Mining the Gap: Harvesting Volatility Risk Premium in Options

Simeng Wu, Lucas Ma, Alice Wang, Luciana Wei
Duke University

April 28, 2026

Abstract

This paper studies the volatility risk premium in SPY options, the discrepancy between realized volatility and market-implied volatility. The main objective is to determine whether model-based forecasts of short-horizon realized volatility can be used to identify reliable option mispricing.

We develop a systematic options trading framework that combines volatility forecasting models with threshold-based trading signals. Specifically, we forecast five-day forward realized volatility with machine learning models, including GARCH-family models, Random Forest, LSTM, and Feedforward Neural Networks, with focus GARCH and Neural Networks. When the forecasted realized volatility is sufficiently above or below implied volatility of SPY option, the strategy takes it as a trading signal. The trading framework also incorporates liquidity filters, fixed holding periods, stop-loss rules, and profit-taking rules to improve practical implementability.

The results suggest that volatility forecasts contain useful information for identifying potential option mispricing, but performance is highly dependent on market regime, data availability and quality, model stability, and execution assumptions. Simpler models such as GARCH provide interpretability and deployment stability, while more flexible neural network models show potential for better performance in the long run by capturing nonlinear volatility dynamics. Overall, the study shows that the volatility risk premium can be approached through a structured forecasting and trading framework, but robust risk management and careful engineering are essential for practical application.

1 Introduction

Volatility plays a central role in financial markets, influencing asset pricing, risk management, and derivatives valuation. Among the most studied phenomena in volatility markets is the *volatility risk premium* (VRP), defined as the systematic difference between implied volatility embedded in option prices and subsequently realized volatility of the underlying asset. Empirically, implied volatility tends to exceed realized volatility on average, reflecting the compensation that option sellers require for bearing volatility risk.

The core intuition behind VRP-based strategies is straightforward. Implied volatility represents the market's consensus expectation of future volatility, while realized volatility reflects the actual movement of the underlying asset. When these two diverge meaningfully, it suggests that options may be mispriced. A key challenge, however, lies in accurately forecasting future realized volatility and translating this information into implementable trading strategies.

Our contributions are threefold. First, we integrate a forward-looking volatility forecasting model with a systematic options trading framework. Second, we introduce a dual-threshold signal construction that combines both absolute and relative volatility discrepancies to improve robustness across different volatility regimes. Third, we implement a set of disciplined position management and exit rules that align the trading horizon with the model forecast and control downside risk.

Overall, our approach demonstrates how predictive volatility models can be transformed into economically meaningful trading strategies, offering insights into the volatility risk premium and practical implications for options-based investment.

1.1 Background

1.1.1 Option Market Intuition

Options markets provide a forward-looking measure of expected volatility through implied volatility, which is inferred from option prices using models such as Black–Scholes. In contrast, realized volatility is backward-looking, calculated from historical price movements of the underlying asset. The difference between these two quantities reflects the market's expectations relative to actual outcomes.

In practice, implied volatility is not merely a forecast of future volatility; it also embeds risk premia. Investors are typically willing to pay a premium for downside protection, leading to systematically elevated implied volatility levels relative to realized volatility. As a result, option sellers tend to earn a premium over time, while option buyers benefit during periods of unexpectedly high volatility.

1.1.2 Economic Significance

The volatility risk premium has important implications for both asset pricing and portfolio management. From an asset pricing perspective, VRP reflects investors' aversion to uncertainty and their demand for hedging against adverse market movements. It is closely related to macroeconomic risk factors and tends to vary with market conditions, such as liquidity, leverage, and investor sentiment.

From a practical standpoint, VRP represents a persistent source of return for systematic trading strategies. Strategies that sell volatility aim to capture the premium embedded in option prices, while strategies that buy volatility seek to profit from periods where realized volatility exceeds expectations, such as during market shocks.

Despite its appeal, exploiting VRP is challenging. Transaction costs, model uncertainty, and market frictions can erode profitability. Moreover, volatility is inherently difficult to predict, and poorly calibrated models can lead to significant losses.

In this context, our approach contributes to the literature by combining a model-based volatility forecast with a structured trading framework, enabling a more precise and adaptive exploitation of the volatility risk premium.

2 Literature Review

2.1 Volatility Risk Premium

A large body of empirical literature documents that implied volatility systematically exceeds realized volatility on average, indicating a persistent positive volatility risk premium. Bollerslev et al. (2009) show that the VRP is time-varying and strongly predicts future stock returns, while Carr and Wu (2009) provide model-free measures of the variance risk premium derived directly from option prices.

These findings imply that profitability in volatility trading strategies depends critically on the ability to forecast realized volatility more accurately than the market's implied expectations.

2.2 Options-Based Evidence

The existence of the volatility risk premium is most directly observed in option markets, where prices are explicitly linked to implied volatility. Empirical studies show that option returns are systematically negative on average for buyers and positive for sellers, consistent with the presence of a premium for bearing volatility risk.

Coval and Shumway (2001) document that options, particularly index options, tend to have low average returns despite their hedging benefits, reflecting the cost of insurance embedded in option prices. Similarly, Broadie et al. (2009) show that option sellers earn compensation for exposure to tail risk, particularly during periods of market stress.

Variance swap literature provides a clean framework for isolating volatility exposure. Demeterfi et al. (1999) demonstrate how variance can be traded directly using options, while Jiang and Tian (2005) develop model-free implied volatility measures that allow for accurate estimation of the variance risk premium.

In addition, delta-hedged option strategies provide a practical method for isolating volatility risk. Bakshi and Kapadia (2003) show that delta-hedged option returns capture the volatility risk premium, independent of directional price movements. This is particularly relevant for volatility trading strategies that aim to profit from discrepancies between implied and realized volatility.

Overall, the options literature provides strong empirical support for the existence of VRP and highlights the role of tail risk, skew, and market frictions in shaping option prices.

2.3 Volatility Forecasting Models

Accurate forecasting of realized volatility is central to exploiting the volatility risk premium. A large body of literature has proposed models that capture the persistence, clustering, and asymmetry of volatility dynamics. In this study, we consider a range of econometric and machine learning models to evaluate their relative performance in forecasting forward volatility.

2.3.1 GARCH Models

The GARCH framework, introduced by Engle (1982) and extended by Bollerslev (1986), models conditional variance as a function of past shocks and past variance, capturing volatility clustering.

We consider a baseline GARCH(1,1) model under Gaussian innovations, followed by a Student-t specification to account for heavy tails in financial returns. These models serve as benchmarks for evaluating more complex specifications.

2.3.2 Asymmetric GARCH Extensions

To capture asymmetric volatility responses, we consider EGARCH and GJR-GARCH models. EGARCH models log-variance dynamics and naturally incorporates leverage effects but requires simulation for multi-step forecasting. To address Jensen's inequality when exponentiating, we use median aggregation across simulation paths.

GJR-GARCH introduces asymmetry through an indicator function and allows for analytical multi-step forecasts. We implement both Gaussian and Student-t variants to assess the impact of heavy-tailed innovations.

