SUNWAY UNIVERSITY | Faculty of ENGINEERING & TECHNOLOGY

CAPSTONE PROJECT 2

Final Report

Comparative Analysis of GARCH-family Econometric Models vs LSTM Models for Volatility Forecasting: A Proxy-Robust Approach in US Equities

by

DARRANCE BEH HENG SHEK

23094907

BSc (Hons) in Computer Science

Supervisor: Dr Muhammad Rabani Mohd Romlay

Semester: May 2026

Date: 3 August, 2026

School of Computing and Artificial Intelligence

Faculty of Engineering and Technology

Sunway University

Abstract

Volatility forecasting underpins portfolio risk management, derivatives pricing, and regulatory capital requirements. While the classical Generalized Autoregressive Conditional Heteroskedasticity (GARCH) family of models has long served as the industry standard, Long Short-Term Memory (LSTM) networks are increasingly proposed as superior alternatives due to their non-linear capabilities. However, applied machine learning literature frequently evaluates these models using proxy-susceptible loss functions, omits formal significance testing, and ignores market regime shifts.

The primary objective of this project is to construct a mathematically sound backtesting framework to systematically evaluate whether deep learning architectures genuinely outperform classical econometric models for short-horizon equity volatility forecasting under proxy-robust conditions. Using 2,766 daily OHLCV observations of the SPDR S&P 500 ETF (SPY) from January 2015 to December 2025, volatility was proxied by the squared logarithmic return. The study compared GARCH(1,1), EGARCH(1,1), and GJR-GARCH(1,1,1) against an LSTM architecture trained under Quasi-Likelihood (QLIKE) and Mean Squared Error (MSE) loss functions. To ensure a strictly fair comparison, all models were restricted to the same foundational dataset, though the LSTM utilized a multivariate feature representation while GARCH specifications relied exclusively on closing returns.

Evaluated over 1,760 out-of-sample trading days, EGARCH(1,1) achieved the lowest overall QLIKE (1.4715). The Diebold-Mariano test revealed no statistically significant full-sample difference between the baseline econometric models and the QLIKE-trained network. However, regime-conditional analysis demonstrated that the LSTM significantly outperformed classical models during stable bull markets, but degraded severely during the acute COVID-19 crash. Furthermore, GARCH(1,1) was the only model to successfully pass the 95 per cent Value-at-Risk conditional coverage test. The findings establish that GARCH-family models remain structurally superior for navigating extreme market stress, while deep learning offers competitive advantages strictly during stable macroeconomic regimes.

Chapter 1 – Introduction

1.1 Introduction

Volatility forecasting is a critical foundational task in quantitative finance, underpinning aspects such as derivatives pricing, portfolio risk management, and regulatory capital requirements. Unlike an asset's price, which can be observed directly from a market feed, there is no direct or definitive way to observe volatility at all, and thus volatility must always be inferred. This inference can either from a historical proxy such as squared returns, or from a forward-looking market signal. This gap between what practitioners need to know and what can actually be measured has made volatility forecasting a persistently active area of both academic research and industry tool-building for over four decades.

For much of that history, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) family of econometric models has served as the dominant approach, reputable for its interpretability, low data requirements, and strong theoretical grounding. However, in the past decade, deep learning architectures, such as Long Short-Term Memory (LSTM) networks, have been increasingly proposed as superior alternatives, known for their ability to ingest and learn complex, non-linear temporal dependencies directly from data without requiring a pre-specified functional form.

In applied literature, claims that deep learning outperforms classical volatility models are common, but they are frequently built on evaluation practices with inappropriate loss functions, absent significance testing, and aggregated rather than regime-aware results that would not withstand close statistical scrutiny. This project is motivated by that gap. Rather than proposing another forecasting architecture, it undertakes the design and implementation of a rigorous, reproducible backtesting framework capable of testing, fairly and under statistically defensible conditions, whether the added complexity of deep learning is genuinely justified for short-horizon equity volatility forecasting. In doing so, the project contributes not only an empirical finding, but a transferable evaluation methodology relevant to any setting where a complex model is compared against a simpler, established baseline on noisy, real-world time-series data, which is of growing importance as increasingly sophisticated machine learning systems are adopted in high-stakes financial decision-making.

1.2 Background

The mathematical modeling of volatility is rooted in well-documented statistical occurrences that any credible forecasting model must account for. Financial returns exhibit a pattern of volatility clustering, where large price movements tend to be followed by further large movements of either sign, while calm periods tend to persist [1]. They also display mean reversion, whereby volatility eventually reverts toward a long-run average rather than trending indefinitely, and an asymmetric response to shocks, commonly termed the leverage effect, in which negative returns tend to raise future volatility more than positive returns of equal magnitude [2].

These regularities directly motivated the development of the GARCH family of models. Engle's ARCH model first proposed that conditional variance could itself be modeled as a function of past shocks [3], an idea generalized by Bollerslev into the GARCH framework [4]. Subsequent extensions, such as Nelson's EGARCH [5] and the GJR-GARCH model of Glosten, Jagannathan, and Runkle [6] further incorporated the leverage effect. These models remain valued because every parameter carries a clear economic interpretation, they can be estimated on relatively small datasets, and their forecasts can be explained and audited.

In parallel, Long Short-Term Memory (LSTM) networks [7] have gained increasing attention for volatility forecasting, owing to their capacity to learn arbitrary non-linear relationships directly from data. However, closer examination reveals persistent weaknesses in how comparisons between these two model families are conducted. Patton formally demonstrated that common metrics such as Mean Squared Error (MSE) are not statistically robust when evaluated against a noisy, unobserved volatility proxy, which is a condition that always holds in practice, and that the QLIKE loss function should be preferred instead [8]. Separately, the Diebold-Mariano test provides a formal method for establishing whether a difference in forecast accuracy between two models is statistically significant, rather than a product of sampling variation in a single test window [9]. Both tools remain inconsistently applied across the applied literature comparing classical and deep learning volatility models, which frequently reports only conventional error metrics and rarely conditions results on the prevailing market regime.

1.3 Problem Statement

Despite substantial academic and industry interest in applying deep learning to volatility forecasting, practitioners currently lack a reliable, methodologically sound basis for deciding whether to adopt LSTM-based models over well-established classical alternatives. This uncertainty stems from three interrelated problems evident in the existing literature and practice.

First, evaluation metrics are frequently statistically inappropriate. As true volatility can never be directly observed, forecast accuracy must be measured against a proxy, which typically refers to squared returns. However, it has been shown that MSE-based rankings can be distorted, even reversed, under such proxy noise [8]. The QLIKE loss function, which is robust to a much broader class of proxy noise, remains comparatively underused in applied and practitioner-facing work.

Second, claims of model superiority are rarely tested for statistical significance. A numerically lower error score for one model is frequently treated as conclusive evidence of superiority, without any formal test such as the Diebold-Mariano test [9] to establish whether the difference is genuine or attributable to sampling variation within a single, finite backtest.

Third, model comparisons are almost never conditioned on market regime. A single aggregated performance score across an entire historical sample can conceal sharply different behavior in calm versus turbulent periods, which is a critical blind spot, since volatility forecasts are arguably most valuable precisely when markets are under stress, such as during a liquidity crisis or a sudden macroeconomic shock.

Collectively, these three problems represent a significant computing, applied data science, and software engineering challenge of designing and implementing a pipeline that ingests real market data, correctly implements both classical econometric and deep learning forecasting models under matched, leakage-free conditions, and applies statistically appropriate, regime-aware backtesting methodology rather than simply training two models and comparing arbitrary error values. Much of the existing literature glosses over the architectural complexities of building a leakage-free, reproducible backtesting engine. Practitioners lack an integrated software solution that automates data ingestion, safely handles sequence-based memory management for deep learning models, and bridges the gap between classical econometrics and

modern machine learning frameworks. Addressing this problem requires designing a robust system architecture that can train custom LSTM models using non-standard, proxy-robust loss functions (such as QLIKE) within their optimization loops, ensuring the computational framework itself does not introduce bias into the model comparison.

1.4 Research Objectives

The main objective of this project is to design, implement, and rigorously evaluate a backtesting framework that fairly compares classical GARCH-family econometric models against an LSTM-based deep learning model for short-horizon (1-day-ahead) equity volatility forecasting. This project would contribute both an empirical finding on model performance and an evaluation methodology for comparing complex and classical time-series models under statistically defensible conditions.

