suggest that volatility dynamics are more heterogeneous than is often assumed. Different assets, driven by different market participants operating with different information and objectives, appear to require different modelling approaches. The search for a single optimal method is probably less productive than developing asset-specific approaches, selected through rigorous out-of-sample testing with appropriate statistical validation.

What works depends on the specific dynamics of each asset. The most reliable path to model selection remains empirical evaluation, with careful attention to statistical significance, conducted on an asset-by-asset basis.

BIBLIOGRAPHICAL REFERENCES

Aras, S. (2021). Stacking hybrid garch models for forecasting bitcoin volatility. *Expert Systems with Applications*, *174*, 114747. https://doi.org/10.1016/j.eswa.2021.114747

Baur, D. G., Dimpfl, T., & Kuck, K. (2018). Bitcoin, gold and the us dollar – a replication and extension. *Finance Research Letters*, *25*, 103–110. https://doi.org/10.1016/j.frl.2017.10.012

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Bollerslev, T., Chou, R. Y., & Kroner, K. F. (1992). Arch modeling in finance: A review of the theory and empirical evidence. *Journal of Econometrics*, *52*(1-2), 5–59. https://doi.org/10.1016/0304-4076(92)90064-X

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Bucci, A. (2020). Realized volatility forecasting with neural networks. *Journal of Financial Econometrics*, *18*(3), 502–531. https://doi.org/10.1093/jjfinec/nbaa008

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Christensen, K., Siggaard, M., & Veliyev, B. (2023). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, *21*(5), 1680–1727. https://doi.org/10.1093/jjfinec/nbac020

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263. https://doi.org/10.1080/07350015.1995.10524599

Dyhrberg, A. H. (2016). Bitcoin, gold and the dollar – a garch volatility analysis. *Finance Research Letters*, *16*, 85–92. https://doi.org/10.1016/j.frl.2015.10.008

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, *50*(4), 987–1007. https://doi.org/10.2307/1912773

Engle, R. F. (2001). Garch 101: The use of arch/garch models in applied econometrics. *Journal of Economic Perspectives*, *15*(4), 157–168. https://doi.org/10.1257/jep.15.4.157

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hull, J. C. (2018). *Options, futures, and other derivatives* (10th ed.). Pearson.

Katsiampa, P. (2017). Volatility estimation for bitcoin: A comparison of garch models. *Economics Letters*, *158*, 3–6. https://doi.org/10.1016/j.econlet.2017.06.023

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. https://doi.org/10.1016/j.eswa.2018.03.002

Liu, Y. (2019). Novel volatility forecasting using deep learning-long short term memory recurrent neural networks. *Expert Systems with Applications*, *132*, 99–109. https://doi.org/10.1016/j.eswa.2019.04.038

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, *59*(2), 347–370. https://doi.org/10.2307/2938260

Poon, S.-H., & Granger, C. W. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic Literature*, *41*(2), 478–539. https://doi.org/10.1257/002205103765762743

Rahimikia, E., & Poon, S.-H. (2021). *Machine learning for realised volatility forecasting* (Working Paper). SSRN. https://doi.org/10.2139/ssrn.3707796

Zahid, M., Iqbal, F., & Koutmos, D. (2022). Forecasting bitcoin volatility using hybrid garch models with machine learning. *Risks*, *10*(12), 237. https://doi.org/10.3390/risks10120237

SUPPLEMENTARY RESULTS

This appendix provides additional results to complement the main analysis, including alternative error metrics, model hyperparameters, and extended Diebold-Mariano test results.

1. Alternative Error Metrics

Table A1 presents the Mean Absolute Error (MAE) for all model-asset combinations, providing a complementary view to the RMSE results reported in the main text.

Table A1.: Out-of-Sample Forecasting Performance (MAE)

Asset	GARCH(1,1)	EGARCH	Random Forest	XGBoost	LSTM
BTC	0.1781	0.1796	0.1455	0.1490	**0.1451**
ETH	0.2511	0.2470	0.2386	0.2428	**0.2352**
SPX	0.0855	0.0813	**0.0695**	0.0711	0.0852
VIX	0.6603	0.6357	**0.4874**	0.5463	0.5557

Note: Bold values indicate the best performing model for each asset.
MAE values are on annualized volatility scale (decimal form).