2.3.3 Machine Learning and Deep Learning Models

Recent literature has shown that machine learning-based models can outperform traditional econometric approaches, particularly in the presence of nonlinear dynamics and high-dimensional feature spaces Gu et al. (2020).

Feedforward Neural Networks Feedforward neural networks (or multilayered perceptrons, MLPs) approximate nonlinear relationships between inputs and volatility through layered transformations. Early applications of neural networks to financial time series demonstrate their ability to capture nonlinear patterns without imposing strong parametric assumptions Zhang et al. (1998). MLP shows powerful capability of mapping non-linear data with residual connection He et al. (2015).

Long Short-Term Memory (LSTM) LSTM networks extend recurrent neural networks by introducing memory cells that capture long-range dependencies Hochreiter and Schmidhuber (1997). This makes them particularly well-suited for modeling persistence, regime shifts, and volatility clustering in financial time series. Empirical studies find that LSTM models often outperform traditional volatility models such as GARCH in forecasting realized volatility, especially in nonlinear and nonstationary environments Nelson et al. (2017); Fischer and Krauss (2018). However, LSTM is poorly compatible with our trading strategy and is therefore not discussed in the result section.

Gated Recurrent Unit (GRU) The Gated Recurrent Unit (GRU) is a simplified variant of the LSTM architecture that reduces computational complexity while retaining the ability to model temporal dependencies Cho et al. (2014). GRUs have been shown to achieve performance comparable to LSTMs in many financial forecasting applications, making them an efficient alternative when computational cost is a concern. We attempted GRU without any promising result.

2.4 Gaps in Literature

Despite extensive research, several gaps remain. First, most studies focus on either econometric or machine learning models in isolation, with limited unified comparison. Second, while VRP is well-documented, fewer studies translate volatility forecasts into implementable trading strategies. Third, the sensitivity of model performance to data construction and market regimes remains underexplored. This study addresses these gaps by integrating econometric and deep learning models within a unified framework and evaluating their effectiveness in generating tradable volatility signals.

3 Data

3.1 Data Sources

This study uses SPY, the ETF tracking the S&P 500, as the underlying asset due to its high liquidity in both equity and options markets. This ensures reliable pricing, tighter bid-ask spreads, and stable implied volatility estimates, which are critical for volatility-based trading strategies.

Daily SPY price data from QuantConnect are used to compute log returns, historical volatility, and the five-day forward realized volatility. Option chain data provide strike, expiry, and pricing information, from which near-the-money implied volatility is extracted as the market's expectation of future volatility.

All data and backtesting are conducted within QuantConnect, ensuring consistency between model development and trading implementation.

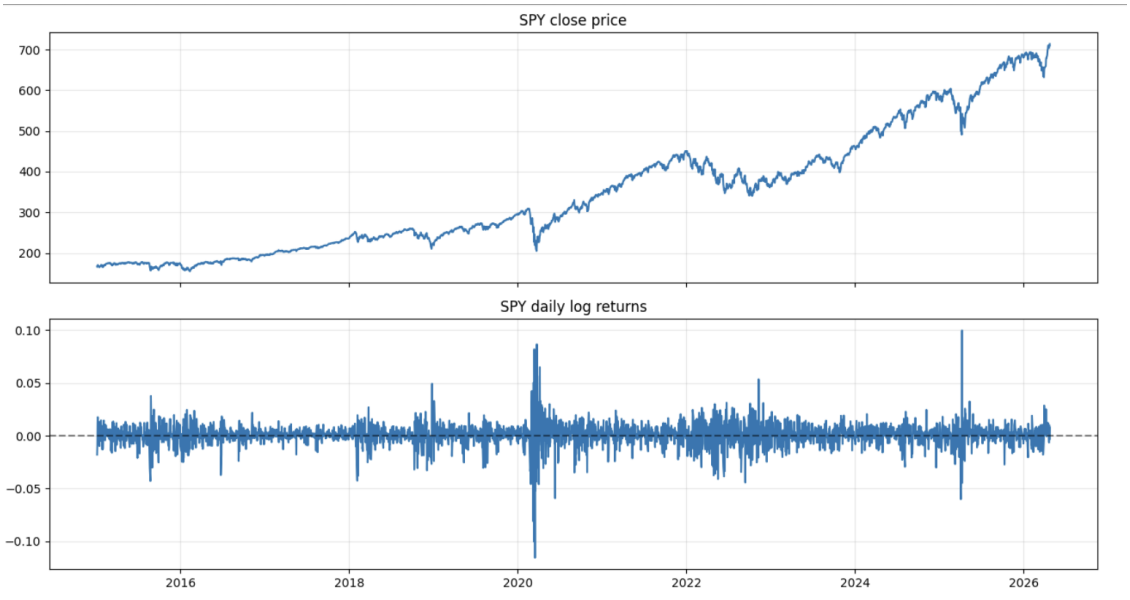

Figure 1: SPY price series and daily log returns. The top panel shows the adjusted closing price of SPY over time, while the bottom panel displays the corresponding daily log returns.

3.2 Variable Construction

The main variables used in this study are implied volatility, realized volatility, and the volatility risk premium. These variables are constructed to compare market-implied expectations of volatility with realized volatility measured from SPY returns.

Implied volatility. Implied volatility (IV) is obtained from SPY option prices and represents the volatility level implied by the option market. Economically, IV reflects the market's forward-looking expectation of volatility over the life of the option, while also incorporating volatility risk premia, supply and demand imbalances, and investor demand for downside protection.

In this study, IV serves as the market-based measure of expected volatility. Since the paper focuses on volatility trading, IV provides the benchmark against which model-based realized volatility forecasts are compared.

Realized volatility. Realized volatility (RV) is constructed from historical SPY returns. Let P_t denote the adjusted closing price of SPY on trading day t. The daily log return is defined as

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

Using these daily returns, realized volatility over a horizon of H trading days is computed as

$$RV_{t,t+H} = \sqrt{252 \times \frac{1}{H}\sum_{j=1}^{H} r_{t+j}^2}. \tag{2}$$

The factor 252 annualizes the volatility measure, since there are approximately 252 trading days in a year. In this paper, the main realized volatility target is the five-day forward realized volatility, corresponding to $H = 5$. This target is used for model training and forecast evaluation.

Volatility risk premium. The volatility risk premium (VRP) is defined as the difference between implied volatility and realized volatility:

$$VRP_t = IV_t - RV_t. \tag{3}$$

In the our analysis, the VRP provides the conceptual foundation for comparing option-implied volatility with subsequently realized volatility. The construction of a tradable signal based on forecasted realized volatility is discussed in the methodology section.

3.3 Model Setup

The objective of the model is to forecast short-horizon realized volatility and use this forecast to identify potential volatility mispricing in SPY options. Rather than predicting the direction of SPY returns, the model focuses on predicting the magnitude of future price movement. This is

appropriate for a volatility trading strategy because option profitability depends heavily on the relationship between implied volatility and realized volatility.

The general forecasting framework is written as

$$\widehat{RV}_{t,t+H} = f(X_t), \tag{4}$$

where $\widehat{RV}_{t,t+H}$ is the model forecast of realized volatility over the next H trading days, and X_t represents the information available at time t. Depending on the model specification, X_t may include past returns, lagged realized volatility, historical volatility measures, and other volatility-related features. The function $f(\cdot)$ represents the forecasting model, such as GARCH-family models or machine learning models.