The specific objectives are:

1. To engineer an automated, robust data pipeline that ingests historical daily OHLCV equity data, programmatically handles missing data ticks, and derives reliable volatility proxies without look-ahead bias.
2. To implement and calibrate three classical econometric baseline models: GARCH(1,1), EGARCH, and GJR-GARCH.
3. To design, build, and optimize an LSTM-based deep learning architecture, specifically implementing a custom QLIKE loss function within the neural network's training loop to evaluate its performance against standard MSE optimization.
4. To develop an object-oriented backtesting engine capable of executing formal statistical hypothesis testing (Diebold-Mariano) and efficiently computing proxy-robust loss metrics across large time-series datasets.
5. To segment backtesting results across distinct historical market regimes (e.g., low-volatility bull markets, the 2020 COVID-19 crash, and the 2022 interest rate-hike cycle) to assess whether relative model performance is stable or regime-dependent.

1.5 Scope and Limitations

1.5.1 Scope

- Asset Universe: The study is limited to the S&P500 index, represented by the SPDR S&P 500 ETF Trust (SPY). This limitation is due to data availability and consistency with prior literature on volatility model benchmarking [10].
- Data Frequency and Source: Daily-frequency Open, High, Low, Close, Volume (OHLCV) data is used, representing the standard for financial market time series.
- Time Horizon: Historical data spanning approximately 2015 to 2025 is used, deliberately selected to capture multiple distinct volatility regimes required for regime-conditional analysis.
- Forecast Horizon: The project is restricted to short-horizon (1-day-ahead) forecasting, the most standard and well-studied horizon in the volatility forecasting literature.
- Model Set: The comparison is limited to GARCH, EGARCH, GJR-GARCH, and a single LSTM architecture trained under two loss functions, QLIKE and MSE. No exhaustive architecture search (e.g., GRU, Transformer variants) or hybrid GARCH-LSTM models are attempted, to preserve a focused and interpretable comparison.
- Technology Stack & Computing Constraints: The system was developed primarily in Python. Deep learning architectures were constructed using PyTorch, requiring careful tensor memory management for sequence data. The backtesting engine was heavily vectorized using NumPy/Pandas to ensure computational efficiency over large historical datasets.

1.5.2 Limitations

- Proxy Uncertainty: As true volatility is fundamentally unobservable, all evaluation is necessarily conducted against a proxy. While QLIKE is used specifically to mitigate the distortion this can cause [8], some residual uncertainty in model ranking cannot be eliminated.
- Limited Asset-Class Generalizability: Findings are derived from specific US equity/ETF data only and may not generalize to other asset classes (e.g., commodities, foreign exchange, or cryptocurrency), which can exhibit materially different volatility dynamics.

- LSTM Sensitivity: Deep learning performance can vary with hyperparameter choices and random initialization. While reasonable design care is taken, an exhaustive hyperparameter search is outside this project's scope, which may understate the best achievable LSTM performance.
- Data Granularity: The reliance on daily data instead of intraday data, which is necessary to remain within the computational requirements, means the project cannot examine intraday volatility dynamics or microstructure effects that a higher-frequency study might reveal.

1.6 Research Significance

This project provides a comprehensive evaluation of the real-world impacts and sustainability of integrating deep learning into high-stakes financial risk management. The significance of this research contributes directly to both computing and society in the following ways:

1. Contributions to Computing: By integrating a proxy-robust Quasi-Likelihood (QLIKE) loss function into a neural network's optimization graph, this research aims to establish a reusable, statistically defensible pipeline for evaluating complex machine learning models against classical baselines. It seeks to address persistent methodological gaps in applied computing literature by enforcing rigorous backtesting constraints and formal statistical significance testing.
2. Economic Sustainability and Real-World Impact: Accurate volatility forecasting is essential for the stability of financial institutions, as undercapitalized systems can trigger severe liquidity crises during market downturns. By critically examining how standard LSTM architectures handle unprecedented exogenous shocks compared to traditional econometric models, this project strives to provide practitioners with a clearer understanding of the structural boundaries of deep learning in risk-mapping systems, thereby supporting broader market sustainability.
3. Ethical Implications: The quantitative finance industry is rapidly adopting complex, black-box machine learning systems, which introduces challenges related to algorithmic opacity. This research addresses these issues by investigating the structural stability of interpretable, parametrically constrained models alongside opaque deep learning architectures. The project emphasizes the importance of algorithmic transparency in high-stakes environments, aiming to clarify the risk trade-offs between adopting novel computational techniques and maintaining systemic robustness.

Chapter 2: Literature Review

2.1 Introduction to Volatility Dynamics and Financial Stylized Facts

Volatility forecasting acts as a primary input for derivative pricing, portfolio optimization, and dynamic risk management [1]. Classical frameworks, such as the Black-Scholes-Merton model, rely on the assumption that asset returns follow a constant volatility and a normal distribution [15]. However, in reality, conditional volatility is a latent, unobservable variable that must be statistically inferred from historical price fluctuations [1].

Financial time series exhibit several universal structural characteristics, known as stylized facts:

1. Volatility Clustering: Documented by Mandelbrot, this describes the tendency for large price variations to be followed by further large variations, and periods of calm to persist [1].
2. Leptokurtosis: Asset returns feature fat-tailed distributions, meaning extreme market events occur far more frequently than a standard normal distribution predicts [1].
3. Leverage Effect: First noted by Black, this dictates that negative returns (market downturns) inflate subsequent volatility significantly more than positive returns of an equal magnitude [2].

An effective forecasting architecture must be able to capture these structural irregularities.

2.2 Classical Econometric Foundations: The GARCH Family Benchmark

The financial econometrics literature has addressed these stylized facts through conditional heteroskedasticity models [3]. Engle established the paradigm with the Autoregressive Conditional Heteroskedasticity (ARCH) model, which modeled conditional variance as a linear function of past squared market innovations [3]. However, ARCH frequently required a large number of lagged parameters, leading to overfitting [3].

Bollerslev generalized this framework to create the GARCH model [4]. The standard GARCH(1,1) specification models the conditional variance σ_t^2 as:

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

In the equation, $\omega > 0$ represents the baseline variance, $\alpha \geq 0$ dictates the short-term reaction to the previous period's squared market shock, and $\beta \geq 0$ captures the long-term persistence of past volatility, and covariance stationarity requires that $\alpha + \beta < 1$ [4].

Hansen and Lunde concluded that the GARCH(1,1) is highly robust, though it remains structurally limited when applied to equity returns because it cannot accommodate the asymmetric leverage effect [10].

2.3 EGARCH and GJR-GARCH Extensions

The standard GARCH(1,1) squares the innovation term ϵ_{t-1}^2, treating positive and negative shocks identically [4]. To capture the leverage effect, asymmetric extensions were developed [5].

Nelson's Exponential GARCH (EGARCH) model shifts the estimation target to the natural logarithm of the conditional variance, mathematically guaranteeing positive volatility estimates without imposing restrictive non-negativity constraints [5]:

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

The parameter γ explicitly measures the asymmetric response; a negative coefficient captures the leverage effect [5].

Alternatively, the GJR-GARCH model introduces asymmetry via a Boolean indicator function $I[\epsilon_{t-1} < 0]$ that activates exclusively during negative shock events [6]:

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \gamma I[\epsilon_{t-1} < 0] \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

Under this specification, a negative shock amplifies the variance by a combined factor of $(\alpha + \gamma)\epsilon_{t-1}^2$ [6].

Table 2.1: Summary of GARCH-Family Specifications

Model	Core Mechanism	Core Advantages
GARCH(1,1)	Autoregressive variance	High parsimony and theoretical tractability.
EGARCH	Log-variance formulation	Ensures positive variance without strict parameter bounds.
GJR-GARCH	Boolean threshold	Highly interpretable, nested structure.

2.4 The Daily Data Constraint versus High-Frequency Paradigms

A parallel advancement in volatility forecasting relies on high-frequency intraday data to construct Realized Volatility (RV), transforming volatility into an observable time series [11]. Models like the HAR-RV capitalize on 5-minute sampling intervals to generate highly accurate baseline forecasts [12].

However, many practitioners and computational systems are constrained to standard daily data frequencies. When high-frequency intraday data is unavailable, models are forced to extract the latent conditional variance directly from daily squared returns, which are notoriously noisy and inefficient proxies [11]. Operating strictly within the confines of daily Open, High, Low, Close, Volume (OHLCV) data removes the advantage of observable RV, placing a premium on a model's ability to efficiently filter daily noise and learn persistent volatility structures purely from end-of-day metrics [12].

2.5 Deep Learning: The LSTM Architecture

Artificial Neural Networks (ANNs) act as universal function approximaters, known for their ability to learn complex, non-linear relationships directly from data without strict parametric constraints. For time-series data, Long Short-Term Memory (LSTM) networks are the standard architecture [7]. LSTMs explicitly resolve the vanishing gradient problem found in standard Recurrent Neural Networks via an internal cell state C_t governed by gating mechanisms [7]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\widetilde{C_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \odot \tanh(C_t)$$

In the context of this OHLCV-constrained project, the input vector x_t is strictly limited to features derived from historical daily data, such as returns, realised-volatility measures over several windows, intraday range estimators and volume. The forget gate f_t determines which obsolete market noise to discard, while the input gate i_t updates the cell state with contemporary daily shocks. This allows the LSTM to inherently learn market dynamics, like the leverage effect, without explicit thresholding parameters.

