Comparing Econometric, Deep Learning and Hybrid Models for S&P 500 Volatility Forecasting

Dissertation
COMP702 - M.Sc. project (2025/26)

Submitted by
Roland Oteniya
201884707

under the supervision of
Achilleas Koufonikos

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF LIVERPOOL

Student Declaration

I confirm that I have read and understood the University’s Academic Integrity Policy. I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.

I confirm that I have not copied material from another source nor committed plagiarism nor fabricated data when completing the attached piece of work. I confirm that I have not previously presented the work or part thereof for assessment for another University of Liverpool module. I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work. I confirm that I have not incorporated into this assignment material that has been submitted by me or any other person in support of a successful application for a degree or this or any other university or degree-awarding body.

SIGNATURE _______
DATE 12 September 2025

Acknowledgements

Glory be to God.
Thanks to my supervisor, Achilleas Koufonikos for his guidance.
Thanks to my family for their constant support.
Thanks to myself for putting in the work.

Comparing Econometric, Deep Learning and Hybrid Models for S&P 500 Volatility Forecasting

Abstract

Accurate forecasting of stock market volatility is a critical task for financial risk management and derivative pricing. This dissertation addresses this challenge with a systematic comparative analysis of different types of forecasting models on the S&P 500 index. Traditional econometric models from the GARCH family, Long Short-Term Memory (LSTM) networks, and sequential hybrid econometric-LSTM models were developed and evaluated using Root Mean Squared Error.

The results from the out-of-sample test has the VIX-LSTM as the superior model, achieving a statistically significant 61.5% improvement in forecast accuracy over the baseline ARCH model. Notably, the simpler LSTM networks outperformed the more complex hybrid architectures, whose performance was found to decrease after being fitted with the econometric models' residuals. The primary contribution of this study is therefore the finding that for this forecasting task, the quality of the input features, especially the forward-looking information and market sentiment contained in the VIX, was more important to the model's success than its architectural complexity. The findings show that deep learning models represent a powerful and robust alternative to traditional econometric approaches.

Statement of Ethical Compliance

Data Category: A

Participant Category: 0

I confirm that I have read the ethical guidelines and have followed them during this project. I have used open-source data from Yahoo Finance, which is not derived from humans or animals. There has also not been any use of human participants in any activities for this project.

Contents

List of Figures

1. Introduction

1.1. Background and Motivation

Accurately forecasting market volatility is of utmost importance in quantitative finance. Volatility is a critical input for a wide range of essential tasks, including derivative pricing, risk management and optimal portfolio construction [1]. The persistent challenge of producing accurate forecasts has led to the development of different schools of thought in modelling.

The traditional approach is rooted in econometrics, with the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) model and its many variants as the academic and industry benchmark for several decades [2]. These parametric models are designed to explicitly capture the well known stylised facts of financial returns, such as volatility clustering. Recently, there has been a rise of non-parametric, data-driven techniques from machine learning. Deep learning models, especially Long Short-Term Memory (LSTM) networks, can learn complex patterns directly from time-series data, making them promising as the new standard of economic forecasting [3].

The ongoing debate over the superiority of these groups of models, coupled with the potential to enhance deep learning models with powerful, forward-looking data, is the motivation for this dissertation. This study conducts a rigorous and systematic comparison of these approaches to determine the superior forecasting model for the S&P 500 index.

1.2. Problem Statement

The primary aim of this dissertation is to conduct a systematic comparative analysis of a range of econometric, deep learning and hybrid models to find the most accurate and robust approach for forecasting the daily realised volatility of the S&P 500 index.
Here are the objectives of the project:

- To implement and evaluate GARCH-family models, including both symmetric (ARCH, GARCH) and asymmetric (EGARCH, GJR-GARCH, APARCH) specifications.

- To develop and evaluate a series of deep learning models, including a standalone LSTM and VIX-LSTM.
- To construct and test sequential hybrid models that combine the features of the best-performing econometric model with an LSTM architecture.
- To conduct a final, out-of-sample evaluation of the best-performing models from each category and to validate the results using statistical significance testing and model interpretation techniques.

1.3. Brief Description of the Approach

To address the research problem, a systematic comparative analysis was designed and implemented. Different types of models were developed. Traditional econometric models from the GARCH family, including both symmetric and asymmetric specifications; LSTM networks, including a standalone model and a model enhanced with the Volatility Index; and sequential hybrid models. These models are two stage architectures where an LSTM is trained using the residuals of an econometric model as an additional feature, designed to combine the strengths of econometric and deep learning approaches. These models were trained and evaluated on historical daily data from the S&P 500, using a chronological train-validation-test split for a robust out-of-sample assessment. The primary metric for comparing forecast accuracy was the Root Mean Squared Error (RMSE), with the final results being validated using the Diebold-Mariano test for statistical significance and SHAP analysis for model interpretability.

1.4. Outcome

The outcome of this comparative analysis was that the VIX-LSTM model is the superior model for forecasting S&P 500 realised volatility. On the unseen test data, this model achieved a statistically significant 61.5% improvement in forecast accuracy over the baseline ARCH model. A key insight from the study is that the quality of the input features proved to be more important than the complexity of the model architecture. The subsequent chapters of this dissertation detail the literature review, methodology, and full experimental results that support this conclusion.

2. Literature Review

2.1. Introduction

This chapter reviews the academic literature on financial volatility forecasting, and discusses the evolution of modelling techniques from the traditional econometric methods to modern deep learning approaches. The review begins by establishing the foundational 'stylised facts' of the financial returns, such as volatility clustering and the leverage effects, essential ideologies for economic modelling. It then examines the development of the symmetric Autoregressive Conditional Heteroskedasticity (ARCH) model and its generalisation, the GARCH model. Afterwards, the asymmetric versions developed to address the limitations of these models will be discussed, including the EGARCH, GJR-GARCH and APARCH models. This leads to discussing the non-parametric Long Short-Term Memory (LSTM) network, and assesses the impact of incorporating the CBOE Volatility Index (VIX). Finally, the rise of hybrid models that aim to combine the strengths of both econometric and deep learning models is discussed. This chapter provides the necessary context for the project's comparative analysis and identifies the research gap it aims to address.

2.2. The Stylised Facts of Financial Volatility

Understanding of financial volatility starts from recognising that asset returns are not statistically random. They exhibit a set of empirical traits, described by Cont [4] as 'stylised facts'. A fundamental stylised fact is the 'absence of significant autocorrelation in raw returns', making returns unpredictable. That said, their volatility is not random and displays several key characteristics that allow for modelling.

Volatility Clustering

The most significant of these facts is that volatility is persistent. Periods of high turbulence tend to be followed by more turbulence, while low volatility periods tend to be followed by further low volatility. Statistically, this is observed as a slowly decaying autocorrelation in absolute returns. This phenomenon is known as conditional heteroskedasticity (meaning that volatility changes over time), and its existence is the primary reason that forecasting models like the ARCH and GARCH models were developed.

Heavy Tails (Leptokurtic)

The distribution of financial returns is also leptokurtic, meaning it has a sharper peak and ‘fatter tails’ than a normal distribution. This indicates that extreme market events like crashes and surges occur far more frequently than expected. This property stresses the importance of testing alternative distribution types, such as Student’s T, for econometric models.

Leverage Effect

Lastly, the leverage effect describes the asymmetric response of volatility to shocks in financial markets. It refers to the tendency for negative shocks (bad news) to increase volatility significantly more than positive shocks (good news) of the same magnitude. This disparity is a key feature of volatility forecasting that symmetric econometric models cannot capture. Asymmetric models like EGARCH, GJR-GARCH and APARCH can model this effect.

2.3. Econometric Volatility Models

GARCH like models provide a parametric method for forecasting volatility. These models function by specifying a formal equation for the conditional variance of a time series, using its past shocks and past variance as inputs. They are designed to capture key stylised facts, particularly volatility clustering. This section begins by reviewing the foundational symmetric models, the ARCH and GARCH, before moving on to the asymmetric EGARCH, GJR-GARCH and APARCH models, which were developed to address the symmetric models’ limitations.

2.3.1. Symmetric GARCH Models

The first generation of econometric volatility models were symmetric. In these models, the conditional variance is a function of the magnitude of past shocks, but not their sign. This means that positive and negative shocks of the same size are treated identically and will have the same effect on future volatility.

2.3.1.1. ARCH Model

The Autoregressive Conditional Heteroskedasticity (ARCH), introduced by Engle [4], was one of the first econometric models created. Traditional time series models assumed constant variance (homoskedasticity), which was inconsistent with the market's nature. The ARCH model addressed the empirical problem of volatility clustering by modelling variance as conditional on past shocks. This made it the first econometric model to explicitly address heteroskedasticity in financial time series.

The mechanism of an ARCH(q) model specifies the conditional variance at time t as a linear function of the last q squared shocks or residuals. The model is represented by the following equation:

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

Where:

- σ_t^2 is the conditional variance at time t.
- α_0 is a constant term.
- q is the number of past shocks included in the model.
- ϵ_{t-i}^2 represents the squared residuals (shocks) from previous periods.
- α_i are the ARCH parameters, which must be non-negative to ensure the variance remains positive.

The model implies that if a large shock occurred in the previous period (a large ϵ_{t-1}^2), the conditional variance in the current period (σ_t^2) will be higher, thus capturing volatility persistence. ARCH's primary contribution was its ability to model volatility clustering, a development in financial econometrics. However, ARCH has two significant limitations. First, it is symmetric; because it uses squared residuals, positive and negative shocks of the same magnitude have an identical impact on volatility, meaning it cannot capture the leverage effect. Second, empirical applications often require a large number of lags (q) to capture the persistence of volatility, which can be computationally inefficient and risk violating the non-negativity constraints. These limitations motivated the development of the more parsimonious (meaning it can capture complex dynamics with relatively few parameters, reducing the risks of overfitting and improving computational efficiency) GARCH model.

2.3.1.2. GARCH Model

To address the limitations of the ARCH model, Bollerslev [2] proposed the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) model. The GARCH model is a direct extension of ARCH that provides a more flexible and parsimonious structure by incorporating a moving average component into the conditional variance equation.

The mechanism of a GARCH(p, q) model specifies the conditional variance as a function of its own past lags (p) and the lags of past squared shocks (q). The model is represented as:

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

Where:

- σ_t^2 is the conditional variance at time t.
- ϵ_{t-i}^2 represented the squared residuals from previous periods (the ARCH term).
- σ_{t-j}^2 represents the past conditional variances from previous periods (the GARCH term).
- α_0, α_i, β_j are non-negative parameters.