The forecasting horizon is set to $H = 5$ trading days. This horizon is chosen because it is short enough to be relevant for active options trading, while still long enough to reduce noise from single-day return fluctuations. During model evaluation, the forecast is compared against the five-day forward realized volatility defined in Section 3.2.

In a real-time trading setting, future realized volatility is not observable at time t. Therefore, the strategy replaces future realized volatility with the model forecast and compares it against contemporaneous implied volatility. The trading spread is defined as

$$Spread_t = \widehat{RV}_{t,t+H} - IV_t. \tag{5}$$

A positive spread indicates that the model expects realized volatility to exceed the volatility currently priced by the option market. In this case, options may appear relatively underpriced, which may support a long-volatility position. A negative spread indicates that implied volatility exceeds the model's forecasted realized volatility, meaning options may appear relatively overpriced, which may support a short-volatility position.

To avoid excessive trading on small or noisy differences, the final strategy uses threshold rules. Trades are entered only when the spread between forecasted realized volatility and implied volatility is large enough to indicate a meaningful discrepancy. This helps reduce transaction costs, avoid overreacting to noise, and ensure that trades are based on stronger volatility signals.

3.4 Volatility Forecasting Framework

3.4.1 GARCH Family

This subsection describes the GARCH-family volatility forecasting models used in this study. These models are designed to capture time-varying conditional volatility in SPY returns and to generate short-horizon realized-volatility forecasts for the options trading framework. The daily log return and the five-day forward realized-volatility target are constructed as described in Section 3.2.

Return scaling. For numerical stability in maximum likelihood estimation, the GARCH-family models are estimated using percentage returns:

$$\tilde{r}_t = 100 r_t. \tag{6}$$

This rescaling does not change the economic content of the return series; it only changes the unit in which conditional variance is measured. The resulting conditional variance forecasts are therefore expressed in percent-return squared units. Before converting the forecasts into annualized volatility, the predicted variance is divided by 100^2.

Baseline GARCH(1,1). The baseline specification is a zero-mean GARCH(1,1) model:

$$\tilde{r}_t = \sigma_t z_t, \tag{7}$$

where z_t is a standardized innovation with mean zero and unit variance. The conditional variance evolves according to

$$\sigma_t^2 = \omega + \alpha \tilde{r}_{t-1}^2 + \beta \sigma_{t-1}^2. \tag{8}$$

Here, ω determines the baseline variance level, α measures the short-run response of volatility to recent squared return shocks, and β captures persistence in the conditional variance. The persistence of the standard GARCH(1,1) process is

$$\phi_{\text{GARCH}} = \alpha + \beta. \tag{9}$$

A value close to one implies that volatility shocks decay slowly.

The zero-mean specification is adopted because the conditional mean of daily SPY returns is small relative to daily volatility. Since this component of the framework focuses on volatility forecasting rather than directional return prediction, estimating a daily drift term is unlikely to add meaningful predictive information and may introduce unnecessary estimation noise.

Two innovation distributions are considered for the baseline GARCH model. The first uses standard normal innovations:

$$z_t \sim N(0,1), \tag{10}$$

while the second uses standardized Student-t innovations:

$$z_t \sim t_\nu, \tag{11}$$

where ν is the degrees-of-freedom parameter estimated from the data. The Student-t specification is included to account for the heavy-tailed behavior commonly observed in equity index returns. Changing the innovation distribution affects the likelihood and the estimated parameters, but it does not change the form of the conditional variance recursion.

GJR-GARCH specification. To capture asymmetric volatility responses, a GJR-GARCH model is also estimated. This specification allows negative return shocks to have a different effect on future volatility than positive shocks of the same magnitude. The GJR-GARCH(1,1) model is given by

$$\sigma_t^2 = \omega + \alpha \tilde{r}_{t-1}^2 + \gamma I(\tilde{r}_{t-1} < 0) \tilde{r}_{t-1}^2 + \beta \sigma_{t-1}^2, \tag{12}$$

where $I(\tilde{r}_{t-1} < 0)$ is an indicator for negative return shocks. Under this convention, $\gamma > 0$ implies that negative shocks have an additional positive effect on future conditional variance. This feature is relevant for equity index volatility because downside market moves are often associated with stronger subsequent volatility responses.

For the one-step-ahead forecast, the sign of the observed current return is known and is therefore used directly:

$$E_t[\sigma_{t+1}^2] = \omega + \alpha\tilde{r}_t^2 + \gamma I(\tilde{r}_t < 0)\tilde{r}_t^2 + \beta\sigma_t^2. \tag{13}$$

For horizons beyond one day, assuming symmetric standardized innovations,

$$E\left[I(\tilde{r}_{t+k} < 0)\tilde{r}_{t+k}^2\right] = \frac{1}{2}E_t[\sigma_{t+k}^2]. \tag{14}$$

Therefore, the effective multi-step persistence term becomes

$$\phi_{\text{GJR}} = \alpha + \beta + \frac{\gamma}{2}. \tag{15}$$

EGARCH specification. As an additional asymmetric volatility model, an EGARCH specification is estimated. Unlike standard GARCH and GJR-GARCH, EGARCH models the logarithm of conditional variance:

$$\log\sigma_t^2 = \omega + \alpha\left(|z_{t-1}| - E|z_{t-1}|\right) + \gamma z_{t-1} + \beta\log\sigma_{t-1}^2, \tag{16}$$

where

$$z_t = \frac{\tilde{r}_t}{\sigma_t}. \tag{17}$$

The parameter α captures the magnitude effect of return shocks, β controls persistence in log variance, and γ captures the asymmetric leverage effect. For equity index returns, a negative value of γ indicates that negative shocks increase future volatility more than positive shocks of the same magnitude.

The log-variance formulation guarantees positive conditional variance without requiring non-negativity constraints on the variance equation. However, it also complicates multi-step forecasting because the model evolves in $\log\sigma_t^2$, while the desired forecast is in variance units. Therefore, EGARCH multi-step forecasts are generated using Monte Carlo simulation. For each forecast date, $N = 500$ paths of standardized future innovations are simulated, and the log-variance equation is propagated forward over the five-day horizon. For simulated path n, the average future variance is

$$\widehat{RV}_{t,t+H}^{(n)} = \frac{1}{H}\sum_{k=1}^{H}\exp\left(\log\sigma_{t+k}^{2,(n)}\right). \tag{18}$$

The EGARCH forecast is obtained by aggregating across simulated paths:

$$\widehat{RV}_{t,t+H}^{\text{EGARCH}} = \text{Agg}\left(\widehat{RV}_{t,t+H}^{(1)}, \ldots, \widehat{RV}_{t,t+H}^{(N)}\right). \tag{19}$$

Both mean and median aggregation are considered. Median aggregation is included as a robustness check because the exponential transformation from log variance to variance can make the mean forecast sensitive to extreme simulated paths, creating upward bias through Jensen's inequality.