2.6 The State of Comparative Literature: GARCH vs. Deep Learning

Recent studies in computing literature [16], [17] heavily favor LSTM networks for financial time-series predictions yet frequently overlook the statistical constraints of proxy noise. Furthermore, the literature comparing classical models to deep learning is rather fragmented. Machine learning studies often report that LSTMs outperform classical models, but these claims are frequently based on the LSTM's ability to ingest massive alternative datasets like natural language sentiment or options-implied indices.

Conversely, financial econometricians note that the deterministic mean-reverting and variance-clustering equations of GARCH approximate the true data-generating process of financial variance remarkably well [4]. When alternative datasets and high-frequency data are removed from the equation, standard GARCH models establish an incredibly high predictive benchmark. The critical academic debate is whether LSTMs retain their predictive superiority when forced to rely solely on the exact same data features as classical econometric models.

2.7 Forecast Evaluation Methodologies

A persistent methodological flaw in applied machine learning literature is the statistical evaluation of forecasts. As true conditional variance is unobservable at any frequency, researchers must rely on a noisy statistical proxy, such as squared daily returns, to compute forecast errors [8].

Patton mathematically proved that evaluating forecasts against a noisy proxy using standard symmetric loss functions, like Mean Squared Error (MSE) or Mean Absolute Error (MAE), can cause severe distortions in model rankings [8]. Furthermore, in financial risk management, forecasting errors are highly asymmetric. For example, under-predicting volatility leads to dangerous under-capitalization, whereas over-predicting merely results in marginally reduced leverage.

Consequently, the Quasi-Likelihood (QLIKE) loss function is the theoretically superior metric. It belongs to a specific family of loss functions that preserves the true ranking of models despite proxy noise [8]:

$$QLIKE \ = \frac{\hat{\sigma}_t^2}{h_t} - \ln\left(\frac{\hat{\sigma}_t^2}{h_t}\right) - 1$$

where $\hat{\sigma}_t^2$ is the observed proxy and h_t is the ex-ante forecast. Deep learning models are routinely optimized using mathematically flawed metrics like standard MSE, leading to sub-optimal weights. Integrating QLIKE directly into the neural network's loss function forces the model to respect the asymmetric realities of financial risk.

2.8 Statistical Significance in Model Comparison: The Diebold-Mariano Test

A marginally lower error metric is statistically insufficient to claim predictive superiority due to finite sample variations. The Diebold-Mariano (DM) test provides the framework to ascertain whether the observed loss differential between two models is significantly different from zero [9]. The DM test inherently corrects for autocorrelated forecast errors by scaling the sample mean of the loss differentials by a heteroskedasticity-and-autocorrelation-consistent (HAC) long-run variance estimator [9]. Without this test, claims that LSTMs outperform GARCH remain statistically indefensible.

2.9 Market Regimes and Structural Breaks

Financial markets are subject to violent structural breaks and regime shifts, such as the March 2020 COVID-19 crash or the aggressive 2022 interest rate hike cycles. While LSTMs can achieve superior accuracy during low-volatility bull markets, they are prone to severe overfitting. When subjected to unprecedented exogenous shocks, LSTMs often degrade rapidly. In contrast, the mathematically constrained mean-reverting properties of classical GARCH(1,1) models act as a stabilizing anchor during acute market stress. Thus, performance must be evaluated across segmented historical regimes to determine if model superiority is stable or merely an artifact of a specific market environment.

2.10 The Research Gap

The intersection of econometric and machine learning literature reveals a critical gap. While machine learning practitioners rapidly deploy LSTMs, they frequently do so using non-robust loss functions (MAE/MSE), omit formal significance testing (Diebold-Mariano), and ignore market regime shifts. Furthermore, many machine learning successes in volatility forecasting rely on high-frequency intraday data or expansive alternative datasets. There is a distinct lack of research that natively forces a deep learning architecture to learn under proxy-robust, econometrically sound conditions while restricted exclusively to daily OHLCV data. This capstone project fills this void by integrating the asymmetrically penalizing QLIKE loss function directly into an LSTM network's optimization graph and evaluating it against classical models under strictly identical daily-data constraints and market regime segmentations.

Chapter 3: Methodology

3.1 Introduction to Methodology

The primary objective of this project is to construct a rigorous, mathematically sound backtesting framework capable of systematically comparing classical GARCH-family models against Long Short-Term Memory (LSTM) networks for short-horizon volatility forecasting.

To achieve this, the methodology will transition the theoretical concepts and planning established in the literature review into a robust, object-oriented software pipeline. This chapter details the architectural system design, data engineering procedures, and model implementations to evaluate these forecasting paradigms under matched, leakage-free conditions. The design explicitly integrates modern computing tools and technologies tailored to solve complex algorithmic tasks, ensuring that the comparative analysis is uncompromised by computational bottlenecks or methodological flaws.

3.2 System Architecture and Technology Stack

To satisfy the requirements of high-performance time-series modeling and visualization, the system architecture is modularized into four primary components, namely being data ingestion, model orchestration, statistical evaluation, and visualization.

The technology stack was deliberately selected to support dynamic computation graphs and efficient vectorization, representing an effective use of modern tools.

- Core Language & Data Processing: Python 3.10+ serves as the foundation. Data manipulation and vectorized operations are handled by pandas and numpy, ensuring maximum memory efficiency when processing decades of daily OHLCV records.
- Classical Econometric Modeling: The ‘arch’ library is utilized to estimate the baseline GARCH(1,1), EGARCH, and GJR-GARCH models via Maximum Likelihood Estimation (MLE).
- Deep Learning Framework: PyTorch is selected over alternative frameworks due to its eager execution model and dynamic computational graph. This is a critical requirement

for integrating the custom, non-standard QLIKE loss function directly into the neural network's backpropagation loop.

3.3 Data Ingestion and Pipeline Engineering

Financial data is inherently noisy and prone to structural anomalies. The data pipeline is engineered to programmatically clean and standardize inputs while strictly preventing look-ahead bias, a fatal flaw in predictive financial modeling.

3.3.1 Data Sourcing and Cleansing

The system ingests historical daily Open, High, Low, Close, Volume (OHLCV) equity data for the S&P 500 ETF (SPY), spanning from 2015 to 2025. The historical data is programmatically sourced via the Interactive Brokers (IBKR) Python API, utilizing the 'reqHistoricalData' function to pull continuous daily bars directly from the exchange server. Upon ingestion, the pipeline programmatically screens for missing trading days (e.g., unexpected market closures) and forward-fills price levels, a standard pre-processing step to maintain information continuity for sequence learning. Furthermore, to ensure high-speed I/O operations and memory efficiency downstream, the cleansed dataset is serialized and stored locally using the Apache Parquet columnar file format.

3.3.2 Feature Engineering and Proxy Generation

As true volatility is unobservable, the pipeline must derive a target proxy from the raw OHLCV feed. The system calculates the daily logarithmic return series r_t:

$$r_t = \ln(\frac{P_t}{P_{t-1}})$$

where P_t is the adjusted closing price at day t. The unobservable daily conditional variance is proxied using squared returns, $\sigma_t^2 \approx r_t^2$. To optimize neural network convergence, the feature space is standard-scaled, with the scaler strictly fit only on the expanding training window to maintain chronological integrity.

3.4 Model Specifications and Implementation

The modeling phase requires translating distinct mathematical concepts such as parametric econometrics and non-linear deep learning into a unified programmatic interface to ensure a mathematically fair comparison.

3.4.1 Classical Econometric Baselines

The system instantiates three distinct objects for the classical baselines of GARCH(1,1), EGARCH, and GJR-GARCH. Each model is implemented using an expanding window approach. At each time step t, the models are calibrated on all available data up to $t-1$ using MLE, optimizing parameters such as ω, α, β and the asymmetric term γ. The converged parameters are then used to output a 1-day-ahead variance forecast h_t. The pipeline explicitly caches these forecasts into a continuous time-series vector for downstream statistical evaluation.