The variance in the current period of the GARCH model is a weighted average of a long-term average variance, the volatility shocks from previous periods (ARCH component), and the conditional variance from previous periods (GARCH component). The inclusion of the lagged variance (σ_{t-j}^2) allows the model to capture long-lasting persistence in volatility with fewer parameters than a high-order ARCH model. The GARCH(1,1) specification has become a benchmark model in financial econometrics due to its effectiveness and simplicity. However, while solving the issue of parsimony, the GARCH model inherits the critical limitation of its predecessor: it is symmetric. Its reliance on squared terms means it cannot differentiate between positive and negative shocks, meaning it cannot account for the leverage effect. This specific weakness necessitates the use of asymmetric models.

2.3.2. Asymmetric GARCH Models

The symmetric models' main weakness is their inability to account for the leverage effect. To address this, a second generation of models, known as ‘asymmetric’ were developed. These models allow positive and negative shocks of the same magnitude to have a different impact on conditional variance, which is more consistent with the observed behaviour of financial asset returns.[5]

2.3.2.1. EGARCH Model

The Exponential Generalised Autoregressive Conditional Heteroskedasticity (EGARCH) model was introduced by Nelson [6]. Unlike the standard GARCH model, the EGARCH takes the logarithm of the conditional variance, which has two significant advantages: it ensures that the variance itself is always positive without needing to impose non-negativity constraints on the parameters, and it allows for an asymmetric response function.

The mechanism of the common EGARCH(1,1) model is represented as:

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

Where:

- $log(\sigma_t^2)$ is the natural logarithm of the conditional variance.
- ω, β, α and γ are the model’s parameters.
- The term $\alpha \left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right|$ captures the symmetric effect of the magnitude of the shock
- The term $\gamma \frac{\epsilon_{t-1}}{\sigma_{t-1}}$ captures the asymmetric or leverage effect.

The model’s defining feature is the parameter γ. If a shock is negative (ϵ_{t-1}<0), the total effect of the shock is α - γ. If a shock is positive (ϵ_{t-1}>0), the total effect is α + γ. For the leverage effect to hold, the parameter γ is expected to be negative, ensuring that a negative shock has a larger impact on the log-variance than a positive shock of the same size. The EGARCH model’s main contribution is its ability to successfully capture both volatility clustering and the leverage effect.

2.3.2.2. GJR-GARCH Model

In 1993, an alternative asymmetric model was proposed by Glosten, Jagannathan, and Runkle, the GJR-GARCH [7]. Rather than modelling the logarithm of the variance, the GJR-GARCH model extends the standard

GARCH framework by incorporating an additional term that explicitly accounts for the leverage effect.
The specification for a GJR-GARCH(1, 1) model is:

$$\sigma_t^2 = \omega + (\alpha + \gamma I_{t-1})\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

Where the indicator function I_{t-1} is defined as:

$$I_{t-1} = \begin{cases} 1 \text{ if } \epsilon_{t-1} < 0 \text{ (bad news)} \\ 0 \text{ if } \epsilon_{t-1} \geq 0 \text{ (good news)} \end{cases}$$

And where:

- σ_t^2 is the conditional variance at time t.
- ω, α and β are the standard constant, ARCH and GARCH parameters, respectively.
- γ is the leverage parameter.
- ϵ_{t-1}^2 is the squared residual from the previous period.
- σ_{t-1}^2 is the conditional variance from the previous period.
- I_{t-1} is the indicator function for negative shocks.

This mechanism introduces asymmetry through the indicator function, which ‘activates’ the leverage parameter, γ, only in the presence of a negative shock. Therefore, a negative shock increases the conditional variance ($\alpha + \gamma$), while a positive shock of the same magnitude increases it only by α. For the leverage effect to be present, the parameter γ must be positive and statistically significant.

2.3.2.3. APARCH Model

A highly flexible and encompassing model within the GARCH family is the Asymmetric Power ARCH (APARCH) model, developed by Ding, Granger and Engle[8]. The APARCH model provides a significant generalisation by including other ARCH-type models in its equation, such as the standard GARCH and the GJR-GARCH, within its framework.

Its specification allows for both a flexible power term and a leverage effect. The APARCH(p,q) model is defined as:

$$\sigma_t^\delta = \omega + \sum_{i=1}^{q} \alpha_i \left(|\epsilon_{t-i}| - \gamma_i \epsilon_{t-i} \right)^\delta + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^\delta$$

Where:

- σ_t is the conditional standard deviation.
- ω, α_i, and β_j are the standard constant, ARCH and GARCH parameters.
- δ is a positive power term, which can be estimated from the data.
- γ_i represents the leverage parameter, capturing the asymmetric response to shocks.

The model's primary strength lies in this flexibility. The power term, δ, can be estimated from the data rather than being pre-specified by the researcher, allowing for a more adaptable model structure. Asymmetry is captured by the leverage coefficient, γ_i. The term $(|\epsilon_{t-i}| - \gamma_i \epsilon_{t-i})$ achieves this by taking the magnitude of the shock and then adjusting it based on its sign, effectively increasing the impact of negative shocks and decreasing the impact of positive ones. A positive value for γ_i indicates that the leverage effect is present. However, this high degree of flexibility is also a limitation. The APARCH model is heavily parameterised compared to simpler econometric models, which can lead to estimation difficulties and a higher likelihood of model non-convergence, particularly with smaller datasets.

2.3.3. Limitations of Parametric Econometric Models

While the GARCH family of models provides a sophisticated framework for capturing specific stylised facts, they are fundamentally parametric. Meaning they assume that the complex, volatility process of an asset can be accurately described by a fixed mathematical equation with a small number of parameters.

The primary limitation of this approach is its rigidity. Financial markets are widely regarded as highly complex, noisy and non-linear systems [9]. If the true data-generating process is more complex than the predefined structure of a model, the model will be misspecified, and its forecasting accuracy will be capped by its structural limitation. This has led researchers to explore non-parametric models, which do not have a rigid structure on the data.

2.4. Deep Learning Approaches

To mitigate the weaknesses of econometric models, researchers have developed non-parametric deep learning alternatives. Unlike the GARCH family, these models do not need a predefined mathematical structure for volatility. Instead, they are designed to learn complex and non-linear patterns directly from historical data.

2.4.1. Long Short-Term Memory Networks

A primary deep learning architecture for time-series analysis is the Recurrent Neural Network (RNN), which is designed to handle sequential data through an internal feedback loop. However, simple RNNs are susceptible to the Vanishing Gradient Problem, which severely limits their ability to learn and remember patterns over long sequences of data [10]. To overcome this limitation, the Long Short-Term Memory (LSTM) network was developed by Hochreiter and Schmidhuber (1997) as a specialised and more robust RNN architecture [11].

The key innovation of the LSTM is its memory cell, which can maintain and regulate information over extended periods. The flow of information within this cell is controlled by gates, as illustrated in the following diagram.

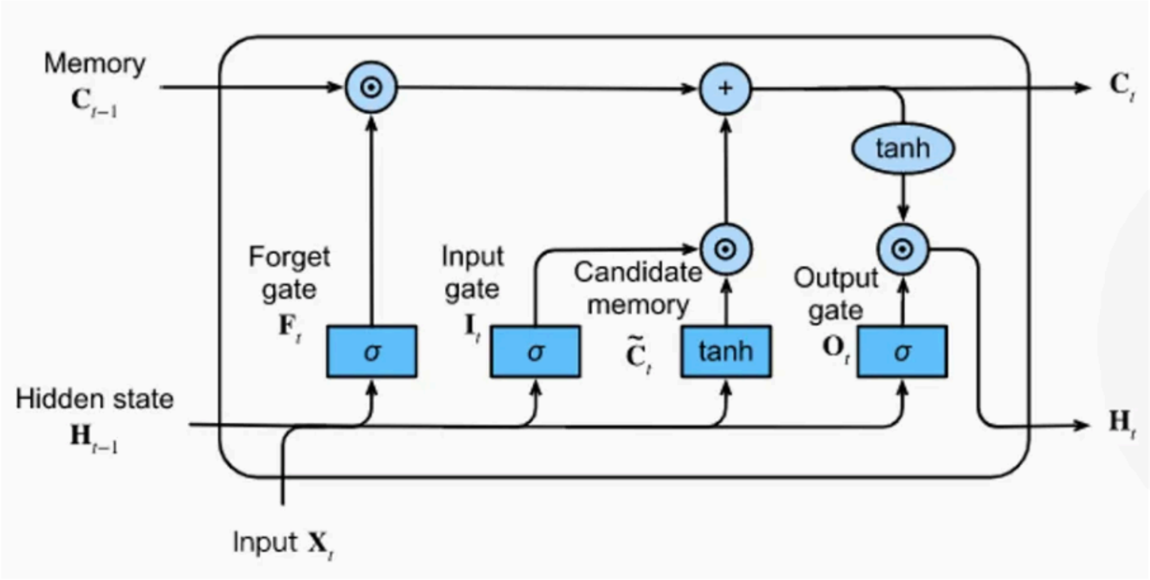

Figure 1: Architecture of an LSTM unit[12]

As shown in Figure 1, the LSTM cell's architecture allows it to manage its memory. The cell state (C_t) acts as the long-term memory, carrying information through the sequence. This memory is precisely controlled by three gates. The forget gate (F_t) examines the new input and previous state to decide which information is no longer relevant and should be discarded from the cell state. The input gate (I_t) then determines which new information is important enough to be added and stored. Finally, the output gate (O_t) takes the filtered cell state and decides which parts of the memory should be used to generate the output for the current time step (H_t).

This gating mechanism allows the LSTM to capture complicated, long-range patterns in the data. By learning which information to retain and which to discard, it can model patterns that unfold over long periods of time, making it a powerful tool for forecasting volatility.

2.4.2. The CBOE Volatility Index

The Chicago Board Options Exchange (CBOE) Volatility Index (VIX) is a real-time index that measures the market's expectation of 30-day forward looking volatility of the S&P 500 index. It is derived from the S&P 500 index options and is also known as the "fear index" as it reflects market sentiment and uncertainty.

The justification for using the VIX as a predictor is its forward-looking nature. Unlike GARCH-family models, which are purely backwards-looking by relying on the history of a price series, the VIX is derived from option prices that reflect the market's current, collective sentiment about future volatility. It combines various data sources, including traders' expectations and reactions to upcoming economic events.

This strength is well-supported by empirical evidence. The literature consistently demonstrates that the VIX contains significant predictive power for future realised volatility [13]. Based on this strong backing, the VIX is a prime candidate for an input to enhance the predictive accuracy of a non-parametric model.