Five-day volatility forecast construction. For all GARCH-family models, the final object of interest is the expected average variance over the next five trading days:

$$\widehat{V}_{t,t+H} = \frac{1}{H}\sum_{k=1}^{H} E_t[\sigma^2_{t+k}]. \tag{20}$$

For standard GARCH and GJR-GARCH models, this quantity is computed using analytical multi-step variance recursions. For EGARCH models, it is computed using the simulation-based procedure described above.

Since the models are estimated using percentage returns, the annualized volatility forecast is converted back into raw-return units:

$$\widehat{\sigma}^{\text{ann}}_{t,t+H} = \sqrt{252 \times \frac{\widehat{V}_{t,t+H}}{100^2}}. \tag{21}$$

This transformation ensures that the GARCH-family forecasts are expressed in the same units as both the realized volatility target and the option-implied volatility used in the trading signal.

Deployment-oriented fixed-parameter implementation. The live QuantConnect algorithm environment does not support real-time re-estimation of GARCH-family models using the `arch` package. To make the volatility models deployable, model estimation and live updating are separated. Model parameters, including ω, α, β, γ, and ν, are estimated in the QuantConnect Research environment and then passed to the algorithm.

During backtesting or live trading, the algorithm does not perform maximum likelihood estimation. Instead, it updates the conditional variance state recursively using the fixed estimated parameters and the newly observed daily return. This structure is consistent with a production deployment setting: the research environment performs parameter estimation, while the trading algorithm performs online filtering and forecasting.

The full-period calibration used for deployment is therefore interpreted as a deployment-oriented parameter calibration rather than as a purely held-out out-of-sample model selection exercise. Its purpose is to obtain stable parameters that can be used by the live algorithm under QuantConnect's technical constraints.

Benchmark and evaluation metrics. Forecast accuracy is evaluated against the five-day forward realized-volatility target. This target is constructed as the square root of the annualized average realized variance over the next $H = 5$ trading days. The five-day horizon is chosen because it matches the short-horizon nature of the options trading signal: the strategy is not intended to forecast long-run volatility, but rather to identify near-term deviations between expected realized volatility and option implied volatility.

As a transparent benchmark, we use the 20-day historical volatility measure:

$$HV20_t = \sqrt{252 \times \frac{1}{20}\sum_{j=0}^{19} r^2_{t-j}}. \tag{22}$$

This benchmark is intentionally simple. It uses only information available up to time t, is directly implementable in the QuantConnect algorithm, and provides a meaningful comparison because it represents a standard moving-average volatility estimate. Therefore, a GARCH-family model should not only fit the data statistically, but should also demonstrate whether it improves upon this simple and deployable volatility filter.

The primary loss function is QLIKE, computed on the variance scale. Although forecast accuracy is reported in annualized-volatility units for interpretability, QLIKE is defined in terms of realized and predicted variances. Let

$$RV^{var}_{t,t+H} = 252 \times \frac{1}{H} \sum_{j=1}^{H} r^2_{t+j} \tag{23}$$

denote the annualized realized variance over the next H trading days. Equivalently, $RV^{var}_{t,t+H} = RV^2_{t,t+H}$, where $RV_{t,t+H}$ is the annualized realized volatility defined in Section 3.2.

For the GARCH-family models, the model-implied annualized variance forecast is

$$\hat{v}_{t,t+H} = 252 \times \frac{\widehat{V}_{t,t+H}}{100^2}, \tag{24}$$

where $\widehat{V}_{t,t+H}$ is the predicted average conditional variance over the next H trading days in percentage-return squared units. The QLIKE loss is then defined as

$$QLIKE_t = \frac{RV^{var}_{t,t+H}}{\hat{v}_{t,t+H}} - \log\left(\frac{RV^{var}_{t,t+H}}{\hat{v}_{t,t+H}}\right) - 1. \tag{25}$$

QLIKE is used because volatility is latent and the realized-volatility target is itself a noisy proxy for the true conditional variance. Compared with a pure squared-error loss, QLIKE is less likely to reward models that simply overfit large realized-volatility spikes. This is important in the present application because volatility forecasts are used for option trading, where both underprediction during stress regimes and systematic overprediction during calm regimes can lead to poor trading signals.

In addition to QLIKE, we report mean squared error:

$$MSE = \frac{1}{T} \sum_{t=1}^{T} \left(\sigma^{\text{realized}}_{t,t+H} - \widehat{\sigma}_{t,t+H}\right)^2. \tag{26}$$

MSE is included as a secondary diagnostic because it is easy to interpret in annualized-volatility units. However, it is not used alone for model selection, since squared-error measures can be disproportionately influenced by a small number of volatility spike observations.

To evaluate whether a model improves upon the benchmark, we compute the benchmark-relative out-of-sample R^2:

$$R^2_{\text{vs HV20}} = 1 - \frac{\sum_t \left(\sigma^{\text{realized}}_{t,t+H} - \widehat{\sigma}_{t,t+H}\right)^2}{\sum_t \left(\sigma^{\text{realized}}_{t,t+H} - HV20_t\right)^2}. \tag{27}$$

A positive value of $R^2_{\text{vs HV20}}$ means that the model reduces squared forecast error relative to the 20-day historical-volatility benchmark. A negative value means that the model is less accurate

than the benchmark over that evaluation period. This relative measure is more informative than a standalone R^2, because the practical question is not whether the model fits volatility in isolation, but whether it adds value beyond a simple volatility estimate that could already be used in live trading.

Out-of-sample diagnostic comparison. The GARCH-family specifications are first compared across two held-out out-of-sample periods. These tests are used as diagnostic evidence rather than as a mechanical rule for selecting the final deployed model within GARCH family. This distinction is important because the two out-of-sample periods represent different market regimes. OOS A is a relatively more normal volatility environment, whereas OOS B contains a more pronounced volatility spike. A model that performs well in one regime may not necessarily generalize to the other.

Table 1: Out-of-sample diagnostic comparison of GARCH-family volatility forecasts.

Model	**OOS A**	**OOS B**
	R^2_{OOS} **relative to HV20**	
GARCH(1,1)-normal	-0.4589	0.2936
GARCH(1,1)-t	-0.5515	0.2122
EGARCH(1,1)-t, mean aggregation	-1.6583	0.5171
EGARCH(1,1)-t, median aggregation	-0.7694	0.5188
EGARCH(1,1)-normal	-0.9019	0.5391
GJR-GARCH(1,1)-normal	-0.6080	0.4478
GJR-GARCH(1,1)-t	-0.6146	0.4529

Table 1 shows that the asymmetric models perform better during OOS B. This is economically plausible: when equity markets sell off, negative return shocks often lead to a stronger increase in subsequent volatility than positive shocks of the same magnitude. EGARCH and GJR-GARCH are designed to capture this asymmetry through leverage terms, so their stronger performance in the heightened-volatility regime is consistent with their model structure.

However, the same asymmetric models perform poorly in OOS A. This is a critical finding. It suggests that the additional leverage parameters improve forecasts mainly in stress-like environments, but may introduce upward bias or unnecessary variance during more stable regimes. Therefore, selecting the model with the highest OOS B score alone would create a regime-specific model-selection problem. From the perspective of live trading, the future regime is not known in advance, so a model that performs well only after a volatility spike has already occurred may not be the most reliable deployment choice.