3.4.2 Long-Short-Term-Memory (LSTM) Deep Learning Architecture

The network architecture and hyperparameter configurations were selected via random search optimization. Crucially, to prevent data leakage, candidate architectures were evaluated exclusively on the 2018 holdout set, which resides entirely within the initial in-sample estimation window, ensuring no information from the out-of-sample evaluation period (2019-2025) influenced the model design. The finalized architecture consists of:

1. An input layer matching a designated sequence length of 10 days. While longer lookbacks were tested, they consistently degraded out-of-sample performance. As the engineered feature set already includes persistent volatility metrics—such as multi-window realized volatility and Exponentially Weighted Moving Averages (EWMA)—the long-term memory components are already summarized. Extending the sequence length beyond two trading weeks forced the network to process redundant noise rather than extracting additional signal.
2. A single LSTM layer of 32 hidden units. Deep learning in quantitative finance is often constrained not by model capacity, but by the low signal-to-noise ratio of financial

returns. Deeper configurations and wider configurations were tested, but they consistently resulted in severe overfitting on the noisy squared-return proxy.

3. Dropout regularization (p = 0.2) applied to the recurrent output. Given the small number of training sequences available at the first refit, which is roughly 1,000 days, the network was highly susceptible to memorizing specific historical volatility paths. A 20% dropout rate was implemented prior to the final projection to explicitly penalize this co-adaptation, forcing the model to learn broader, regime-agnostic representations of market shocks.
4. A fully connected linear output layer with a Softplus activation. As conditional variance is strictly non-negative, the raw linear output is passed through a Softplus function. Unlike a standard ReLU, which admits exactly zero forecasts (where the QLIKE loss and its gradient become mathematically undefined), Softplus is smooth everywhere and naturally bounded away from zero, preserving a stable gradient descent during optimization.
5. The network was optimized using the Adam optimizer with a batch size of 32. To prevent overfitting to historical market regimes, an early stopping mechanism was employed, halting training if the validation loss failed to improve for 25 consecutive epochs.

3.4.3 Custom Loss Integration (QLIKE)

The core technical contribution of this methodology is the implementation of the proxy-robust QLIKE loss function within the LSTM's optimization graph. Standard PyTorch functions inherently minimize Mean Squared Error (MSE), which mathematically distorts model rankings when evaluated against noisy proxies like r_t^2.

To enforce the LSTM to respect the nature of financial risk being asymmetric, a custom nn.Module is engineered to compute the QLIKE loss dynamically during backpropagation. Let y_t be the observed proxy (squared returns) and $\hat{y}_t$ be the network's variance forecast. The forward pass of the custom loss function calculates:

$$Loss = \frac{y_t}{\hat{y}_t} - \ln\left(\frac{y_t}{\hat{y}_t}\right) - 1$$

By replacing standard MSE with this metric, the optimization algorithm computes gradients that heavily penalize under-predictions of volatility, directly aligning the deep learning model's objective with econometric reality.

3.5 The Backtesting System

The backtesting system is designed using object-oriented programming (OOP) principles, prioritizing modularity, reproducibility, and computational efficiency. In order to simulate the robustness of real-world forecasting conditions and strictly prevent look-ahead bias, the pipeline utilizes an expanding window out-of-sample testing methodology. For any given day t, the models are trained and calibrated using exclusively historical data from time t_0 to $t-1$, generating a single 1-day-ahead forecast. The window then expands by one trading day, and the calibration process repeats. The GARCH-family models were re-estimated at every step. Re-training the network at the same frequency was not computationally practical, so the LSTM was retrained every 126 sessions and filtered forward on fixed weights in between. Section 4.6 reports a sensitivity analysis over retraining intervals of 21, 63, 126 and 252 sessions confirming that the comparison does not depend on this choice.

The system also implements programmatic regime segmentation. Instead of computing a single, aggregated performance metric across the entire dataset, the timeline is sliced into distinct market environments. The system specifically isolates low-volatility bull markets, the acute liquidity shock of the 2020 COVID-19 crash, and the prolonged structural shifts of the 2022 interest rate hike cycle. This isolation allows the methodology to evaluate whether the deep learning model's predictive superiority is consistent and stable or merely an artifact of a specific macroeconomic regime.

3.6 Forecast Evaluation and Formal Hypothesis Testing

A core principle of this methodology is the application of statistically defensible evaluation metrics. As the true conditional variance remains unobservable, all models are evaluated against the derived proxy of squared daily returns. The evaluation module computes standard symmetric metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and the coefficient of determination (R^2) to maintain comparability with existing literature. However, the primary evaluation metric is the mathematically robust QLIKE loss function.

To move beyond simple point-estimate comparisons, the methodology integrates the Diebold-Mariano (DM) test into the backtesting pipeline. The DM test formally determines whether the difference in forecast accuracy between the LSTM and the classical GARCH-family baselines is statistically significant. To account for the highly autocorrelated nature of financial time-series errors, the implementation scales the sample mean of the loss differentials using a Heteroskedasticity and Autocorrelation Consistent (HAC) variance estimator, ensuring the resulting p-values are mathematically sound.

3.7 Value-at-Risk (VaR) Estimation

While statistical loss functions measure abstract prediction errors, translating these forecasts into actionable risk metrics is essential for evaluating real-world financial utility. To achieve this, the generated variance forecasts are mapped to Value at Risk (VaR) estimations. VaR is a metric that quantifies the maximum expected loss of an asset over a specific time frame at a given confidence level. It depends on three core elements, a time period, a confidence level, and an amount of potential loss.

The one-day-ahead conditional variance forecasts (σ_t^2) generated by both the GARCH baselines and the LSTM architectures were used to compute 95% VaR thresholds. Assuming a standard parametric approach, the VaR is computed as $VaR_t = z\_\alpha\sqrt{\hat{\sigma}_t^2}$, where z_α represents the critical value corresponding to the chosen confidence level (e.g., 1.645 for 95%).

Once the VaR thresholds are generated, the models are subjected to rigorous regulatory backtesting using two standard frameworks:

1. The Kupiec Proportion of Failures (POF) Test: The Kupiec Proportion of Failures (POF) test evaluates whether the total number of VaR breaches aligns with the expected nominal rate [13]. For a 95% VaR model, exceptions should theoretically occur on exactly 5% of the trading days. The test utilizes a likelihood ratio statistic, LR_{POF}, to determine if the empirically observed failure rate significantly deviates from this target.
2. The Christoffersen Test: A critical vulnerability of many forecasting models is that they may produce the correct aggregate number of exceptions but allow those exceptions to cluster consecutively during acute market shocks, which would indicate a failure to

quickly adapt to surging volatility. To test for this, the Christoffersen Independence Test utilizes a two-state Markov chain to evaluate whether a VaR exception today increases the probability of an exception tomorrow [14]. The resulting independence likelihood ratio, LR_{CCI} , is combined with the Kupiec statistic to form the comprehensive Conditional Coverage test (LR_{CC}), which evaluates the joint null hypothesis that exceptions are both correct in frequency and completely independent.

Integrating this regulatory backtesting framework serves as the definitive empirical validation for the project's proxy-robust approach. By contrasting the Kupiec and Christoffersen pass rates of an MSE-trained LSTM against the custom QLIKE-trained LSTM, this methodology tests whether asymmetric loss optimization directly translates into superior capital preservation and risk capture.

3.8 Computing Factors and Constraints

The system architecture was designed to address core computational constraints.

- Performance and Scalability: Processing daily OHLCV data spanning a decade requires substantial computational overhead, particularly during the calibration and recalibration of the neural network. To maintain high performance, the data pipeline heavily leverages NumPy vectorization and the Apache Parquet file format. By serializing all historical datasets, engineered features, and cached out-of-sample forecast arrays as Parquet files, the system drastically reduces its storage footprint and accelerates in-memory data processing compared to traditional flat files. The PyTorch LSTM architecture handles sequence memory management by utilizing optimized tensor data loaders, ensuring that the gradient descent operations do not exceed memory limitations during backpropagation.
- Security and Ethics: In quantitative finance, data integrity is critical. The system is engineered to strictly prevent data leakage, ensuring no future market information would contaminate the historical training features. Ethically, the project addresses the black-box problem present in algorithmic finance. By benchmarking the highly complex LSTM against interpretable GARCH models, the methodology questions the ethical viability of deploying opaque machine learning systems in high-stakes risk management without definitively proving their statistical superiority.

Chapter 4: Results and Discussion

4.1 Dataset and Descriptive Statistics

The dataset consists of 2,766 daily observations of the SPDR S&P 500 ETF (SPY) from 2 January 2015 to 31 December 2025, obtained through the Interactive Brokers API. Prices are adjusted for dividends and expressed in US dollars. The first four years, through 31 December 2018, form the initial estimation window, and all results reported below are out-of-sample: 1,760 trading days from 2 January 2019 to 31 December 2025.