2.4.3. Hybrid GARCH-LSTM Models

To leverage the distinct strengths of both econometric and machine learning paradigms, there has been research on hybrid models. The rationale for this approach is to create a system where a GARCH-family model first captures the well-defined linear structure and stylised facts of volatility, and a neural network then models the remaining complex, non-linear patterns.

The sequential error-fitting approach is a two-stage process. First, an econometric model is fitted to an asset's return series to generate a primary volatility forecast and a series of standardised residuals. Second, an LSTM network is trained on these residuals to learn and predict the errors that the GARCH model made. Kim and Won applied this method to the KOSPI 200 index, testing various GARCH family components [14]. Their findings show that the hybrid models consistently produced more accurate forecasts than either the standalone GARCH or LSTM models, demonstrating the value of using the LSTM to correct the GARCH model's weaknesses. Rosyk and Slepaczuk also follow this methodology and prove that the hybrid econometric-deep learning models produce better forecasts than its components.[15]

2.5. Literature Review Conclusion

This chapter has shown the evolution of volatility forecasting, from the parametric GARCH family models designed to capture specific stylised facts to the non-parametric deep learning approaches like LSTMs that offer greater flexibility. The review has highlighted the persistent limitations of symmetric models, the advances made by asymmetric specifications, and the critical role of forward-looking information, such as the Volatility Index. This project looks at a comparison between these models to identify the superior forecasting model on the S&P 500 and VIX.

3. Design and Implementation

3.1. Introduction

This chapter details the design and implementation of the experimental methodology used to systematically compare volatility forecasting models. The methodology is a comparative analysis designed to evaluate the accuracy of various econometric, deep learning and hybrid models on the S&P 500 index.

The chapter will first describe the data acquisition and preprocessing, introduce the models being tested, the experimental design and evaluation framework and conclude with a discussion of implementation details and software tools employed.

3.2. Data Acquisition and Preprocessing

3.2.1. Data Sources

The primary dataset for this study consists of the S&P 500 index (Ticker: ^GSPC) and the CBOE Volatility Index (Ticker: ^VIX). The data for both series was sourced from Yahoo Finance, using the daily 'Close' price from 2 December, 1999 to 27 July, 2025. This time frame was selected to train the models on a variety of market conditions, including the dot-com bubble, the 2008 financial crisis and the COVID-19 pandemic. The S&P 500 was chosen as it is the benchmark for the U.S. equity market and the VIX was chosen for its predictive power and forward-looking nature.

Figure 2 shows the historical plot of the S&P 500 while Figure 3 displays the historical plot of the VIX. The S&P 500 (Figure 2) clearly shows a strong upward trend, meaning that the series is non-stationary. This means that its log return will need to be calculated before being fitted to the econometric models.

Figure 2: Daily Close Prices of the S&P 500 Index (2000- 2025)

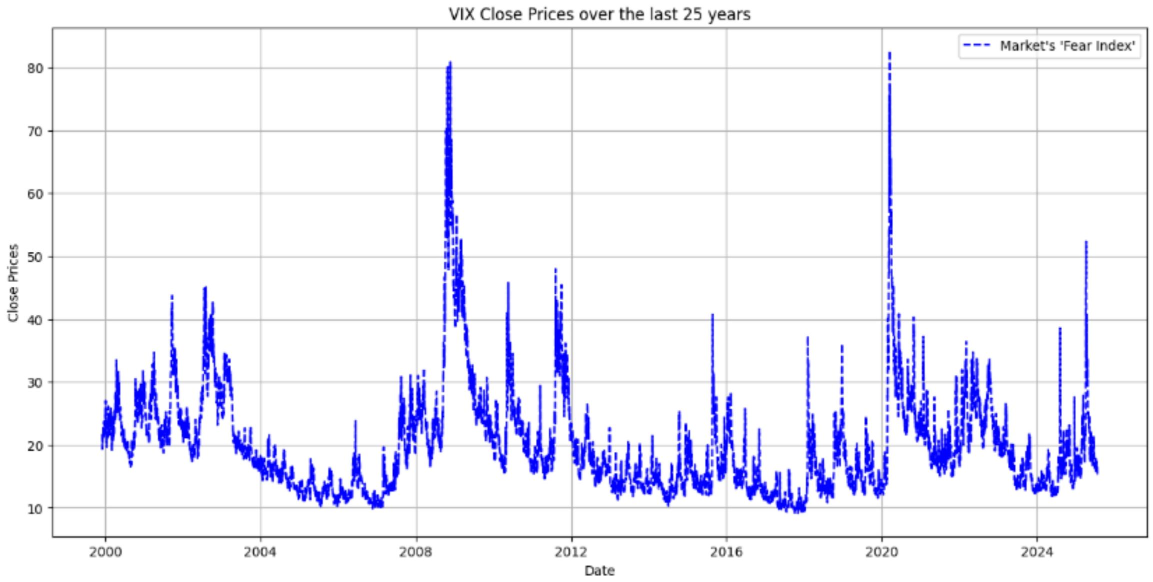

Figure 3: Daily Close Prices of the VIX (2000-2025)

3.2.2. Input Data Preprocessing

Log Return Calculation and Stationarity Testing

The raw S&P 500 'Close' price was first converted into a log return series, r_t, using the formula $r_t = log(P_t) - log(P_{t-1})$. Log returns are standard in financial analysis as traditional econometric models require stationary (constant mean and variance) time-series data to function.
The log return series (Figure 4) has a constant mean and clustered variance without a discernible trend, meaning it is stationary.

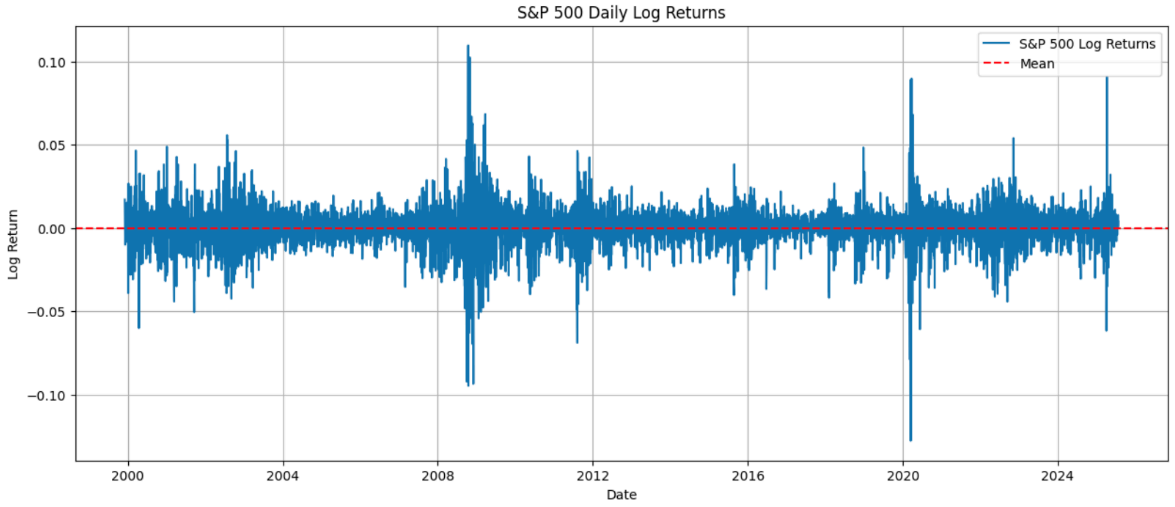

Figure 4: Daily Log Returns of the S&P 500 Index (2000-2025).

To confirm this, the Augmented Dickey-Fuller (ADF) test was performed. This yielded a Test Statistic of -19.28 and a p-value of < 0.01, allowing for the rejection of the null hypothesis of a unit root. This confirms that the log return series is stationary and suitable for the subsequent modelling stages.

Target Variable Engineering

The target variable for all models in this study is the 21-day future realised volatility. This was engineered from the log return series to serve as the "ground truth" for the forecasting task. The 21-day window was chosen to approximate one trading month. The realised volatility, σ_{RV}, for a given day, t, was calculated as the annualised standard deviation of the daily log returns over the next 21 trading days.
The formula used:

$$\sigma_{RV,t} = \sqrt{252} \times \sqrt{\frac{1}{20} \sum_{i=1}^{21} (r_{t+i} - \bar{r})^2}$$

Where:

- r_{t+i} is the log return on day $t + i$,
- $\bar{r}$ is the average log return over the 21-day future window,
- The $\sqrt{252}$ term annualises the volatility based on the approximate number of trading days in a year.

Final Dataset Alignment and Splitting

The processed S&P 500 log return series, the raw VIX 'Close' price series, and the engineered future realised volatility target variable were aligned by date into a single DataFrame. Rows containing missing values, which arose from the log return calculation (the first day) and the future realised volatility calculation (the final 21 trading days of the period), were removed to create a complete dataset.

This final dataset was then split chronologically into three distinct sets: a training set (70%), a validation set (15%), and a test set (15%). The specific dates are in Figure 5. A chronological split is essential for time-series data to ensure the model is trained on past data and is evaluated on future data, preventing any lookahead bias. The training set was used to train the models, the validation set was used for hyperparameter tuning and the intermediate model selection, and the test set was held out as completely unseen data for the final evaluation of the models' out-of-sample performance.

Dataset	Proportion	Date range
Training	70%	3/1/2000 - 10/27/2017
Validation	15%	30/10/2017 - 24/8/2021
Test	15%	25/8/2021 - 25/6/2025

Figure 5: Chronological Data Split

3.3. Model Specifications

This section provides a detailed technical specification for every model included in the project. The models are grouped into three categories: traditional econometric models, deep learning models, and hybrid models that combine both approaches. The final specification for each model was determined through a rigorous process of tuning and testing on the validation dataset, to identify the most promising model from each category to use for final testing.

3.3.1. Econometric Models

These models were selected to represent both symmetric and asymmetric econometric models. The specific models implemented and tested were:

- ARCH (Autoregressive Conditional Heteroskedasticity): The foundational model used to capture volatility clustering.
- GARCH (Generalised ARCH): The more parsimonious and widely-used extension of the ARCH model.
- EGARCH (Exponential GARCH): An Asymmetric model that captures the leverage effect by modelling the logarithm of the variance.
- GJR-GARCH (Glosten-Jagannathan-Runkle GARCH): An alternative asymmetric model that uses an indicator function to capture the leverage effect.

- APARCH (Asymmetric Power ARCH): A highly flexible generalisation that nests many of the other models within its framework.

To find the optimal (p,q) order for the econometric models, a preliminary test was conducted on the standard GARCH using the validation dataset. The selection was based on two criteria: the statistical significance of the estimated parameters (p-value < 0.05) and the model's information criteria scores. The primary information criteria metrics used were the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). These are statistical measures that balance a model's goodness of fit with its complexity, penalising the inclusion of additional parameters to prevent overfitting. A lower AIC and BIC score indicates a more parsimonious and effective model.