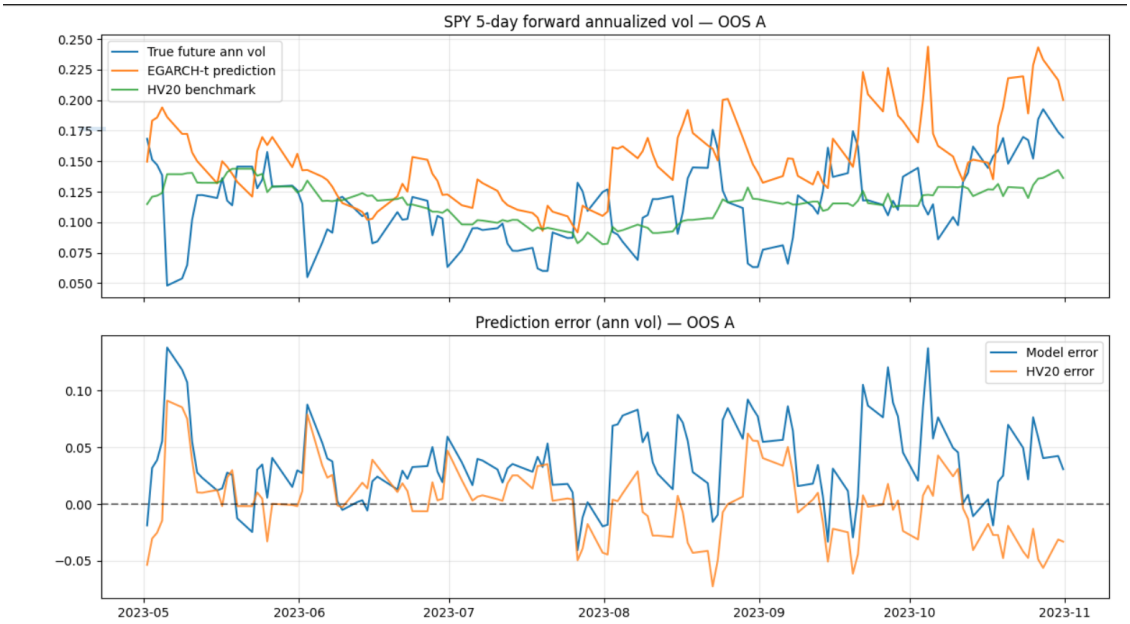

Figure 2: EGARCH(1,1)-t volatility forecasts and prediction errors in OOS A. The model exhibits persistent upward bias in predicted volatility and larger forecast dispersion relative to the HV20 benchmark.

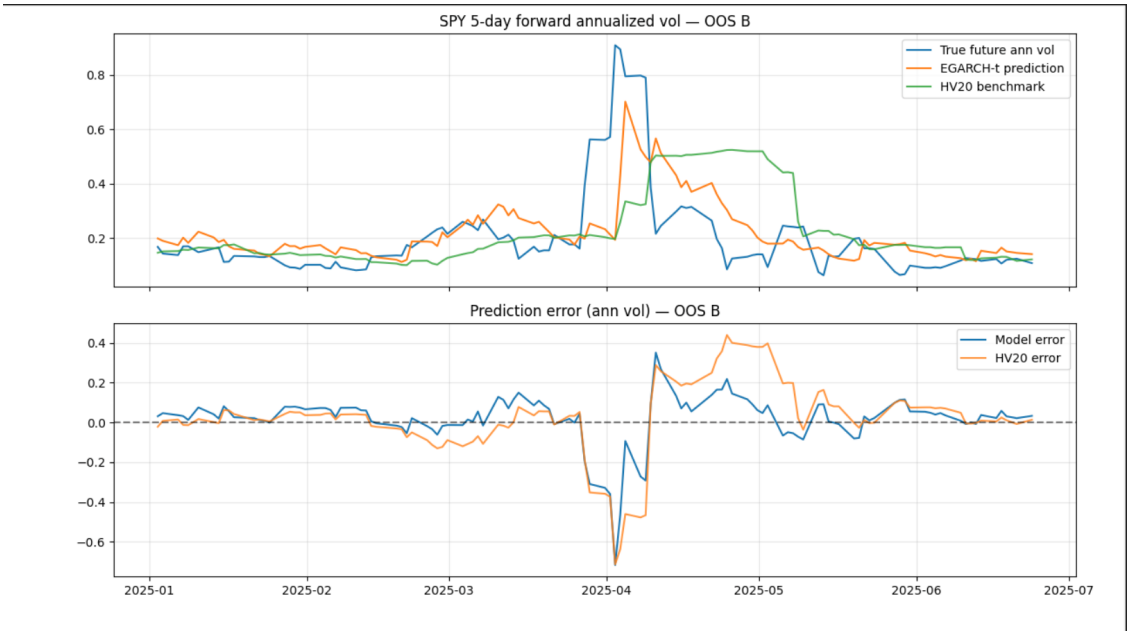

Figure 3: EGARCH(1,1)-t volatility forecasts and prediction errors in OOS B. The model captures volatility spikes more effectively, but exhibits increased instability and overshooting during regime transitions.

Visual diagnostic of EGARCH forecasts. Figures 2 and 3 provide a visual comparison of EGARCH(1,1) forecasts against realized volatility and the HV20 benchmark. These figures help clarify the model behavior underlying the numerical results in Table 1.

In OOS A, which corresponds to a relatively stable volatility regime, the EGARCH model tends to produce persistently elevated volatility forecasts relative to realized outcomes. This results in a systematic positive bias and increased forecast variance, as reflected in the larger prediction errors. While the model captures volatility clustering, the leverage mechanism appears to introduce unnecessary amplification in periods without strong directional shocks.

In contrast, OOS B contains a pronounced volatility spike, during which EGARCH responds more aggressively than symmetric models. As shown in Figure 3, the model captures the initial volatility surge more effectively than the HV20 benchmark, leading to improved performance in this regime. However, this responsiveness comes at the cost of instability: the model exhibits overshooting behavior and slower mean reversion following the spike, resulting in larger forecast errors during the normalization phase.

These visual diagnostics reinforce the key trade-off observed in the quantitative results. EGARCH provides better performance in stress-like environments due to its ability to model asymmetric volatility responses, but this same feature leads to instability and bias in more typical market conditions.

Deployment-oriented fixed-parameter implementation. The final implementation must also satisfy the practical constraints of the QuantConnect environment. The `arch` package is used in the Research environment for parameter estimation, but the live algorithm does not re-estimate GARCH-family models through maximum likelihood. Instead, the estimated parameters are fixed and the algorithm updates the conditional variance recursively as new daily returns arrive.

This separation between estimation and online filtering is deliberate. In the Research environment, the model can use a sufficiently long historical sample to obtain stable parameter estimates. In the live algorithm, the objective is not to refit the model every day, but to update the volatility state using the already-estimated recursion. This structure reduces computational overhead, avoids live-estimation instability, and makes the model more consistent with a production trading workflow.