Data integrity was verified against a reconstructed New York Stock Exchange (NYSE) trading calendar, and both data sources agreed exactly, with 2,766 expected sessions and 2,766 observed, with no missing days, no duplicate timestamps, no non-positive prices and no bars violating the high-low envelope. Returns and squared returns were computed as defined in the Chapter 3 methodology and expressed in percentage points.

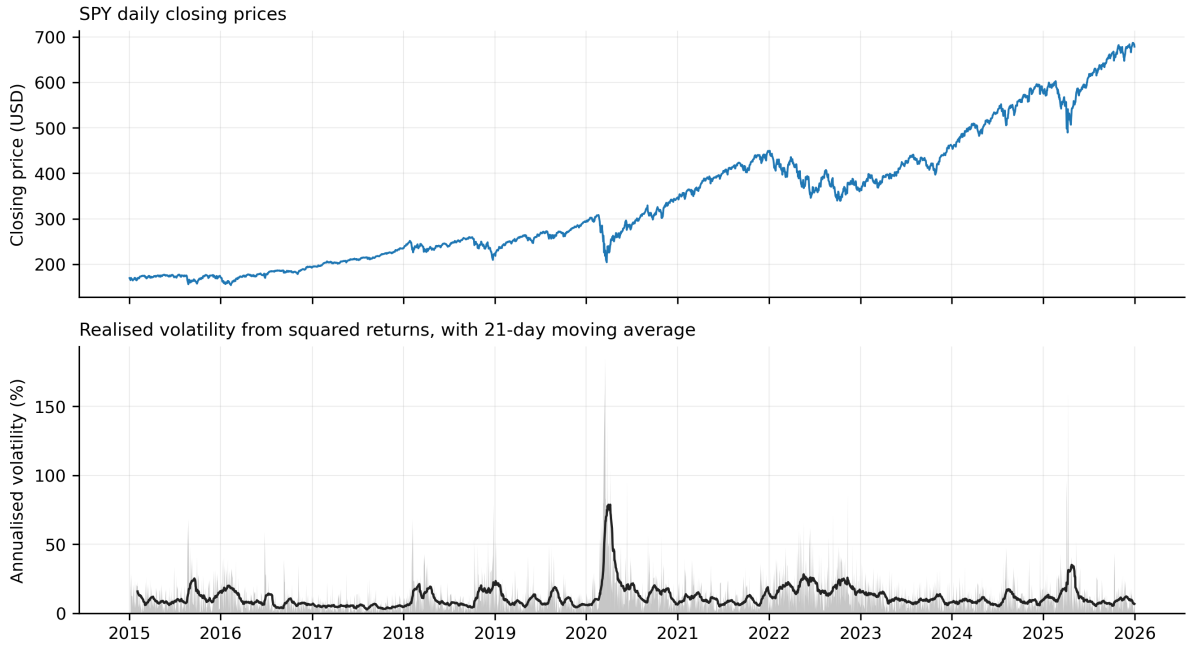

Figure 4.1: SPY daily closing prices (upper) and realised volatility from squared returns (lower), 2015 to 2025.

Figure 4.1 presents the price series and the corresponding realised volatility. The index rose steadily across the sample, interrupted by the March 2020 COVID-19 crash and by extended weakness through 2022. The volatility panel shows pronounced clustering: large movements

are followed by further large movements, and quiet periods persist. The March 2020 spike is by far the largest in the sample, with the 21-day average briefly exceeding 75 per cent annualised against a full-sample level of 17.8 per cent.

Table 4.1: Descriptive statistics for SPY volatility variables

Series	N	Mean	Std. dev.	Min	Max	Skewness	Kurtosis	JB p-value
Return (%)	2,765	0.0501	1.1233	-11.5875	9.9869	-0.5829	17.5353	0.000
Squared return (%²)	2,765	1.2639	5.1151	0.0001	134.2711	15.2987	308.5924	0.000

Kurtosis is reported on the raw scale, where a Gaussian distribution has a value of three. JB denotes the Jarque-Bera test of normality.

Table 4.1 reports the descriptive statistics of the dataset. The mean daily return is 0.05 per cent with a standard deviation of 1.12 per cent, giving an annualised volatility of 17.8 per cent. The return distribution is negatively skewed at -0.58, indicating that extreme negative returns occur more frequently than positive ones, which is a common phenomenon in equity markets. The kurtosis of 17.54 is far above the Gaussian benchmark of three, confirming the heavy-tailed nature of the return distribution. The squared returns exhibit strong positive skewness of 15.30, reflecting extreme volatility spikes and persistent clustering. The Jarque-Bera test rejects normality for both series at any conventional significance level. These statistics substantiate the use of ARCH-type frameworks, which explicitly model time-varying variance, and confirm that the stylised facts described in Chapter 2 are present in this sample.

4.2 Forecast Accuracy

Table 4.2 reports out-of-sample accuracy for all five models across four error metrics. QLIKE is the primary criterion, since it is the only one of the four that ranks forecasts consistently when the target is a noisy proxy for the latent variance.

Table 4.2: Comparative performance metrics of volatility forecasting models

Model	QLIKE	Rank	RMSE	MAE	R^2
GARCH(1,1)	1.4900	3	**5.2826**	1.6028	**0.2835**
EGARCH(1,1)	**1.4715**	**1**	5.3582	1.5036	0.2629
GJR-GARCH(1,1,1)	1.4746	2	5.3587	1.6028	0.2628
LSTM (QLIKE)	1.6450	4	6.1288	1.4883	0.0357
LSTM (MSE)	1.6862	5	6.1453	**1.4803**	0.0304

Lower is better for QLIKE, RMSE and MAE. Higher is better for R^2. Best value in bold.

Among the econometric models, EGARCH(1,1) achieved the lowest QLIKE at 1.4715, followed closely by GJR-GARCH(1,1,1) at 1.4746 and GARCH(1,1) at 1.4900. Both asymmetric extensions therefore outperformed the symmetric baseline, consistent with the leverage effect documented for equity indices. GARCH(1,1) recorded the lowest RMSE at 5.2826 and the highest R-squared at 0.2835.

The two LSTM architectures produced higher QLIKE values, 1.6450 for the QLIKE-trained network and 1.6862 for the MSE-trained network, placing both behind all three econometric models. They also recorded the highest RMSE values and the lowest R-squared values, at 0.0357 and 0.0304 respectively. Notably, however, both networks achieved the lowest MAE values in the study, at 1.4883 and 1.4803, ahead of every GARCH specification.

This divergence between metrics is expected. Mean absolute error is minimised by a forecast that stays near the middle of a right-skewed target distribution and avoids committing to large values, which is what the LSTM networks do. QLIKE, RMSE and R-squared all penalise the resulting under-prediction on high-volatility days. As QLIKE is the criterion that remains reliable under proxy noise, the ranking it produces is the one we carry forward.

4.3 Performance Across Market Regimes

The out-of-sample period was divided into five regimes using dates fixed in advance from documented macroeconomic events. Table 4.3 reports average QLIKE within each regime.

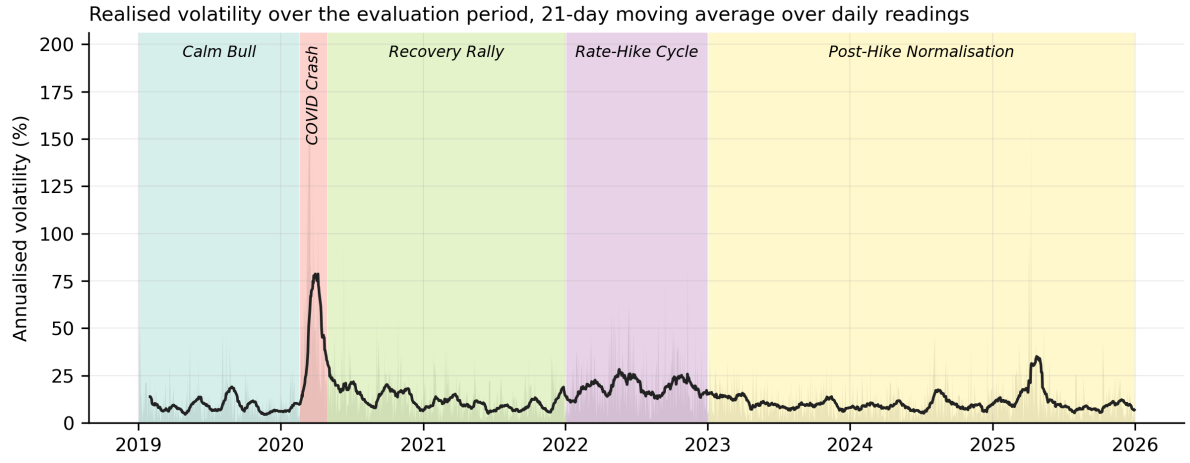

Figure 4.2: Realised volatility over the evaluation period, 21-day moving average over daily readings, with the five market regimes shaded.