While preliminary tests (Appendix A) showed that the GARCH(2,2) specification yielded the lowest AIC and BIC values, the model was rejected as several of its key parameters were not statistically significant, indicating over-parameterisation (see Appendix A for the full comparison table). Therefore, the GARCH(1,1) specification, which produced the next lowest AIC/BIC scores while maintaining statistically significant parameters, was selected as the most robust and parsimonious choice for all models in this category. Every model was implemented in Python using the $arch$ library and were initially tested with both a standard Normal (Gaussian) and a Student's T distribution for the errors.

3.3.2. Standalone LSTM Models

Two standalone Long Short-Term Memory (LSTM) networks were developed to serve as the primary deep learning benchmarks. The first is a sole LSTM, which only uses the S&P 500 log return series as input. The second is a VIX-Enhanced LSTM, which uses both the VIX and the S&P 500 log return series as the inputs.

Final Model Architecture

To prepare the data for the LSTM, the input features were first scaled to a range of [0,1] using a MinMaxScaler. This standard preprocessing step ensures that all inputs are on a comparable scale, which improves the stability and performance of the model during training. The scaled data was then transformed into sequences of fixed length. A lookback window of 60 days was used, meaning each input sample provided to the model consisted of the 60 previous time steps of the relevant features.

Following a hyperparameter tuning process, a final, optimised architecture was selected for both standalone models. The architecture consists of a stacked LSTM with 2 layers, each containing 50 units. Dropout with a rate of 0.2 was applied after each LSTM layer for regularisation and to prevent overfitting. This was followed by a Dense layer of 25 units and a final Dense output layer with a single unit to produce the volatility forecast. The key parameters of the final architecture are summarised in Figure 6.

Hyperparameter	Value
Lookback Window	60
LSTM Layers	2
Units (per LSTM Layer)	50
Dense Layers	2
Dropout Rate	0.2
Optimiser	Adam
Loss Function	Mean Squared Error (MSE)
Epochs	25
Batch Size	32

Figure 6: Final Hyperparameters for LSTM Models

The model was implemented in Python using the Keras library with TensorFlow. The standalone LSTM's architecture is shown in Figure 7.

```
lstm_model = Sequential()
lstm_model.add(LSTM(units=50, return_sequences=True,
input_shape=(x_train.shape[1],1)))
lstm_model.add(Dropout(0.2))
lstm_model.add(LSTM(units=50, return_sequences=False))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(units=25))
lstm_model.add(Dense(units=1))
lstm_model.compile(optimizer='adam', loss='mean_squared_error')

history = lstm_model.fit(x_train, y_train, epochs=25, batch_size=32,
validation_data=(x_val, y_val), verbose=0)

lstm_predictions = lstm_model.predict(x_val)
rmse_lstm = np.sqrt(mean_squared_error(y_val, lstm_predictions))
```

Figure 7: Keras Implementation of LSTM Architecture

Hyperparameter Tuning Process

The final architecture was determined through a hyperparameter tuning process conducted on the validation set. The goal was to choose an optimal configuration that balanced accuracy with computational efficiency. Parameters such as the number of LSTM layers, units, batch size, epochs, and dropout rate were all tested.

An initial comparison between a one and two-layer LSTM network showed that the two-layer model performed better. Afterwards, a grid search was performed to find the optimal number of units, dropout rate, and batch size, with the full results presented in Appendix B. While the tuning results indicated that a configuration with 150 units produced a marginally lower RMSE, it was not selected as the final architecture. This was because performance gain was minimal but computational cost was significantly higher. Furthermore, when these parameters were applied to the hybrid model, they led to a degradation in performance. Therefore, the best performing 50-unit architecture was chosen as the final specification, as it provided the optimal balance of predictive accuracy, computational efficiency and robust performance across all model configurations.

The number of training epochs was set to 25. As shown in Figure 8, the training loss consistently decreased as the number of epochs increased, while the validation loss plateaued around 20-25 epochs, indicating that further training would not improve performance on unseen data and could risk overfitting.

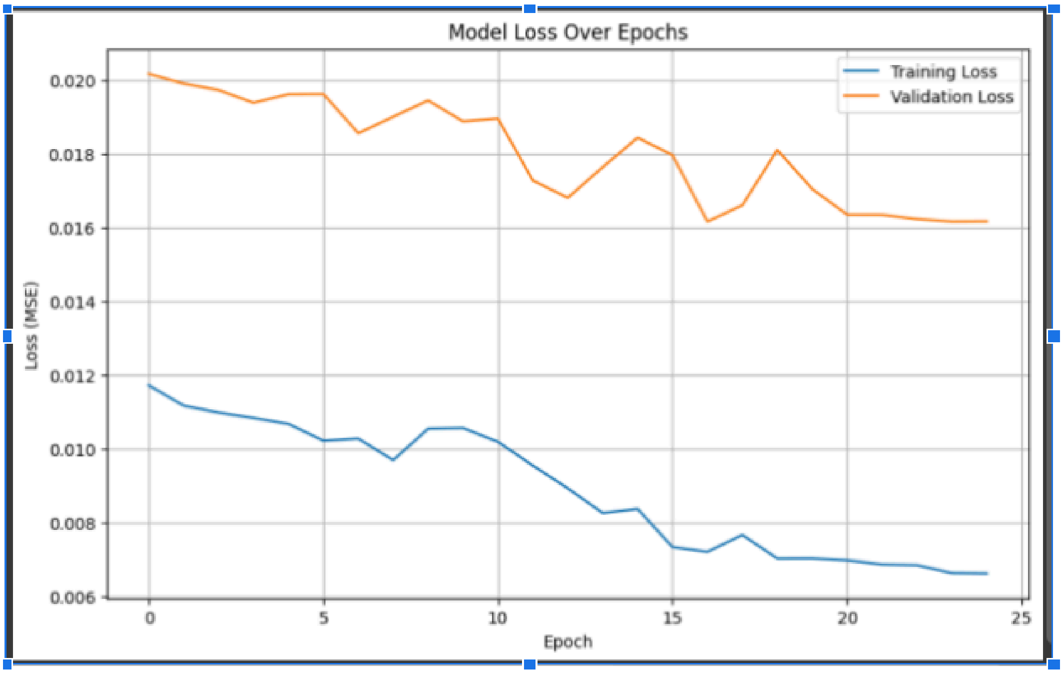

Figure 8: Model Training and Validation Loss (MSE) over 25 Epochs

3.3.3. Hybrid Model Specifications

To test if a combination of econometric and deep learning techniques could result in superior forecasts, two sequential hybrid models were developed. The core design is to use an APARCH model as a feature generator. The residuals from the APARCH model, which represent the information not captured by the parametric specification, are used as an additional input to an LSTM network.

The process for building these models involved two stages. First, the finalised APARCH(1,1) model was fitted to the log return series to generate a series of standardised residuals for the training, validation and test sets. Second, this residual series was used as a new feature for an LSTM network. Two configurations were constructed:

- APARCH-LSTM: A standard hybrid model where the LSTM component was trained on a two-feature input sequence: the original S&P 500 log returns and the generated APARCH residuals.
- APARCH-VIX-LSTM: A hybrid model that used a three-feature input sequence: the original S&P 500 log returns, the generated APARCH residuals, and the daily VIX ‘Close’ price.

In both configurations, the LSTM was trained to directly predict the 21-day future realised volatility. The LSTM component for both hybrids utilised the same optimised architecture and hyperparameters detailed in Section 3.3.2. The architecture of the APARCH-VIX-LSTM model is illustrated in Figure 9.

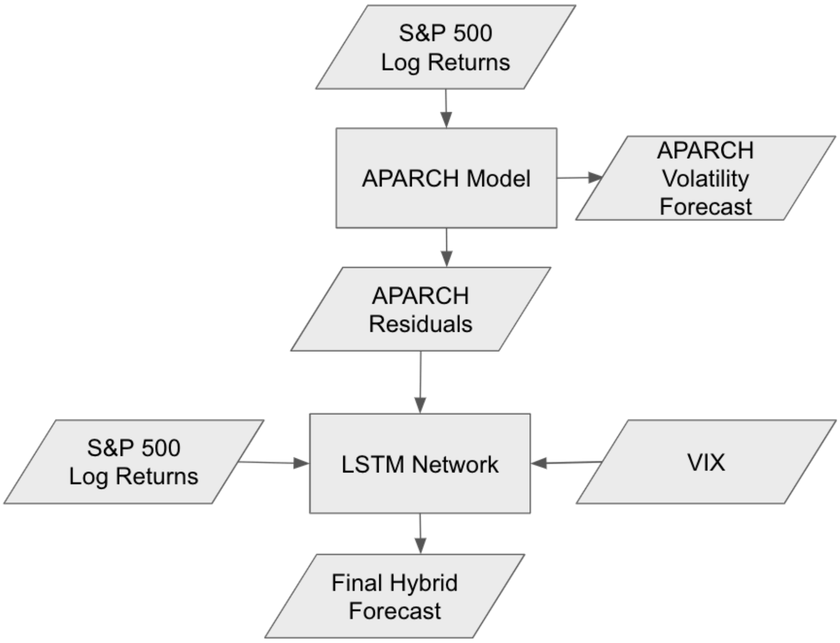

Figure 9: Architecture of the Sequential APARCH-VIX-LSTM Model. APARCH-LSTM follows the same structure without the VIX input.

The code snippet in Figure 10 shows the training data preparation for the three inputs required by the APARCH-VIX-LSTM model.

```
hybrid_vix_features_train = train_data[['sp500 log rtns', 'close
vix']].copy()
hybrid_vix_features_train['residuals'] = aparch_residuals
scaler_vix_hybrid = MinMaxScaler(feature_range=(0,1))
hybrid_vix_features_train_scaled =
scaler_vix_hybrid.fit_transform(hybrid_vix_features_train)
train_target = train_data['future volatility'].values
x_train_vix_hy, y_train_vix_hy = [],[]
for i in range(sequence_length,
len(hybrid_vix_features_train_scaled)):

x_train_vix_hy.append(hybrid_vix_features_train_scaled[i-sequence_len
gth:i, :])
  y_train_vix_hy.append(train_target[i])
x_train_vix_hy, y_train_vix_hy = np.array(x_train_vix_hy),
np.array(y_train_vix_hy)
```

Figure 10: Training Data preparation for the APARCH-VIX-LSTM Hybrid Model

3.4. Experimental Design and Evaluation Framework

This section details the complete process designed to rigorously and fairly compare all specified models. It outlines the forecasting methodologies employed for the different model classes, the primary performance metric used for evaluation, and the statistical test for determining the significance of performance differences. Furthermore, it describes the methods used for final model interpretation and the robustness checks conducted to validate the stability of the results.