The deployed calibration is therefore interpreted differently from the out-of-sample diagnostic exercise. The out-of-sample results in Table 1 are used to understand model behavior across regimes. The final deployment calibration, by contrast, is a fixed-parameter volatility filter calibrated on the available historical sample for live implementation. This distinction avoids overstating the full-sample diagnostic results as a pure held-out model-selection result.

Final GARCH-family model selection. The final deployed GARCH-family model is the GARCH(1,1)-normal specification. This choice is not based on the claim that it has the highest out-of-sample forecast accuracy in every regime. Instead, it is selected because it provides the best balance among forecast performance, numerical stability, parameter parsimony, and QC implementability.

The first reason is analytical tractability. For GARCH(1,1), the multi-step variance forecast has a

closed-form recursive structure:

$$E_t[\sigma^2_{t+k}] = \omega + (\alpha + \beta)E_t[\sigma^2_{t+k-1}], \qquad k \geq 2. \tag{28}$$

This allows the algorithm to compute five-day volatility forecasts quickly and deterministically. By contrast, EGARCH evolves in log variance and requires simulation-based aggregation to obtain forecasts in variance units. Although EGARCH can capture leverage effects more explicitly, the simulation step adds computational complexity and introduces additional implementation risk in live trading.

The second reason is parameter parsimony. GARCH(1,1)-normal requires only three variance parameters, ω, α, and β. GJR-GARCH adds a leverage parameter, while Student-t specifications add a degrees-of-freedom parameter. These additional parameters can improve regime-specific fit, but they also increase estimation sensitivity. In this study, the GJR-GARCH-t specification is close to the integrated-GARCH boundary, which weakens the stability of its long-run variance and makes it less attractive for deployment.

The third reason is deployment stability rather than uniformly superior out-of-sample accuracy. The asymmetric specifications perform better in OOS B, but their poor OOS A results indicate that their advantage is regime-specific. The baseline GARCH(1,1)-normal model is less specialized: it captures volatility clustering without making the live strategy overly dependent on a particular stress-regime pattern. Therefore, the model is selected as a stable volatility filter, not as the specification with the best forecast accuracy in every market environment.

Table 2: Full-sample diagnostic evaluation of the deployed GARCH(1,1)-normal calibration.

Model	**MSE**	**RMSE**	**MAE**	**Standard R^2**	$\boldsymbol{R^2_{\text{vs HV20}}}$
GARCH(1,1)-normal	0.006713	0.081936	0.055011	0.441438	0.239672

The full-sample diagnostic in Table 2 provides a final implementation check. The deployed GARCH(1,1)-normal calibration achieves a positive $R^2_{\text{vs HV20}}$ of 0.2397, indicating that it reduces squared forecast error relative to the HV20 benchmark over the diagnostic sample. This does not imply that GARCH(1,1)-normal dominates every alternative in every market regime. Rather, it supports the more limited claim that, under the deployment calibration, the selected model provides a simple and implementable volatility filter that improves upon the HV20 benchmark over the diagnostic sample. Its advantage should therefore be interpreted as implementation-oriented rather than as evidence of uniformly superior out-of-sample forecasting performance.

Overall, the model-selection decision follows a deployment-oriented principle: the volatility input should be accurate enough to improve on a simple benchmark, but also simple enough to update reliably in real time. Under this criterion, GARCH(1,1)-normal is preferred as the primary volatility forecasting component, while EGARCH and GJR-GARCH are retained as diagnostic robustness checks showing how asymmetric models behave under different market regimes.

3.4.2 Random Forest Regressor

To complement the parametric GARCH-family models, we also implement a non-parametric machine learning benchmark based on the Random Forest (RF) regressor. Unlike GARCH models,

which specify an explicit conditional variance equation, the RF model learns the forecasting relationship directly from lagged market information. This allows the model to capture nonlinearities and interaction effects that may not be well represented by a fixed parametric volatility specification.

Model Specification. Let x_t denote the feature vector observed at the close of trading day t. All predictors are constructed using information available no later than time t. The RF regressor consists of an ensemble of regression trees, and its forecast is obtained by averaging the predictions across trees:

$$\hat{\sigma}^{RF}_{t,t+5} = \frac{1}{B}\sum_{b=1}^{B} T_b(x_t;\Theta_b), \tag{29}$$

where $T_b(\cdot)$ denotes the b-th regression tree and Θ_b represents its fitted structure and terminal-node values. The averaging procedure reduces the variance of individual trees and produces a more stable volatility forecast.

Feature Engineering. The feature vector x_t summarizes the recent market state using lagged and contemporaneously observable variables. These include realized-volatility measures, recent return and trend indicators, downside-risk and drawdown variables, trading-volume information, and VIX-based volatility proxies. The VIX-related variables are used only as explanatory market-state features in the forecasting model. In the trading stage, the RF forecast is compared with option-chain implied volatility rather than VIX itself.

Due to the very limited computing power on QuantConnect coding server, Random Forest inferences are very slow at runtime, especially for in-sample period. This makes it unconducive to conduct further analysis on Random Forest for the purpose of this research.

3.4.3 LSTM-Based Volatility Forecasting

Model Formulation. The LSTM model is designed to forecast short-horizon realized volatility using a sequence of historical observations. Let $X_t \in \mathbb{R}^d$ denote the feature vector observed at time t, and define a lookback window of length L. The input to the model is a sequence of past observations:

$$\mathbf{X}_t = \{X_{t-L+1}, X_{t-L+2}, \ldots, X_t\}.$$

The forecasting problem is formulated as a supervised learning task:

$$\hat{RV}_{t,t+H} = f_\theta(\mathbf{X}_t),$$

where $\hat{RV}_{t,t+H}$ denotes the predicted realized volatility over the next H trading days, and $f_\theta(\cdot)$ represents the nonlinear mapping learned by the LSTM network with parameters θ.

The LSTM processes the input sequence recursively. At each time step τ, the hidden state h_τ and cell state c_τ are updated as:

$$h_\tau, c_\tau = \mathrm{LSTM}(X_\tau, h_{\tau-1}, c_{\tau-1}),$$

where the LSTM cell internally uses gated mechanisms (input, forget, and output gates) to control the flow of information across time.

After processing the full sequence, the final hidden state h_t summarizes the relevant temporal information:

$$h_t = \text{LSTM}(\mathbf{X}_t).$$

The predicted volatility is then obtained through a fully connected output layer:

$$\hat{RV}_{t,t+H} = g(h_t),$$

where $g(\cdot)$ is a nonlinear transformation (e.g., a linear layer followed by a Softplus activation) that ensures the output remains positive.

This formulation allows the model to learn both temporal dependencies and nonlinear feature interactions directly from the data, without imposing a fixed parametric structure on volatility dynamics.

Feature Engineering. The feature engineering process captures key drivers of short-horizon volatility dynamics, including return shocks, volatility persistence, market regime changes, and nonlinear transformations of recent price behavior.

We begin with a baseline set of features commonly used in volatility forecasting. These include daily returns r_t, absolute returns $|r_t|$, and historical volatility measures computed over multiple horizons. Specifically, realized volatility is estimated using rolling standard deviations of log returns:

$$HV_t^{(k)} = \sqrt{252} \cdot \text{std}(r_{t-k+1}, \ldots, r_t),$$

for $k \in \{5, 10, 21\}$. These features capture short-, medium-, and longer-term volatility persistence, which is a well-documented property of financial time series.