Figure 4.2 shows the actual realized volatility environment the models were asked to forecast, with the five regime boundaries overlaid. The boundaries are used to separate genuinely different conditions based on real-world events and exogenous shocks rather than dividing the sample arbitrarily. The COVID band is narrow but contains the only stretch in seven years where the 21-day average rises above 75 per cent, roughly six times the level of the calm periods before or after it. The rate-hike band is elevated but sustained, peaking near 30 per cent without any single dramatic spike, showing a slower, more persistent kind of market stress rather than an abrupt shock like the COVID crash. The post-hike period returns to something close to the calm bull market, aside from a brief spike in April 2025. The pale daily readings behind the moving average also give an early sense of how noisy the squared-return proxy is at the daily frequency, which is the measurement problem the rest of the chapter works around.

Table 4.3: Average QLIKE by market regime

Model	Calm Bull	COVID Crash	Recovery	Rate-Hike	Post-Hike	Full sample
GARCH(1,1)	1.500	1.550	1.385	1.411	1.567	1.490
EGARCH(1,1)	1.413	1.635	1.472	1.389	1.510	1.472
GJR-GARCH(1,1,1)	1.442	1.372	1.450	1.430	1.522	1.475
LSTM (QLIKE)	1.369	7.072	1.471	1.306	1.599	1.645
LSTM (MSE)	1.421	8.116	1.468	1.364	1.589	1.686

Regime lengths are 285, 50, 422, 251 and 752 sessions respectively, with annualised volatility of 12.5, 66.1, 15.8, 24.2 and 15.3 per cent.

The regime breakdown reveals a pattern that the aggregate figures conceal. The QLIKE-trained LSTM achieved the lowest value of any model during the calm bull market, at 1.369, and during the rate-hike cycle, at 1.306. In the recovery rally and the post-hike period it was mid-table and close to the econometric models. Its poor full-sample result is driven almost entirely by the COVID-19 crash, where it recorded 7.072 against 1.372 for GJR-GARCH, a difference of more than five times.

GJR-GARCH(1,1,1) was the strongest model during the crash, ahead of GARCH(1,1) at 1.550 and EGARCH(1,1) at 1.635. Its threshold term, which amplifies the variance response specifically to negative shocks, is most valuable when large negative shocks are arriving. In calmer regimes EGARCH(1,1) tended to lead.

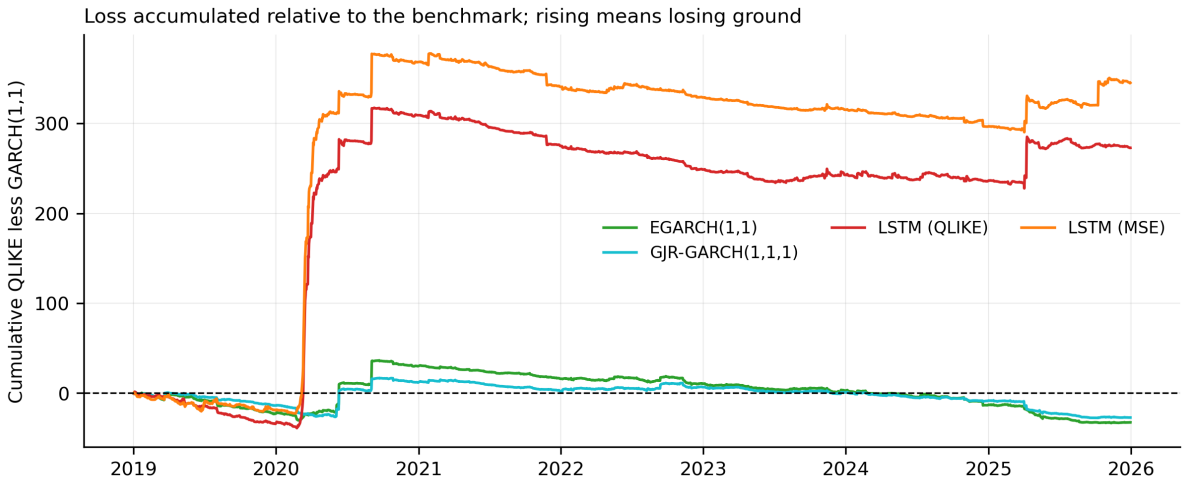

Figure 4.3: QLIKE accumulated relative to the GARCH(1,1) benchmark. A rising line indicates a model losing ground.

Figure 4.3 reveals the differences in performance that the simple aggregate full-sample performance concealed. Both LSTM lines decline through 2019 and into early 2020, reaching approximately -40, indicating that the networks were ahead of GARCH(1,1) over the first fourteen months of the evaluation. In March 2020 both lines rise almost vertically to roughly +280, after which they drift gradually downward again until a second, smaller step in April 2025.

The concentration is more extreme than the figure alone suggests. The eight largest daily loss differentials between the QLIKE-trained network and GARCH(1,1) sum to 273.8 QLIKE points, while the total accumulated across all 1,760 days is 272.8. Those eight sessions, representing less than half of one per cent of the evaluation period, account for the entire full-sample deficit. When we observe the remaining 1,752 days, the network is marginally ahead of the benchmark. Seven of the eight occurred in 2020 and the eighth on 9 April 2025.

4.4 Statistical Significance

Differences in average loss are only informative if they exceed what sampling variation would produce. Thus, the project applied the Diebold-Mariano test to the QLIKE loss differentials using an autocorrelation-robust variance estimator. Table 4.4 reports each model against the GARCH(1,1) benchmark, first over the full sample and then within each regime for the QLIKE-trained network.

Table 4.4: Diebold-Mariano tests against GARCH(1,1) on QLIKE loss

Scope	Model	DM statistic	p-value	Result
Full sample	EGARCH(1,1)	-0.754	0.4511	No difference
Full sample	GJR-GARCH(1,1,1)	-0.797	0.4255	No difference
Full sample	LSTM (QLIKE)	1.597	0.1104	No difference
Full sample	LSTM (MSE)	1.786	0.0742	No difference
Calm Bull	**LSTM (QLIKE)**	**-3.962**	**0.0001**	**LSTM better**
COVID Crash	**LSTM (QLIKE)**	**2.852**	**0.0063**	**GARCH better**
Recovery Rally	LSTM (QLIKE)	0.685	0.4938	No difference
Rate-Hike Cycle	**LSTM (QLIKE)**	**-3.730**	**0.0002**	**LSTM better**
Post-Hike Normalisation	LSTM (QLIKE)	0.482	0.6300	No difference

A negative statistic favours the model named in column two. Results use a 5 per cent significance level.

With these results, we observe that none of the full-sample differences are statistically significant. The observed advantage of EGARCH(1,1) over GARCH(1,1) in Table 4.2 carries a p-value of 0.4511, and the gap between GARCH(1,1) and the QLIKE-trained LSTM, despite amounting to about 10 per cent of the QLIKE value, has a p-value of 0.1104. On this evidence, the three econometric models and the two networks cannot be separated over the full evaluation period.

However, when we observe the regime-conditional tests, three of the five hold statistical significance. The LSTM network is significantly more accurate than GARCH(1,1) during the calm bull market and the rate-hike cycle, and significantly less accurate during the COVID-19 crash. The full-sample null result can therefore be interpreted as not an absence of significance but two opposing effects that cancel when averaged together.

4.5 Value-at-Risk Forecasting

Loss functions measure statistical accuracy but do not indicate whether a forecast would be adequate for risk management. To assess practical usefulness, we can use the Value-at-Risk metric. Each variance forecast was converted into a 95 per cent Value-at-Risk threshold using the Gaussian quantile, and the resulting exception sequence was tested with the Kupiec proportion-of-failures test and the Christoffersen conditional coverage test.

Table 4.5: 95 per cent Value-at-Risk coverage over 1,760 trading days

Model	Exceptions	Expected	Kupiec p	Joint p	Passes
GARCH(1,1)	106	88.0	0.0561	0.1562	Yes
EGARCH(1,1)	115	88.0	0.0047	0.0181	No
GJR-GARCH(1,1,1)	113	88.0	0.0087	0.0205	No
LSTM (QLIKE)	129	88.0	0.0000	0.0001	No
LSTM (MSE)	132	88.0	0.0000	0.0000	No

GARCH(1,1) is the only model to pass the joint conditional coverage test, recording 106 exceptions against an expectation of 88. The asymmetric specifications record 115 and 113 exceptions and fail, while both networks breach substantially more often, at 129 and 132. The Christoffersen independence component is high for most models, indicating that exceptions are not clustered in time even where there are too many of them; the problem is the level of the threshold rather than the speed of the model's reaction.