3.4.1. Forecasting Methodology

For the initial model selection and hyperparameter tuning on the validation set, a static forecast was employed for all models. In this approach, each model is trained once on the training data and then used to generate predictions for the entire validation period. This was chosen for its computational efficiency, which allowed for quicker iteration.

The initial plan for the final, more rigorous evaluation on the unseen test set was to use dynamic methods that better simulate a real-world application. This involved using a rolling forecast for the econometric models, where the model is re-trained at each time step on a fixed-size window of the most recent past data to make a one-step-ahead prediction. For the deep learning models, a walk-forward forecast was planned, which is similar but uses an expanding window that incorporates all available data at each new step.

However, during the implementation phase, a significant challenge arose. While the baseline ARCH model was successfully evaluated using the rolling forecast, the APARCH model consistently failed to converge under the rolling forecast. This was a problem, as comparing models evaluated with different techniques would not be a fair comparison. Concurrently, tests on the VIX-LSTM and APARCH-VIX-LSTM models showed that the static and walk-forward forecast scored essentially the same, as shown in Figure 11.

Model	Static Forecast RMSE	Walk-Forward Forecast RMSE	Favoured Method
VIX-LSTM	0.06603	0.06603	Static (negligible difference, more efficient)
APARCH-VIX-LSTM	0.07122	0.07370	Static (Better performance)

Figure 11: Comparison of Static and Walk-Forward Forecasting on VIX-LSTM and APARCH-VIX-LSTM

Therefore, to ensure a fair and directly comparable evaluation across all models, a final decision was made to adopt a static forecast methodology for the final comparison on the test set.

3.4.2. Performance Metric

Root Mean Squared Error (RMSE) was the performance metric used to evaluate and compare all models in this study. It is a standard metric for regression and forecasting tasks that measures the quadratic mean of the errors between the predicted values and the actual values. The RMSE is calculated using the following formula:

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

Where:

- n is the number of data points in the data set.
- $\hat{y}_i$ is the predicted volatility for a given day.
- y_i is the actual realised volatility for that day.

RMSE was selected due to two key properties that are particularly relevant for financial forecasting. First, since the error terms $(\hat{y}_i - y_i)$ are squared, the metric places a disproportionately high penalty on large forecast errors. In the context of financial volatility, where such large errors are particularly undesirable, this is a useful characteristic. Second, the final square root transforms the metric back into the original units of volatility, making the model's performance directly interpretable. A lower RMSE value signifies a smaller average error between the forecasted and actual volatility, indicating a more accurate model.

3.4.3. Statistical Significance Testing

To assess whether the observed differences in forecast accuracy between models are statistically meaningful, the Diebold-Mariano (DM) test was used. The test is based on a null hypothesis (H_0) that both models have the same forecast accuracy. The test statistic is calculated based on the loss differential between the two models' forecast errors. This yields a p-value, which is used to make a decision. If the p-value is below a chosen significance level (α = 0.05), the null hypothesis is rejected. Rejecting the null hypothesis provides strong statistical evidence that the performance of the two models is not the same and that the observed difference in their RMSE scores is significant.

3.4.4. Model Interpretation using SHAP

A significant limitation of all Artificial Intelligence models such as LSTMs is their "black box" nature, where the decision making process is unknown. To overcome this and gain insight into the model's behaviour, an Explainable AI (XAI) technique was used. This study uses SHAP (SHapley Additive exPlanations), to interpret a model's decision making by analysing the contribution of each feature to the final output [16].

A SHAP value for a feature represents its impact on moving the prediction from a baseline (the average prediction) to its final value, with positive values increasing it. SHAP analysis was applied to the final winning VIX-LSTM model to interpret its individual forecasts on the test set. The objective is to identify which features are the most influential drivers of the model's predictions in different circumstances.

3.4.5. Model Robustness Checks

To ensure the reliability and stability of the models, two specific robustness checks were performed. These checks were designed to verify that the models' performance was not influenced by the stochastic (random) elements of their estimation processes, therefore ensuring the replicable and trusted results.

LSTM Performance Stability

Deep Learning models such as the LSTMs have a stochastic nature, primarily due to the random initialisation of their weights before training. To control this randomness for replicable results, a random seed can be set. This is an integer that acts as a starting point for the sequence of random numbers generated during the training process. To ensure the findings were not dependent on luck, the standalone LSTM and the VIX-LSTM were each trained and evaluated five times using different random seeds. The distribution of the resulting RMSE scores on the validation set is presented in Figure 12.

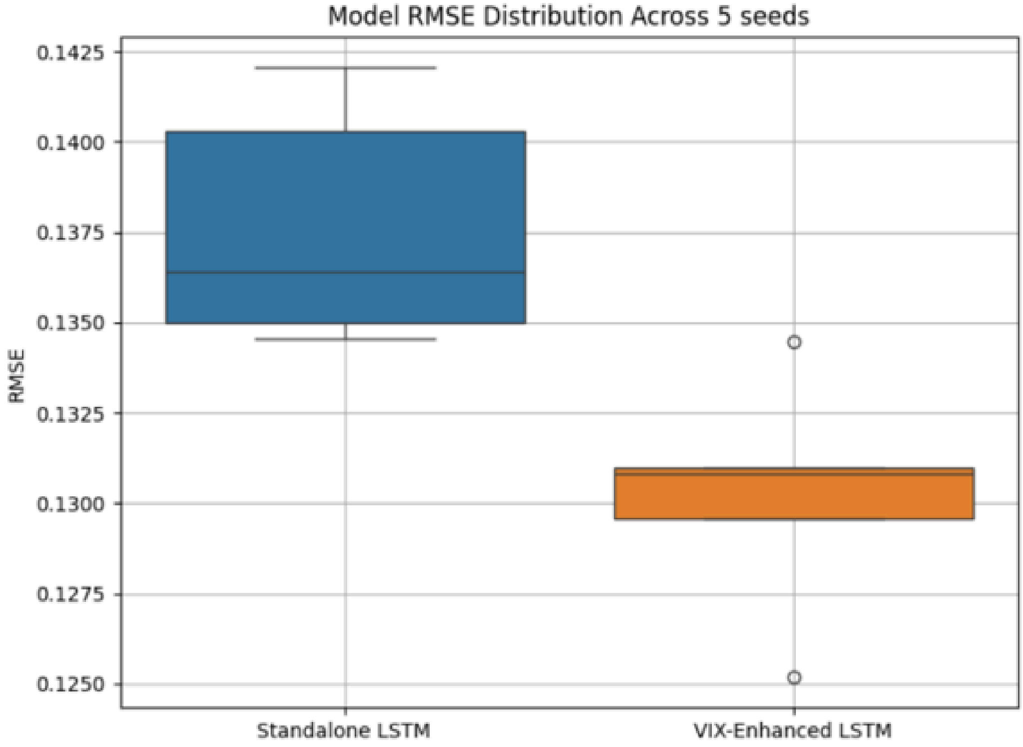

Figure 12: Standalone LSTM's and VIX-LSTM's distribution of RMSE scores across 5 random seeds

The results show that while there is variance in performance between runs, the VIX-LSTM is robustly and consistently superior to the standalone LSTM. The standard deviation for both models is also small, meaning that different seeds do not cause dramatically different results. This confirms that the VIX-LSTM is consistently better than the standalone LSTM and that different random seeds do not extremely affect results.

The box plot (Figure 12) reveals two key findings. First, the standard deviation for both models was small (0.00299), indicating that performance was stable and not drastically affected by the random seed. Second, and more importantly, the performance distributions are clearly separated, meaning that VIX-LSTM consistently outperformed the standalone LSTM, achieving a lower average RMSE (0.13020 vs. 0.13766). This confirms that VIX-LSTM is definitely better than the standalone LSTM. To ensure replicable results, a single random seed was chosen for the entirety of the project.

APARCH Replicability

During initial testing, the APARCH model displayed minor variations in its output on runs. To investigate this instability, the model was run 20 times on the validation set. As shown in the histogram in Figure 13, the resulting RMSE scores were tightly clustered with an average RMSE of 0.201020 and a small standard deviation of 0.001016, confirming the model was largely stable but not perfectly replicable.

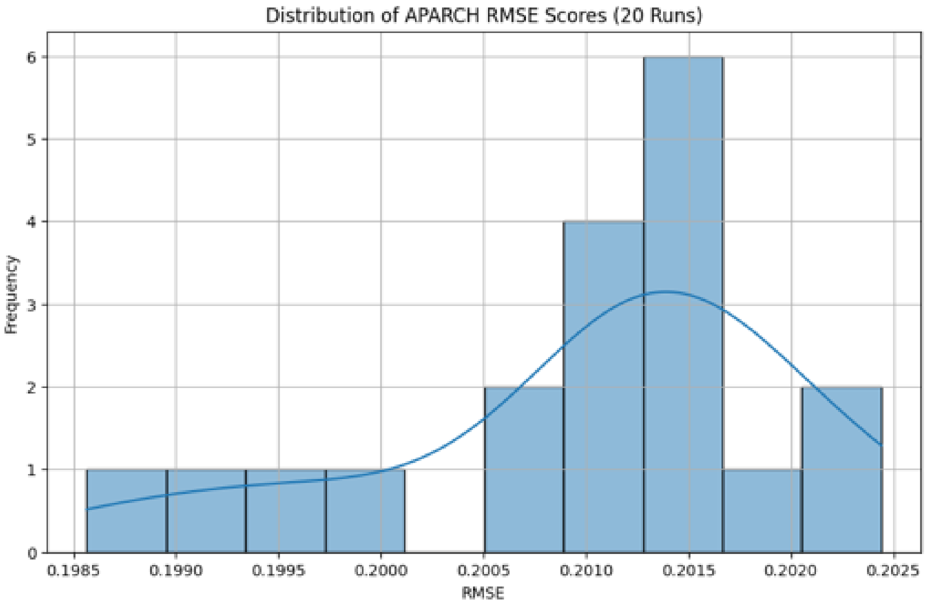

Figure 13: Distribution of APARCH RMSE scores across 20 unseeded runs.

The source of this variation was identified as a stochastic element within the model's forecasting function. By setting a RandomState for this component, it was possible to achieve fully deterministic and replicable results. Therefore, for all final reported results, a fixed RandomState was used during the APARCH model's forecasting process to ensure the findings are fully replicable.