In addition, we include a simple trend-related feature defined as the deviation of the current price from its moving average:

$$\text{sma\_dist}_{21,t} = \frac{P_t}{\text{SMA}_{21,t}} - 1,$$

which provides information about recent market direction and potential mean-reversion effects.

The final selected feature set is:

$$X_t = \{r_t, |r_t|, HV_t^{(5)}, HV_t^{(10)}, HV_t^{(21)}, \text{sma\_dist}_{21,t}\}.$$

This finding suggests that short-horizon volatility forecasting is primarily driven by persistent volatility structure rather than highly reactive or complex nonlinear transformations. While additional features can capture rapid changes and regime shifts, they also introduce noise and increase model variance, particularly in high-capacity models such as LSTMs.

From a practical perspective, the selected feature set provides a balance between predictive power, robustness, and stability across different market regimes. It captures the essential dynamics of volatility clustering while avoiding overfitting and excessive sensitivity to transient market fluctuations.

Table 3: Model Comparison Across Specifications and Data Splits

Model	Split	RMSE	MAE	R^2
Naive HV(5)	Validation	0.060113	0.050386	-0.646378
Naive HV(5)	OOS A	0.037662	0.030205	-0.436034
Naive HV(5)	OOS B	0.198994	0.115311	-0.309903
Univariate LSTM (HV(5))	Validation	0.047552	0.040592	-0.030215
Univariate LSTM (HV(5))	OOS A	0.032284	0.026384	-0.055175
Univariate LSTM (HV(5))	OOS B	0.183605	0.106316	-0.115140
Univariate LSTM (RV(1d))	Validation	0.050933	0.043013	-0.181901
Univariate LSTM (RV(1d))	OOS A	0.032749	0.026882	-0.085812
Univariate LSTM (RV(1d))	OOS B	0.173098	0.097265	0.008835
Multivariate LSTM	Validation	0.044976	0.037010	0.078361
Multivariate LSTM	OOS A	0.028944	0.022627	0.151854
Multivariate LSTM	OOS B	0.162182	0.093120	0.129913

Table 4: Feature Set Comparison Across Data Splits (R^2)

Feature Set	OOS A	OOS B	Train	Validation
A_price_plus_volume	-0.265761	0.070595	0.611064	-0.069428
B_price_plus_raw_vix	-0.085379	0.058277	0.558512	-0.080293
C_price_plus_vix_features	-0.265303	-0.021475	0.716062	0.022467
D_vix_heavy	-0.220216	0.170303	0.558576	-0.211456
E_fast_reactive	-0.311361	-0.036302	0.673211	-0.087201

Model Architecture and Training Procedure. The final LSTM specification uses the selected multivariate feature set

$$X_t = \{r_t, |r_t|, HV_t^{(5)}, HV_t^{(10)}, HV_t^{(21)}, \text{sma\_dist}_{21,t}\}.$$

For each forecast date t, the model receives a rolling lookback window of $L = 20$ trading days:

$$\mathbf{X}_t = \{X_{t-19}, X_{t-18}, \ldots, X_t\}.$$

Thus, the input to the LSTM is a sequence of length 20, where each time step contains six explanatory variables.

The LSTM network consists of one recurrent LSTM layer with hidden dimension 32. A dropout rate of 0.1 is applied during training to reduce overfitting and improve generalization. After the LSTM processes the full input sequence, the final hidden representation is passed through a fully connected output layer to generate the predicted five-day forward realized volatility:

$$\hat{RV}_{t,t+5} = Wh_t + b,$$

where $h_t \in \mathbb{R}^{32}$ is the final hidden state of the LSTM.

The model is trained as a supervised regression problem using mean squared error as the loss

function:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{t=1}^{N} \left(\hat{RV}_{t,t+5} - RV_{t,t+5} \right)^2 .$$

Visual Diagnostics of LSTM In-sample fit. Figure 4 shows the model performance over the training period. The LSTM is able to closely track the overall level and clustering behavior of realized volatility. In particular, it successfully captures major volatility regimes, including extended periods of low volatility and large spikes such as the market dislocation around 2020.

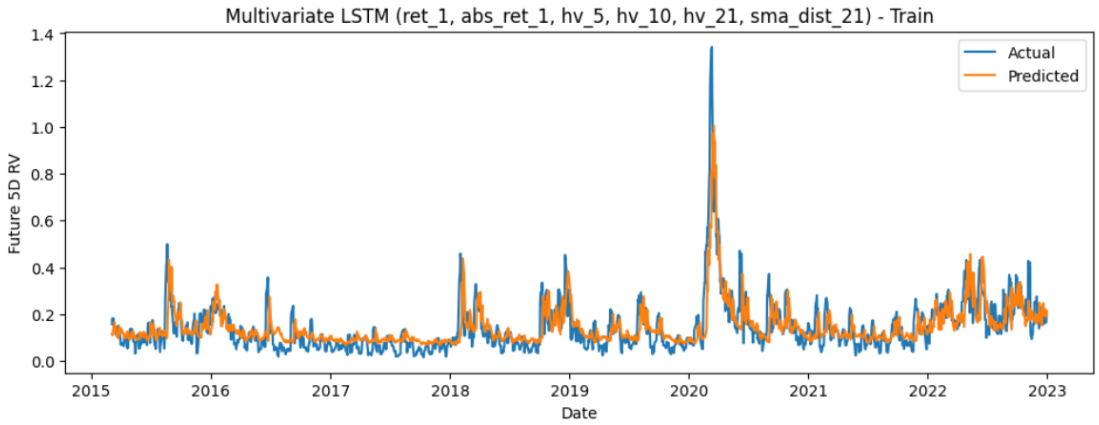

Figure 4: Multivariate LSTM predictions vs actual realized volatility (training period).

Visual Diagnostics of LSTM Out-of-sample performance (OOS A). Figure 5 presents the model performance over the OOS A period, which corresponds to a relatively stable volatility regime. The model continues to capture the general level and direction of volatility, and the predicted series broadly follows the trend of realized volatility. However, at a shorter time scale, the model exhibits noticeable lag.

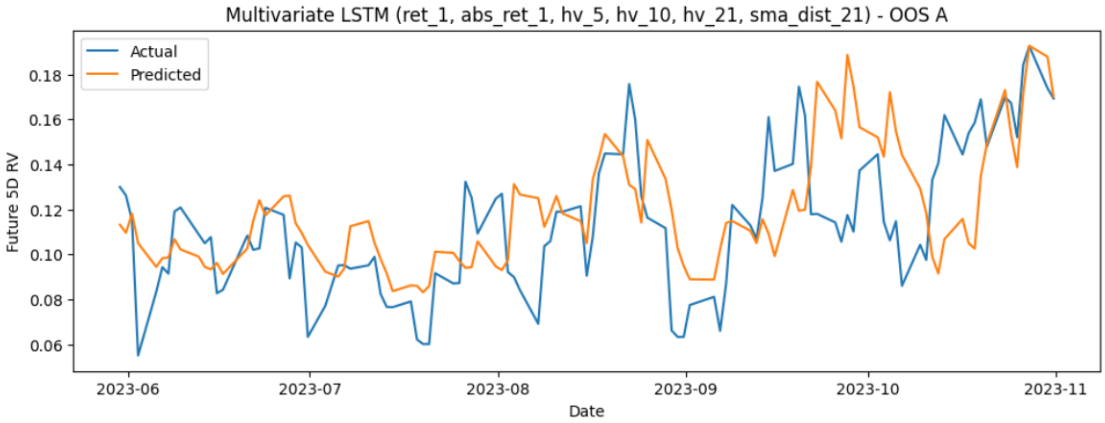

Figure 5: Multivariate LSTM predictions vs actual realized volatility (OOS A).