Table A2 presents the Mean Absolute Percentage Error (MAPE), which provides a scale-independent measure of forecasting accuracy.

Table A2.: Out-of-Sample Forecasting Performance (MAPE, %)

Asset	GARCH(1,1)	EGARCH	Random Forest	XGBoost	LSTM
BTC	81.2	82.5	64.3	63.9	**55.3**
ETH	69.3	69.3	62.4	61.8	**58.9**
SPX	85.7	82.2	**60.0**	62.4	72.2
VIX	74.9	72.0	50.9	56.2	**44.4**

Note: Bold values indicate the best performing model for each asset.

The MAE and MAPE results are broadly consistent with the RMSE findings: tree-based methods dominate for SPX and VIX, while LSTM achieves the lowest MAE for Bitcoin and Ethereum and the lowest MAPE for Bitcoin, Ethereum, and VIX. The differing rankings across error metrics reflect LSTM's tendency toward proportional accuracy, whereas tree-based methods minimise squared loss more reliably.

2. Model Hyperparameters

Table A3 summarizes the hyperparameters used for each model. These represent standard configurations without extensive tuning, consistent with our focus on practical deployability.

Table A3.: Model Hyperparameters

Model	Parameter	Value
GARCH(1,1)	Distribution	Normal
	Mean model	Constant
	Estimation	Quasi-MLE
EGARCH(1,1)	Distribution	Normal
	Mean model	Constant
	Estimation	Quasi-MLE
Random Forest	n_estimators	100
	max_depth	10
	min_samples_split	5
	random_state	42
XGBoost	n_estimators	100
	max_depth	6
	learning_rate	0.1
	random_state	42
LSTM	LSTM layers	2 (stacked)
	LSTM units	64, 32
	Batch normalization	After each LSTM layer
	Dropout rate	0.15
	Dense units	16
	Lookback window	30 days
	Optimizer	Adam (lr=0.001)
	Early stopping patience	25 epochs

3. Extended Diebold-Mariano Test Results

Table A4 provides the full Diebold-Mariano test statistics (not just significance levels) for key model comparisons.

Table A4.: Diebold-Mariano Test Statistics (Full Results)

Comparison	BTC	ETH	SPX	VIX
RF vs GARCH	-2.67^{***}	0.98	-0.18	-3.60^{***}
XGBoost vs GARCH	-4.47^{***}	1.60	-0.75	-2.11^{**}
EGARCH vs GARCH	1.65^{*}	-2.35^{**}	-3.97^{***}	-3.13^{***}
LSTM vs GARCH	-1.42	-0.77	-0.04	-0.54

Note: *** p<0.01, ** p<0.05, * p<0.10.
Negative values indicate the first model outperforms the second.
Test conducted using squared error loss differential.

Key observations from the extended results:

- For BTC, RF and XGBoost significantly outperform GARCH, with XGBoost showing the strongest improvement (DM = -4.47). LSTM produces a lower RMSE but the difference is not statistically significant (DM = -1.42).
- For ETH, only EGARCH significantly outperforms GARCH (DM = -2.35). Neither tree-based methods nor LSTM show significant improvement.
- For SPX, only EGARCH significantly outperforms GARCH (DM = -3.97). Tree-based methods and LSTM show no significant improvement; on matched samples LSTM is essentially indistinguishable from GARCH (DM = -0.04).
- For VIX, RF and XGBoost significantly outperform GARCH, as does EGARCH. LSTM shows no significant difference.

4. Feature Set Description

Table A5 lists all features used in the machine learning models.

Table A5.: Feature Set for Machine Learning Models

Feature Group	Variables
Lagged Returns	$r_{t-1}, r_{t-2}, \ldots, r_{t-22}$ (22 features)
Realized Volatility	rv_5 (5-day), rv_10 (10-day), rv_21 (21-day)
Lagged Volatility	rv_lag_1, rv_lag_2, rv_lag_3, rv_lag_4, rv_lag_5

Note: All features are constructed using only information available at time $t-1$ to avoid lookahead bias.
LSTM models use standardized features (zero mean, unit variance).

Data with Purpose.

Campus de Campolide, 1070-312 Lisboa, Portugal
novaims.unl.pt