For practical purposes, this means GARCH(1,1) is the only model in the study that would satisfy a standard regulatory backtest at the 95 per cent level.

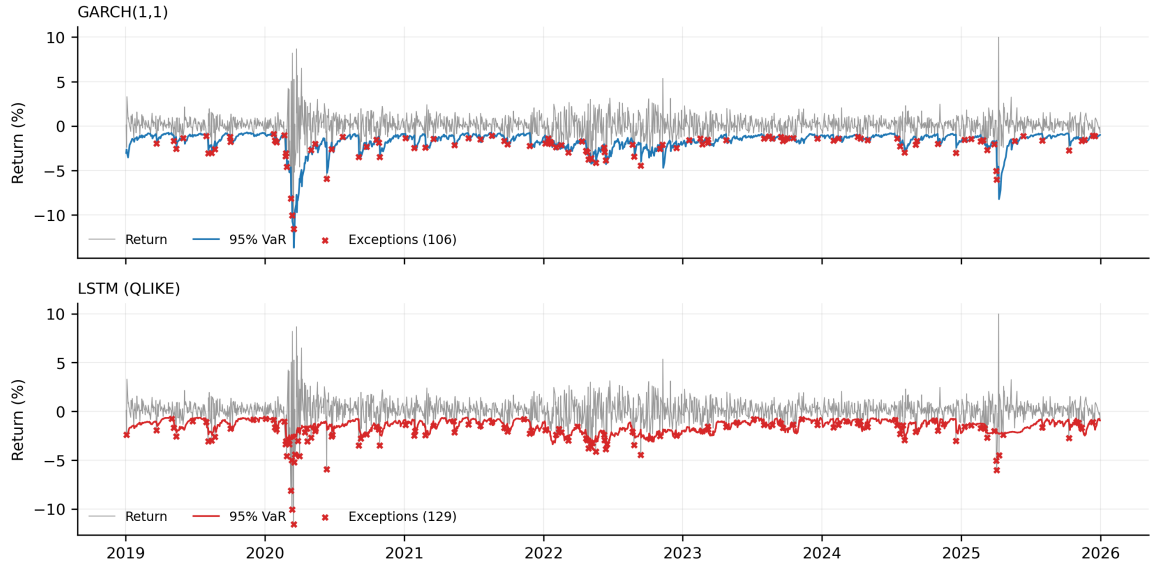

Figure 4.4: Daily returns against the 95 per cent VaR threshold, with exceptions marked, for GARCH(1,1) and the QLIKE-trained network.

Figure 4.4 plots the breach sequence, which occurs when an asset's actual returns are larger than the maximum loss predicted by the VaR model, for GARCH(1,1) and the QLIKE-trained network.

Figure 4.4 makes the practical consequence of the forecast ceiling visible. The GARCH threshold drops sharply in March 2020, reaching about minus 13 per cent at its widest, and then contracts again as conditions normalize as the model widens its risk estimate to match the environment and narrows it afterwards. The LSTM threshold barely moves across the same window, holding near minus 3 per cent, so a cluster of large negative returns passes straight through it. Over the full period the network records 129 exceptions against 106 for GARCH(1,1), and the difference between the two panels is concentrated exactly where a risk function would care most.

4.6 Discussion

The results highlight several points about the relative strengths of the two approaches. Under QLIKE, all three GARCH-type models produced consistent and stable results across regimes, with values ranging between 1.372 and 1.635 in every market environment. The asymmetric specifications, EGARCH(1,1) and GJR-GARCH(1,1,1), performed marginally better than the symmetric baseline overall, and GJR-GARCH(1,1,1) was clearly the strongest during the crash, confirming the value of modelling the leverage effect explicitly.

The LSTM architectures behaved differently. Rather than being uniformly weaker, they were competitive or better in four of five regimes and failed sharply in one. The cause is visible in the range of forecasts each model is capable of producing. Over seven years, the highest variance forecast issued by the QLIKE-trained network corresponds to approximately 32 per cent annualised volatility, compared with 132 per cent for GARCH(1,1) and 151 per cent for GJR-GARCH(1,1,1). The worst realised session in the sample corresponds to 184 per cent annualised.

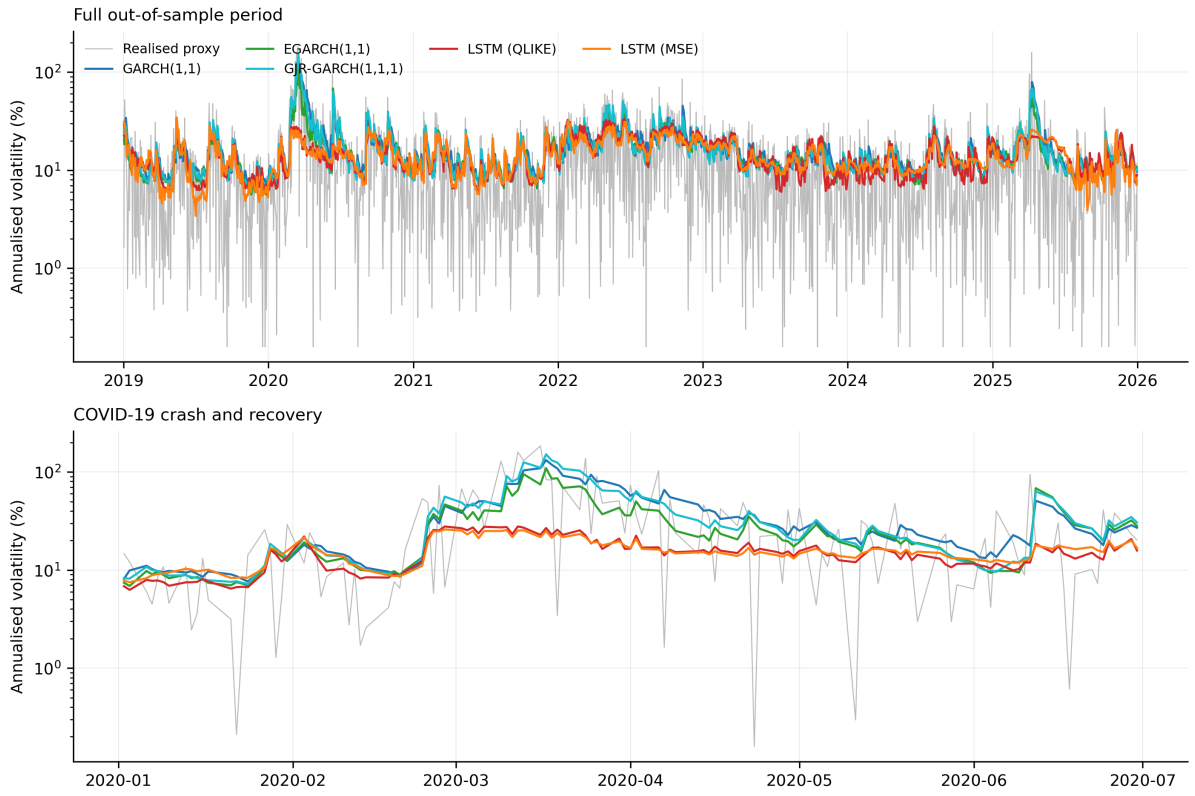

Figure 4.5: Forecasts against the realised proxy on a logarithmic axis. Upper panel, full out-of-sample period; lower panel, the COVID-19 crash and recovery.

When we observe Figure 4.5, in the upper panel the models are difficult to separate visually, which is itself informative as the differences that matter statistically are not apparent at this scale. However, the separation in the lower panel is clear. The GARCH lines climb above 100 per cent within days of the initial shock and continue to track the realised level upward, while both LSTM lines rise to roughly 25 to 30 per cent and then flatten for the remainder of the episode.

Two features of the LSTM network design produce this limit. First, standard recurrent neural network architectures inherently cap how high they can predict. The variance forecast is generated by applying a softplus function to a recurrent state bounded by its tanh activations, physically preventing the network from outputting a number high enough to match the extreme market movements during the COVID-19 crash. For the fitted network, this structural ceiling corresponds to about 32 per cent annualised volatility, and the highest forecast issued during the crash reached 75 per cent of it. Second, the input features are standardised using statistics from the training window, meaning inputs from a period far outside that window take extreme, out-of-distribution standardised values. Both are conventional choices in applied work, and both were recorded in Chapter 3 as design decisions rather than discovered afterwards.