3.5. Implementation Details

This section covers the project's details. It discusses the key challenges, design iterations, and problem-solving that led to the final experimental setup, and concludes by listing the specific software environment used.

3.5.1. 'Dead Ends' and Design Iterations

The final methodology described in this chapter is the result of an iterative development process. Many challenges were encountered and pragmatic decisions were made to ensure the final experimental design was both robust and feasible. The most significant of these iterations are summarised below:

- Forecasting Methodology: The initial plan for the final evaluation was to use a rolling forecast for the econometric models. However, since the APARCH consistently failed to converge, the static forecast was used. To ensure a fair and consistent comparison across all competitors, a static forecast was adopted for all models, as detailed in Section 3.4.1.
- Econometric Model Order: During the selection of the GARCH(p, q) order, the GARCH(2,2) had the lowest AIC/BIC scores. It was rejected due to its parameters being statistically insignificant, a possible sign of over-parameterisation. The more robust GARCH(1,1) specification was therefore chosen for all econometric models (see Section 3.3.1).
- LSTM Architecture Selection: The hyperparameter tuning concluded that the 150-unit LSTM was slightly better than the 50-unit model. The 50-unit architecture was ultimately selected as it was a better trade-off between performance and computational efficiency (being three times faster to train).
- Hybrid Model Performance: A key finding from the validation phase was that the hybrid models, which had access to more data, did not have better performance compared to VIX-LSTM. This discovery indicates that the quality of the input features is more important than the model's complexity.

- SHAP Analysis Scope: During the final analysis phase, there were issues with implementing the SHAP library, which limited the model interpretation to a single day's forecast.
- GARCH Model Convergence Failure: The standard GARCH(1,1) model, while stable on the validation set, failed to converge on the unseen test data, therefore, it was replaced by the ARCH to be the final stage's benchmark model. The original plan was to have the GARCH model as the final benchmark to replicate the current industry standard.

3.5.2. Software and Libraries

The entire project was developed in Python on Google Colab. The core libraries used were:

- yfinance: To download historical market data directly from Yahoo Finance.
- Pandas and NumPy: For data pipeline, data manipulation, cleaning and numerical operations.
- Matplotlib and Seaborn: To generate all plots, charts, and figures.
- Statsmodels: To perform the Augmented Dickey-Fuller (ADF) test for stationarity.
- Arch: Used for the implementation, fitting and forecasting of the econometric models.
- Scikit-learn: Used for data scaling (MinMaxScaler) and calculating the final Root Mean Squared Error (mean_squared_error).
- TensorFlow (with Keras API): Used to build, train and evaluate the LSTM models.
- Dieboldmariano: Used to conduct the Diebold-Mariano test for statistical significance.
- SHAP: Used to perform the Explainable AI analysis on the final model.

4. Experimental Results

4.1. Introduction

This chapter presents the results of the comparative analysis of the volatility forecast models. These results are directly from the methodologies discussed in Chapter 3. The results are presented sequentially, following the comparative analysis process. First, the performance of all models on the validation set is reviewed to select the finalist for each category. This is followed by the main evaluation of these finalists on the unseen test set, where the definitive winning model is identified. Then, the results of the Diebold-Mariano tests to confirm the statistical significance of the findings are presented. Finally, an interpretability analysis of the winning model is conducted using SHAP to provide insight into its predictive behaviour.

4.2. Econometric Model Comparison on Validation Set

The first stage of the experiment was to conduct a comparison of the econometric models to find the best-performing one. Each of the five econometric models was fitted and evaluated on the validation set. The models were ARCH(1), GARCH(1,1), EGARCH(1,1), GJR-GARCH(1,1) and APARCH(1,1). The performance of each model, measured by RMSE, is presented in Figure 14. A lower RMSE score indicates a more accurate forecast.

Model	RMSE
ARCH(1)	0.206818
GARCH(1,1)	0.201838
EGARCH(1,1)	0.207694
GJR-GARCH(1,1)	0.982810
APARCH(1,1)	0.162922

Figure 14: Econometric Models Performance on Validation Set

The results in Figure 14 show a wide range of performance across the models. The APARCH(1,1) model was the definitive winner, achieving an RMSE of 0.162922. This is a 19.3% improvement in forecast accuracy over the next best performing mode, GARCH(1,1). Notably, the results also show that the simpler symmetric models, ARCH and GARCH, both outperformed the asymmetric EGARCH and GJR-GARCH models. Moreover, the GJR-GARCH model performed exceptionally poorly with an RMSE of 0.982810, indicating potential instability in its specification for this dataset.

The APARCH model was selected as the top-performing econometric model. This means it was chosen as a model to be tested on the unseen test data and also the econometric model to be part of the hybrid system.

4.3. Deep Learning and Hybrid Model Comparison on Validation Set

Following the econometric evaluation, deep learning and hybrid models were tested on the validation data. The models tested were: the Standalone LSTM, VIX-LSTM, APARCH-LSTM and APARCH-VIX-LSTM. The RMSE scores for the four models are presented in Figure 15.

Model Category	Model	RMSE
Deep Learning	Standalone LSTM	0.130645
	VIX-LSTM	0.121921
Hybrid	APARCH-LSTM	0.139997
	APARCH-VIX-LSTM	0.127511

Figure 15: LSTM and Hybrid Models Performance on Validation Set

The results from this comparative analysis reveal several key findings. The VIX-LSTM was the clear winner, achieving the lowest RMSE of 0.121921. The power of the VIX as an input feature is evident: its inclusion improved the Standalone LSTM's accuracy by 6.7% and the APARCH-LSTM's accuracy by 8.9%.

Surprisingly, there's a negative effect when fitting the APARCH's residuals to LSTM networks. Adding the APARCH residuals as a feature worsened the performance of both Standalone LSTM and the VIX-LSTM by 7.2% and 4.6% respectively. This leads to the most significant finding from the validation phase: the simpler VIX-LSTM was 4.4% more accurate than the most complex hybrid model, the APARCH-VIX-LSTM.

4.4. Final Evaluation on the Test Set

The final stage of the comparative analysis was the evaluation of the selected finalist models on the held-out test set. This out-of-sample test provides the definitive assessment of each model's real-world forecasting performance. The finalists from each category were the ARCH model (as a baseline), the APARCH model (best of the econometric models), the VIX-LSTM model (best of the deep learning models) and the APARCH-VIX-LSTM model (best of the hybrid models). The final RMSE scores for each model are presented in Figure 16.

Model	Test Set RMSE	% Improvement vs ARCH
ARCH(1)*	0.1715	-
APARCH(1,1)*	0.1634	4.75%
VIX-LSTM	0.0660	61.50%
APARCH-VIX-LSTM	0.0773	54.94%

*Figure 16: Finalist Models Performance on Test Set *Plots not available as static forecasts fails to plot econometric models.*

The results from the test set confirm the findings from the validation phase, establishing the VIX-LSTM as the definitive winning model with an RMSE of 0.0660. This represents a substantial 61.5% improvement in forecast accuracy over the baseline ARCH model and a 59.6% improvement over the best econometric model, the APARCH. Consistent with the validation set results, the more complex APARCH-VIX-LSTM hybrid was again outperformed by the simpler VIX-LSTM, which was 14.6% more accurate.

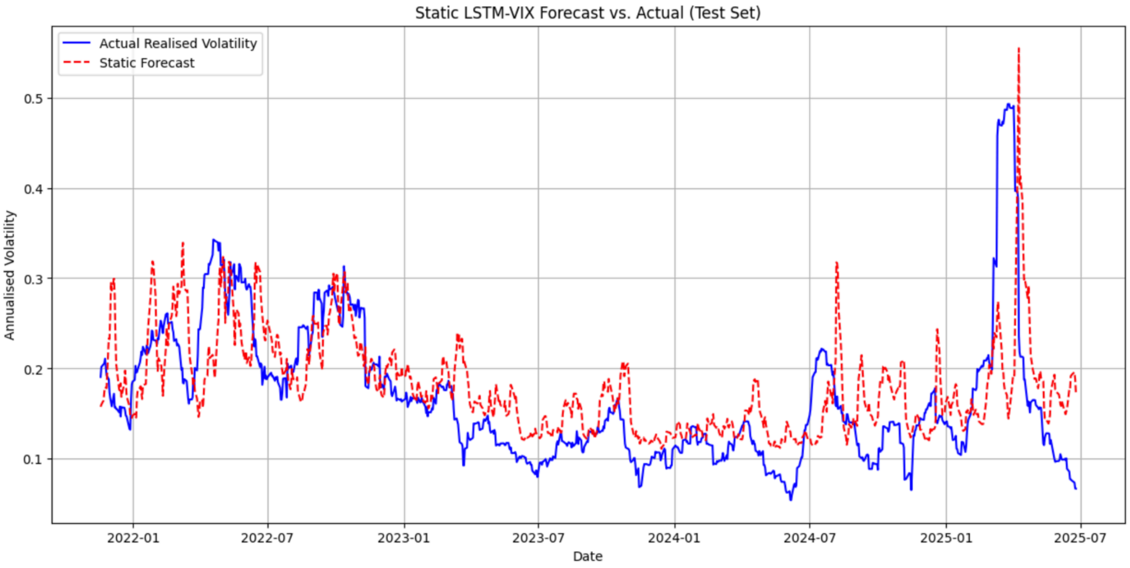

Figure 17: VIX-LSTM Forecast vs Actual Realised Volatility on the Test Set.

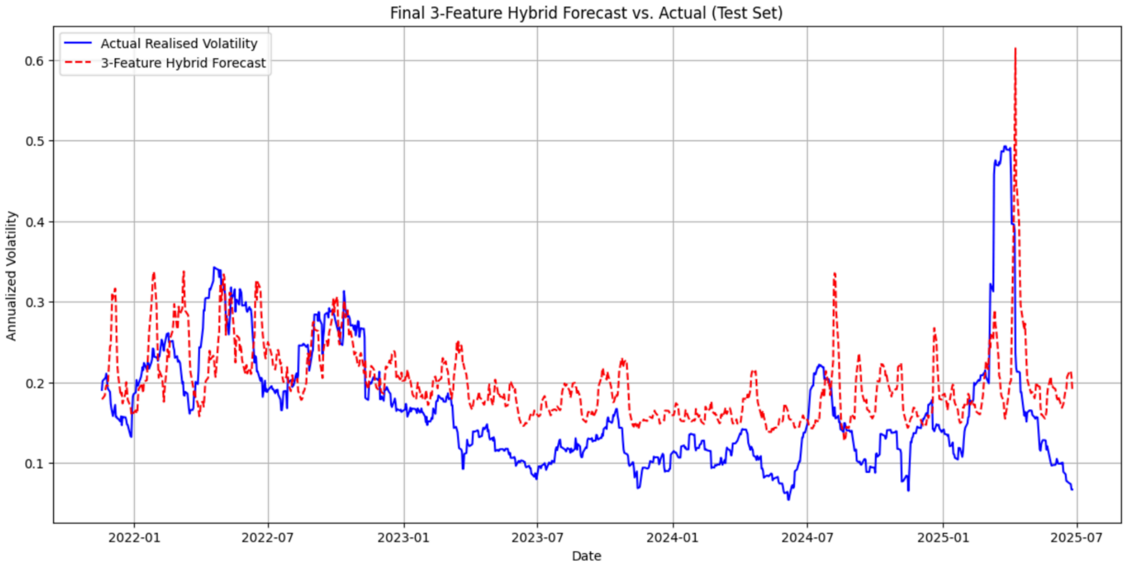

Figure 18: APARCH-VIX-LSTM Forecast vs Actual Realised Volatility on the Test Set.

The two best performing models were plotted against the actual volatility on the test set in Figure 17 and Figure 18. While both models track the major spikes in volatility, there's a noticeable difference in calmer periods. The VIX-LSTM forecast (Figure 17) appears to track the actual volatility more closely, whereas the APARCH-VIX-LSTM forecast (Figure 18) tends to over-predict volatility, remaining elevated even when actual volatility is low. This bias contributes to its higher overall RMSE score.

4.5. Statistical Significance of Results

To formally validate the findings from the test set, the Diebold-Mariano (DM) test was used. The test was conducted by comparing the winning model against all other finalists. The null hypothesis (H_0) of the test was that the two models being compared have equal forecast accuracy. The results are presented in Figure 19.

Comparison	DM Statistic	p-value	Conclusion (at 5% level)
VIX-LSTM vs ARCH	28.020	< 0.001	Reject H_0
VIX-LSTM vs APARCH	25.903	< 0.001	Reject H_0
VIX-LSTM vs APARCH-VIX-LSTM	10.355	< 0.001	Reject H_0

Figure 19: Diebold-Mariano Test Results comparing VIX-LSTM to other models.

The results of the Diebold-Mariano tests provide strong statistical evidence for the superiority of the VIX-LSTM. As shown in Figure 19, the p-value for every comparison was below the 0.05 significance level, rejecting H_0 in all cases. This confirms that the VIX-LSTM's lower RMSE is not a random occurrence, but a statistically significant improvement in forecast accuracy over the ARCH, APARCH and APARCH-VIX-LSTM models.

4.6. Interpretation of the Winning Model (SHAP Analysis)

To understand the 'black box' nature of the winning VIX-LSTM, a SHAP analysis was performed. This shows how each input feature contributes to the final prediction. The analysis for a single, representative day from the test set is shown in Figure 20.

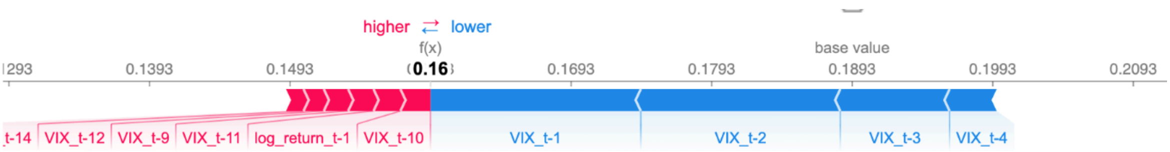

Figure 20: SHAP Force Plot for the VIX-LSTM Prediction on 25/08/2021. Red features push the prediction higher, blue features push it lower.

The force plot provides insights into the model's predictive behaviour. The main finding is the overwhelming importance of the VIX feature. The plot shows that the prediction for this day was almost entirely driven by the lagged values of the VIX, with the single log return feature (log_return_t-1) having only a minor impact in comparison.

Furthermore, the analysis reveals a distinct temporal pattern in how the model uses the VIX. The most recent VIX values (from t-1 to t-4) all have negative SHAP values, indicating that they pushed the volatility forecast lower than the average. In contrast, older VIX values (from t-9 to t-14) have positive SHAP values, pushing the forecast higher. This demonstrates that the VIX-LSTM has learned a complex, non-linear relationship where the VIX's impact changes over the 60-day lookback period.

4.7. Chapter Summary

This chapter presented the results of the comparative analysis. The initial test on the validation set identified the APARCH, VIX-LSTM and APARCH-VIX-LSTM as the strongest candidates from their respective categories. The final evaluation on the unseen test data confirmed the VIX-LSTM as the definitive winning model, demonstrating a 61.5% improvement in forecast accuracy over the baseline ARCH model. This result was verified to be statistically significant by the Diebold-Mariano test. Finally, a SHAP analysis provided insight into the winning model's behaviour, revealing that its predictions were overwhelmingly driven by the VIX data. The following chapter will discuss the implications of these findings in greater detail.

5. Discussion

5.1. Introduction

The previous chapter presented the results of the comparative analysis, identifying the VIX-LSTM as the superior forecasting model with a statistically significant 61.5% improvement in accuracy over the baseline ARCH model. This chapter aims to interpret results. The purpose is to explore the potential reasons behind the key findings, connect them to the theories discussed in the literature review, and consider their implications. The discussion begins by analysing the performance of the key models, then considers the practical and academic implications of the results, and concludes by acknowledging the limitations of this study.

5.2. Interpretation of Key Findings

The Power of VIX and Deep Learning

The VIX-LSTM was the clear winner due to the forward-looking VIX and the LSTM's flexible, non-parametric learning ability. As econometric models are backward-looking, they are constrained to information solely from the history of the price series itself. While effective at capturing certain stylised facts, their predictive power has a ceiling based on the input data's quality.

In contrast, the VIX provides forward-looking data, which includes the market's real-time expectation of future volatility. It represents a much richer predictive signal, evidenced by the SHAP analysis. The LSTM, unlike the econometric models, is capable of learning complex, non-linear relationships between this feature and volatility. The experimental results strongly suggest that providing a superior model architecture (LSTM) with a superior data source (VIX) leads to improved forecasting accuracy.
This conclusion is further substantiated by the SHAP analysis. The VIX-LSTM's force plot clearly shows that the various lagged VIX values were the dominant features, while the historical log return had a little impact in comparison. This proves that the model learned to prioritise the forward-looking information from the VIX over the backward-looking price series to generate its better forecasts.

The Danger of Architectural Complexity

One of the most significant findings of this study is the underperformance of the complex hybrid models compared to their simpler versions. Tests on the validation data showed that fitting APARCH residuals as an input feature worsened the LSTM's forecast accuracy by up to 7.2%.

A likely explanation for this counterintuitive result is that the APARCH residuals, rather than providing clear data, may have introduced statistical noise. While econometric models capture some structure, their residuals still contain randomness. By forcing the LSTM to learn from this noisy feature in addition to the cleaner signals from the log returns and the VIX, its predictive power has been compromised.

This confirms that the quality of the input features was more critical than the architectural complexity of the model. The clean, forward-looking data from the VIX proved to be a far more valuable input for the LSTM than the noisy, backward-looking residual information from the APARCH model.

The Fragility of Parametric Models

A consistent theme throughout the experimental results was the fragility of the econometric models. This was shown by the exceptionally poor performance of the GJR-GARCH model on the validation set, the underperformance of the EGARCH model compared to symmetric econometric models, and, most importantly, the convergence failures of the GARCH and APARCH models when applied to the final, unseen test data.

These issues are evidence of parametric models' limitations. Since GARCH-family models are parametric, their performance depends on the data conforming to their strict mathematical assumptions. The convergence failures on the test set suggest that the statistical properties of the data in this out-of-sample period were sufficiently different from the training period to cause a breakdown. In contrast, the non-parametric LSTM models are more robust, successfully generating forecasts under all conditions. This suggests that for practical applications, the adaptability of a data-driven deep learning approach may be more reliable than the theoretical and unreliable parametric model.

The Impact of Data Volume

A final key observation from the results is the significant performance improvement of most models on the test set compared to their performance on the validation set. The winning VIX-LSTM, for instance, saw a 45.9% improvement on RMSE from the validation set to the test set.

This improvement could be due to the final models being trained on the combined training and validation data prior to the final evaluation. The larger dataset provided a wider variety of market conditions for the models to learn from, improving their ability to generalise to the unseen test data. This finding shows the role that training data volume plays in developing robust forecasting systems.

5.3. Implications of Findings

Implications for Practitioners

For practitioners such as risk managers and quantitative traders, the superior accuracy of the VIX-LSTM, 61.5% more accurate than the ARCH model, leads to more accurate volatility forecasts, more precise calculations of key risk metrics like Value at Risk (VaR) and more effective capital allocation. In derivative pricing, where the price of an option is heavily dependent on expected volatility, a superior forecast can lead to more accurate pricing and robust hedging strategies. Moreover, the fragility of GARCH-family models on unseen data should be a deterrent. Practitioners should be cautious when deploying these models, as their performance may not be stable across different market periods.

Implications for Academic Researchers

For academic researchers, the results provide strong evidence to guide future work in volatility forecasting. The main finding of the VIX-LSTM outperforming a more complex hybrid model, suggests that future research should prioritise feature engineering over increasing architectural complexity. This study exposes the limitations of hybrid models, demonstrating that simpler forecasting methods can be a more effective benchmark. Finally, the research reinforces the value of LSTMs as a robust and effective tool for financial time series, particularly highlighting their stability in comparison to the fragility of some GARCH-family models.

5.4. Limitations of the Study

While this study provides valuable insights, there have been limitations in explaining the findings and guiding future research.

A primary limitation is that the analysis was conducted solely on the S&P 500 index. This asset is known to closely follow market stylised facts. Furthermore, the winning VIX-LSTM model's success is greatly dependent on the availability of the VIX. As many other assets do not have a linked volatility index, the applicability of this study's most successful model is constrained to specific markets.

The study's scope was also bound by specific methodological choices. The forecast horizon was fixed at 21 days, and the relative performance of the models could differ at other horizons. While the LSTM proved effective, the comparative analysis did not include other advanced deep learning architectures, such as Transformers, which could yield different results.

A real-world implementation would need to account for things such as transaction costs, which were not considered in this study and would impact the profitability of any trading strategy.

Finally, the interpretability analysis using SHAP was based on a single day's forecast. A more extensive application of SHAP across different market conditions would be required to form a complete picture of the VIX-LSTM's decision-making process.