There is a more general point here about the information set. A model that learns its response to shocks from historical examples can only forecast magnitudes that its training window taught it to consider, and the window ending in December 2019 contained nothing resembling March 2020. The GARCH recursion has no such requirement, because its conditional variance is a fixed function of whatever squared return actually arrives. When both families are restricted to daily price and volume data, with no intraday or forward-looking signal available to either, this structural difference is what separates them under stress. Much of the applied literature reporting deep learning superiority supplies the network with information the econometric baseline never receives, such as intraday realised volatility or sentiment indicators, which makes those comparisons a statement about the data rather than about the models.

Finally, the loss function used to train the network mattered less than expected. The QLIKE-trained network outperformed the MSE-trained network on QLIKE, 1.6450 against 1.6862, but the difference was not statistically significant. Its clearer benefit was stability: when the retraining interval was shortened from 126 to 21 sessions, the QLIKE-trained network improved to 1.5957 while the MSE-trained network degraded to 2.8696, because under squared-error loss a handful of extreme observations in a recently updated training window dominate the objective.

In terms of computational cost, GARCH(1,1) completed 1,760 full re-estimations in 17.4 seconds, while the QLIKE-trained LSTM required 50.6 seconds for 14 retrainings. The network is therefore more expensive overall despite re-estimating far less frequently, and given the absence of a statistically significant accuracy gain, that cost is not recovered on this data.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

This study compared three GARCH-family specifications against two LSTM architectures for one-day-ahead volatility forecasting of the SPY index, using daily OHLCV data from 2015 to 2025 and evaluating on 1,760 out-of-sample trading days. Both model families accessed the same daily OHLCV dataset, though by design the GARCH specifications consume closing returns alone while the LSTM network uses a broader set of engineered features.

The results demonstrate that GARCH-family models produced the better overall forecasting accuracy under QLIKE, with EGARCH(1,1) achieving the lowest value of 1.4715. GARCH(1,1) was also the only model to pass the 95 per cent Value-at-Risk conditional coverage test and was substantially cheaper to operate. However, the Diebold-Mariano test could not reject equal predictive accuracy between the econometric models and the QLIKE-trained network over the full sample, so the aggregate advantage is not statistically established. The regime-conditional analysis proved more informative than the aggregate comparison. The LSTM was significantly more accurate than GARCH(1,1) during the low-volatility bull market and the 2022 rate-hike cycle, and significantly less accurate during the COVID-19 crash. Eight trading days out of 1,760 account for its entire full-sample deficit. The cause is a limit on the magnitude of variance the network can forecast, arising from its bounded output parameterisation and from feature standardisation fitted to a calmer training window.

The practical implication is that model choice should reflect the intended operating environment. For risk management applications where performance under stress is the binding requirement, the GARCH family remains the more dependable choice, and GJR-GARCH(1,1,1) in particular during acute market shocks. For applications with typical market conditions, the LSTM is competitive and was significantly better in two of the five regimes examined. The findings also confirm that evaluating volatility forecasts requires a proxy-robust loss function and formal significance testing, since the ranking of the same five models reverses under mean absolute error and none of the full-sample differences survive a Diebold-Mariano test.

5.2 Limitations

1. Deep learning architectures are inherently constrained by their activation functions and scaling methodologies. The use of a Softplus head applied to a tanh-bounded recurrent state artificially capped the network from outputting a number high enough to match the extreme market movements. Furthermore, static feature standardization fit on calm historical windows maps unprecedented exogenous shocks (like the COVID-19 crash) to extreme, out-of-distribution tensor values. This highlights a fundamental limitation of supervised sequence learning in finance where models struggle to extrapolate magnitudes of volatility they have never explicitly seen in their training distribution.
2. The study covers a single, highly liquid asset in a developed market. SPY is one of the most heavily traded, highly liquid, and structurally efficient instruments globally. Its volatility dynamics may be uniquely well-behaved compared to less liquid equities. This may create a survivorship-like bias in the findings, as the comparative performance between GARCH and LSTM might completely invert when faced with illiquid assets that suffer from frequent sudden, discontinuous price gaps or noise.
3. Volatility is proxied by squared daily returns, which are unbiased but very noisy. This limits the R-squared values attainable by any model and is the reason no specification exceeded 0.29.
4. The methodology relies on a standard normal distribution assumption to map variance forecasts to VaR thresholds. However, real-world financial data is leptokurtic, and contains frequent extreme outliers that violate this normal assumption. Due to this mismatch, standard VaR formulas will inherently underestimate risk during market crashes, regardless of how accurate the underlying GARCH or machine learning predictions are. This highlights a limitation in the standard risk-mapping equation itself, rather than a flaw in the proposed forecasting models.
5. Only one neural architecture family was examined, at a single forecast horizon, without hybrid or attention-based alternatives.

5.3 Future Work

1. Extending the analysis to multiple assets and markets, and to longer forecast horizons, to establish whether the regime dependence observed here is general.

2. Future research should shift from daily OHLCV data to high-frequency intraday data to construct Realized Volatility (RV). This would reduce the measurement noise in the target variable, allowing deep learning models to fit on a much cleaner signal. This would also permit benchmarking against the Heterogeneous Autoregressive model of Realized Volatility (HAR-RV), providing a definitive test of whether LSTMs can outperform the industry's strongest high-frequency econometric baseline.
3. To resolve the forecast ceiling issue observed during the COVID-19 crash, future implementations must redesign the network's output layer and normalization pipeline. Predicting the natural logarithm of variance (similar to the EGARCH formulation) would allow the network to project unbounded positive variance without artificial Softplus saturation. Additionally, implementing dynamic, rolling-window feature standardization would prevent extreme exogenous shocks from pushing the network's inputs into dead, out-of-distribution activation zones.
4. Exploring the idea of hybrid GARCH-LSTM frameworks in which the econometric variance estimate is supplied to the LSTM network as an input feature, combining the response of the recursion with the flexibility that produced the LSTM network's regime-specific advantages. A hybrid architecture could use the parametric GARCH(1,1) variance forecast as an engineered input feature for the LSTM network, which would allow the network to delegate the baseline autoregressive and mean-reverting properties to the econometric formula, while the neural network focuses purely on learning the complex, non-linear residuals and regime-dependent asymmetric shocks.
5. While LSTMs are effective for sequence learning, they inherently suffer from recency bias, heavily weighting immediate historical shocks. Future research should evaluate Transformer-based architectures for volatility forecasting. The self-attention mechanism could theoretically learn long-term macroeconomic volatility cycles independently from short-term daily shocks, offering a more robust understanding of market regimes than a standard recurrent state cell.

References

[1] B. B. Mandelbrot, "The variation of certain speculative prices," The Journal of Business, vol. 36, no. 4, pp. 394–419, 1963.

[2] F. Black, "Studies of stock price volatility changes," in Proc. 1976 Meetings of the American Statistical Association, Business and Economics Statistics Section, 1976, pp. 177–181.

[3] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," Econometrica, vol. 50, no. 4, pp. 987–1007, 1982.

[4] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," Journal of Econometrics, vol. 31, no. 3, pp. 307–327, 1986.

[5] D. B. Nelson, "Conditional heteroskedasticity in asset returns: A new approach," Econometrica, vol. 59, no. 2, pp. 347–370, 1991.

[6] 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," The Journal of Finance, vol. 48, no. 5, pp. 1779–1801, 1993.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] A. J. Patton, "Volatility forecast comparison using imperfect volatility proxies," Journal of Econometrics, vol. 160, no. 1, pp. 246–256, 2011.

[9] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," Journal of Business & Economic Statistics, vol. 13, no. 3, pp. 253–263, 1995.

[10] P. R. Hansen and A. Lunde, "A forecast comparison of volatility models: Does anything beat a GARCH(1,1)?," Journal of Applied Econometrics, vol. 20, no. 7, pp. 873–889, 2005.

[11] T. G. Andersen and T. Bollerslev, "Answering the skeptics: Yes, standard volatility models do provide accurate forecasts," International Economic Review, vol. 39, no. 4, pp. 885–905, 1998.

[12] F. Corsi, "A simple approximate long-memory model of realized volatility," Journal of Financial Econometrics, vol. 7, no. 2, pp. 174–196, 2009.

[13] P. Kupiec, "Techniques for Verifying the Accuracy of Risk Measurement Models," *The Journal of Derivatives*, vol. 3, no. 2, pp. 73-84, 1995.

[14] P. F. Christoffersen, "Evaluating Interval Forecasts," International Economic Review, vol. 39, no. 4, pp. 841-862, 1998.

[15] F. Black and M. Scholes, "The pricing of options and corporate liabilities," Journal of Political Economy, vol. 81, no. 3, pp. 637–654, 1973.

[16] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654-669, 2018.

[17] Y. Kim and C. H. Won, "Forecasting the Volatility of Stock Price Index: A Hybrid Model Integrating LSTM with Multiple GARCH-Type Models," Expert Systems with Applications, vol. 103, pp. 25-37, 2018.