5.5. Chapter Summary

In summary, this chapter interpreted the experimental results, concluding that the superior performance of the VIX-LSTM is a result of combining a powerful, forward-looking feature with a powerful deep learning model. The findings suggest that the quality of input features is more important than architectural complexity, a conclusion with significant implications in risk management and for future academic research.

6. Project Ethics

I confirm that I have read the ethical guidelines and have followed them during this project. I have used open-source data from Yahoo Finance, which is not derived from humans or animals. There also has not been any use of human participants in any activities for this project.

Data Category: A

The data used in this research (S&P 500 index and CBOE VIX) is public, aggregated and anonymous. It was sourced from Yahoo Finance

Participant Category: 0

The project did not use any human participants at any stage of the process.

6.1. Data Source and Management

To adhere to the Ethical Guidelines, I made sure that:

- The data chosen was completely anonymous.
- All the data was obtained from publicly accessible and reputable financial databases.
- A fixed date range was used for the dataset. This makes sure that the findings presented are fully reproducible by others.

6.2. Ethical Considerations

Although I do not use personal data, there are some possible ethical implications with my project.

- Volatility forecasting models can be misused and could contribute to market instability or personal loss of capital. My project doesn’t aim to be a live trading tool. My project’s sole purpose is to investigate different forecasting models and propose a hybrid econometric-deep learning model.
- Artificial Intelligence’s ‘black box’ nature means humans often struggle to understand how AI models make decisions. In this project, I attempt to answer this question by applying SHAP (Shapley Additive Explanations), an Explainer AI tool, to my best

performing model. This helped understand the rationale behind an AI model's decisions.

6.3. Conclusion

To conclude, this project fully adheres to the university's guidelines, and it is a category A0 project.

7. Conclusion and Future Work

7.1. Project Summary

This project looked to improve the forecast of daily realised volatility for the S&P 500 index. To achieve this, a comparative analysis was conducted, where different models were developed and evaluated. In the comparison there were traditional econometric models from the GARCH family, Deep Learning models based on the LSTM architecture, and econometric-deep learning hybrid models. This study identified a VIX-enhanced LSTM as the superior model, achieving a statistically significant 61.5% improvement in forecast accuracy over the baseline ARCH model on the unseen test data.

7.2. Restatement of Contribution

The main takeaway of this dissertation is finding the VIX-LSTM model as the best for forecasting daily S&P 500 realised volatility, demonstrating a statistically significant and substantial improvement in accuracy over traditional econometric models and complex econometric-deep learning hybrid models.

The findings contribute to the literature by sharing strong evidence that market volatility forecasts feature quality is more important than architectural complexity. The study also touches on the fragility of parametric models on out-of-sample data, a valuable insight for both researchers and practitioners.

7.3. Future Work

Future work I would like to see in this field is:

- Apply the VIX-LSTM methodology to a diverse range of financial assets, such as individual stocks, commodities, or cryptocurrencies. This would test the generalisability of the findings and determine the model's effectiveness on assets that may exhibit different volatility dynamics than an index fund.
- Compare the VIX-LSTM against other deep learning architectures such as Transformers, which have shown great success in other sequence-modelling tasks. This could lead to further RMSE improvements.
- Given the power of VIX, exploring the predictive value of incorporating other exogenous variables. Features such as trading

volume, interest rates, or market sentiment indicators derived from news headlines could provide the model with additional, valuable information.

7.4. Concluding Remarks

This dissertation successfully achieved its primary aim of developing and evaluating a novel forecasting model for S&P 500 volatility. The VIX-LSTM was proven to be a superior alternative to both traditional econometric models and complex hybrid models. The findings not only provide a valuable tool for practitioners but also show the importance of deep learning approaches to solve complex problems in financial econometrics.

8. BCS Project Criteria and Self Reflection

8.1. BCS Project Criteria

1. An ability to apply practical and analytical skills gained during the degree programme.
   I demonstrated practical skills in Python by using key data science and machine learning libraries such as pandas, arch and TensorFlow, as shown in Chapter 3. I demonstrated analytical skills through the manipulation of financial data. I also implemented multiple forecasting models and evaluated their performance, as shown in Chapter 3 and Chapter 4.
2. Innovation and/or creativity.
   I displayed innovation and creativity by combining econometric forecasting models with deep learning networks to make a hybrid model, as shown in Chapter 3 and Chapter 4.
3. Synthesis of information, ideas and practices to provide a quality solution together with an evaluation of that solution.
   Synthesis of information was key to this project, as shown in Chapter 1 and Chapter 2. The literature review covers ideas about theory-driven econometric models, data-driven Deep Learning models, financial stylised facts and the hybrid model's architecture. My 'quality solution' was the hybrid model, which was subject to extensive evaluation, as shown in Chapter 4.
4. That your project meets a real need in a wider context.
   This project addressed a high-value need within the global financial industry. Accurate volatility forecasting is important for risk management and portfolio optimisation, as shown in Chapter 1 and Chapter 5.
5. An ability to self-manage a significant piece of work.
   This project required independent self-management of a large task, from the initial planning and literature review to the implementation and testing of the best-performing forecasting model, as well as writing the dissertation.
6. Critical self-evaluation of the process.
   I self-evaluated continuously during each methodological decision, such as the choice of forecasting methodology and the final model architecture. Extensive self-evaluation of the project's success, challenges and learnings are presented in the discussion chapter and below.

8.2. Self Reflection

This dissertation represented a significant academic challenge as well as a personal journey in learning and applying my skills to the field of data science and AI.

My main challenge during this project was scope creep. The more I researched, the more possibilities I discovered, like more models to test, more metrics to evaluate and complex reasons for model failures that I wanted to investigate. This spiralled to a point where the project was becoming unsustainable within the given timeframe, which in turn led to a feeling of being rushed as deadlines approached. A crucial turning point was discussing this with my supervisor, who guided me to re-narrow the scope of my project. I learned to prioritise the core research question and make pragmatic decisions rather than pursuing every single interesting avenue.

On a technical level, I consider two areas to be key strengths of this project. Firstly, the depth of the background research was essential. I dedicated significant time to understanding how to implement the various econometric and LSTM models, as well as the theory behind why and how they work. This foundational knowledge was invaluable when interpreting unexpected results, like the failure of the hybrid model to outperform its simpler counterpart. Secondly, I am proud of the technical achievement of successfully constructing several innovative models, including the APARCH-VIX-LSTM model. While the simpler VIX-LSTM ultimately proved superior, the process of building and debugging these more complex networks was a significant practical learning experience.

Ultimately, this project taught me that success is a balance between technical depth and pragmatic project management. My greatest learning was not just how to build a state-of-the-art forecasting model, but how to define a clear, achievable research scope and adapt to the inevitable challenges and ‘dead ends’ that arise. This dissertation has solidified my technical skills in Python, TensorFlow and financial econometrics as well as my confidence in managing an extensive, independent research project from conception to conclusion.

9. Bibliography

[1] S.-H. Poon and C. W. J. Granger, 'Forecasting Volatility in Financial Markets: A Review', 2003.

[2] T. Bollerslev, 'GENERALIZED AUTOREGRESSIVE CONDITIONAL HETEROSKEDASTICITY', 1986.

[3] T. Fischer and C. Krauss, 'Deep learning with long short-term memory networks for financial market predictions', *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: 10.1016/j.ejor.2017.11.054.

[4] R. Engle, 'Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation', 1982.

[5] J. Campbell and L. Hentschel, 'No News is Good News: An Asymmetric Model of Changing Volatility in Stock Returns', National Bureau of Economic Research, Cambridge, MA, w3742, June 1991. doi: 10.3386/w3742.

[6] D. B. Nelson, 'Conditional Heteroskedasticity in Asset Returns: A New Approach', *Daniel B Nelson*, 1991.

[7] L. R. Glosten, R. Jagannathan, and D. E. Runkle, 'On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks', *J. Finance*, vol. 48, no. 5, pp. 1779–1801, Dec. 1993, doi: 10.1111/j.1540-6261.1993.tb05128.x.

[8] Z. Ding, C. W. J. Granger, and F. Engle, 'A long memory property of stock market returns and a new model', 1993.

[9] N. F. Johnson, *Financial market complexity*. Oxford: Oxford University Press, 2003. doi: 10.1093/acprof:oso/9780198526650.001.0001.

[10] Sepp Hochreiter, 'The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions', 1991.

[11]Sepp Hochreiter and Jurgen Schmidhuber, 'Long Short Term Memory', 1997.

[12] O. Calzone, 'Medium', An Intuitive Explanation of LSTM. Accessed: Sept. 11, 2025. [Online]. Available: https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c

[13] H. Wang, 'VIX and volatility forecasting: A new insight', *Phys. Stat. Mech. Its Appl.*, vol. 533, p. 121951, Nov. 2019, doi: 10.1016/j.physa.2019.121951.

[14] H. Y. Kim and C. H. Won, 'Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models', *Expert Syst. Appl.*, vol. 103, pp. 25–37, Aug. 2018, doi: 10.1016/j.eswa.2018.03.002.

[15] N. Roszyk and R. Ślepaczuk, 'The Hybrid Forecast of S&P 500 Volatility ensembled from VIX, GARCH and LSTM models', July 23, 2024, *arXiv*: arXiv:2407.16780. doi: 10.48550/arXiv.2407.16780.

[16] S. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions', Nov. 25, 2017, *arXiv*: arXiv:1705.07874. doi: 10.48550/arXiv.1705.07874.

10. Appendix

10.1. Appendix A: GARCH Order Selection

The table represents the results of the GARCH order selection process conducted on the validation dataset. The GARCH(1,1) specification was selected as it offered the best balance of model fit (low AIC/BIC) and parameter stability.

GARCH(p, q)	AIC	BIC	Outcome
(1,1)	-24410.3	-24384.7	Selected
(1,2)	-16364.8	-16332.8	
(2,1)	65463.2	65495.3	
(2,2)	-28845.5	-28807.1	Rejected (Insignificant parameters)

Table A: AIC and BIC scores for different orders of GARCH

10.2. Appendix B: LSTM Hyperparameter Tuning

The table shows the results of the grid search for the optimal LSTM hyperparameter conducted on the validation set. The configuration with 50 units, a 0.2 dropout rate, and a batch size of 32 was selected as it was the best balance of performance and computational efficiency.

Units	Dropout	Batch Size	RMSE
50	0.2	32	0.141554
50	0.2	64	0.141811
50	0.3	32	0.141601
50	0.3	64	0.141761
100	0.2	32	0.141549
100	0.2	64	0.141696
100	0.3	32	0.141568

100	0.3	64	0.141824
150	0.2	32	0.141600
150	0.2	64	0.141705
150	0.3	32	0.141547
150	0.3	64	0.141704

Table B: Hyperparameter tuning table