Visual Diagnostics of LSTM Out-of-sample performance (OOS B). Figure 6 illustrates model behavior during the OOS B period, which contains a pronounced volatility spike. While the LSTM captures the general upward movement in volatility, it significantly underestimates the magnitude of the spike and responds with delay.

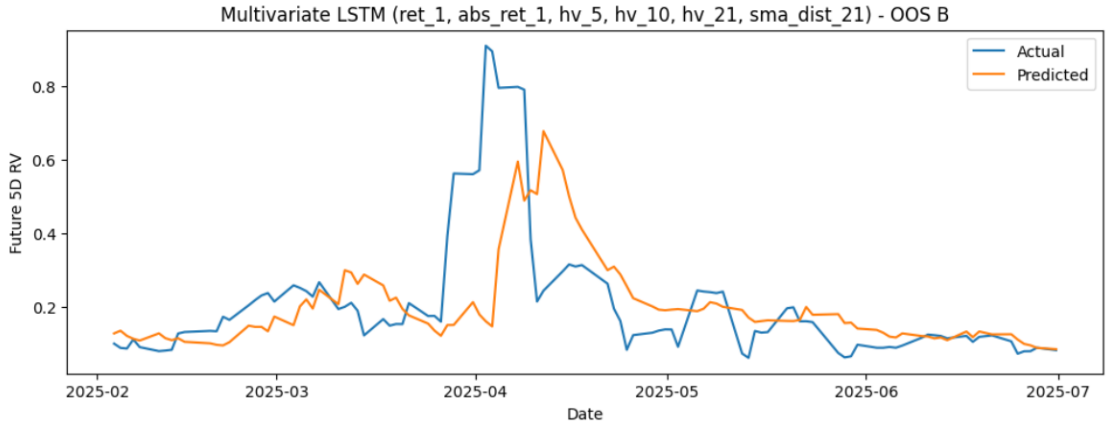

Figure 6: Multivariate LSTM predictions vs actual realized volatility (OOS B).

Discussion. Overall, the visual evidence suggests that the multivariate LSTM captures the long-term structure and clustering behavior of volatility, consistent with its positive out-of-sample R^2 performance. However, at shorter horizons, the model exhibits systematic lag and smoothing effects.

From a trading perspective, this lag is particularly important. Since the strategy relies on detecting short-horizon discrepancies between predicted and implied volatility, delayed responses can weaken signal timing and reduce the effectiveness of volatility trades, especially during periods of rapid market adjustment.

While this idea undoubtedly has great potential, it eventually does not show great compatibility to our trading strategy and is therefore not included in the result section of this paper. We believe that more careful engineering of LSTM forecast is needed to translate into reliable trading signal.

3.4.4 Feedforward Neural Network for Realized Volatility Forecasting

To complement both parametric econometric models and tree-based machine learning approaches, we implement a deep feedforward neural network (multi-layer perceptron, MLP) to forecast short-horizon realized volatility. The motivation for this model is to capture nonlinear relationships between option market microstructure variables, historical volatility dynamics, and macro-volatility proxies that may not be adequately represented in traditional models.

Model Objective. The neural network is designed to forecast the five-day forward realized volatility, denoted by $\widehat{RV}_{t,t+5}$, using information available at time t. The forecasting problem is formulated as a supervised regression task:

$$\widehat{RV}_{t,t+5} = f_\theta(X_t), \tag{30}$$

where $X_t \in \mathbb{R}^d$ represents the feature vector constructed from market data, and $f_\theta(\cdot)$ denotes a nonlinear function parameterized by neural network weights θ.

Target Construction. The prediction target is the annualized five-day forward realized volatility:

$$RV_{t,t+5} = \sqrt{252 \cdot \frac{1}{5} \sum_{j=1}^{5} r_{t+j}^2}, \tag{31}$$

where $r_t = \log(P_t/P_{t-1})$ denotes daily log returns of SPY. Importantly, the target is constructed by shifting backward the realized volatility computed over the next five trading days:

$$y_t = RV_{t,t+5}. \tag{32}$$

This ensures that all predictors in X_t are strictly observable at time t, avoiding any look-ahead bias.

Feature Engineering. The feature set combines option market characteristics, historical volatility measures, and macro-volatility indicators. Specifically, the input vector X_t includes:

- **Option microstructure features:** bid-ask spread, trading volume, moneyness, and days-to-expiration (DTE), constructed from near-the-money call options with maturities between 10 and 60 days.
- **Volatility history:** lagged realized volatility measures ($RV^{(5)}$, $RV^{(20)}$, $RV^{(60)}$), capturing short-, medium-, and long-term volatility persistence.
- **Lagged dynamics:** lagged spreads and lagged realized volatility values ($RV_{t-1}^{(5)}$, $RV_{t-5}^{(5)}$), capturing temporal dependence.
- **Macro-volatility proxy:** VIX level and daily percentage change in VIX, representing market-implied aggregate volatility expectations.

The inclusion of option-specific variables is motivated by the hypothesis that option market liquidity and pricing frictions contain forward-looking information about volatility. For example, wider bid-ask spreads may reflect heightened uncertainty or reduced liquidity, both of which are associated with elevated future volatility.

Data Preprocessing. All input features are standardized using a z-score transformation:

$$\tilde{X}_{t,i} = \frac{X_{t,i} - \mu_i}{\sigma_i}, \tag{33}$$

where μ_i and σ_i are the mean and standard deviation estimated from the in-sample (training) period only. This normalization ensures numerical stability during optimization and prevents features with larger magnitudes from dominating the learning process.

Model Architecture. The neural network is a fully connected feedforward architecture with residual connections. Let d denote the input dimension. The model consists of the following layers:

$$h_1 = \mathrm{ELU}(W_1 x + b_1), \quad W_1 \in \mathbb{R}^{256 \times d} \tag{34}$$
$$h_2 = \mathrm{ELU}(W_2 h_1 + b_2), \quad W_2 \in \mathbb{R}^{128 \times 256} \tag{35}$$
$$h_3 = \mathrm{ELU}(W_3 h_2 + b_3), \quad W_3 \in \mathbb{R}^{128 \times 128} \tag{36}$$
$$h_4 = \mathrm{ELU}(W_4 h_3 + b_4), \quad W_4 \in \mathbb{R}^{64 \times 128} \tag{37}$$

A residual (skip) connection projects the input directly into the final hidden representation:

$$h_{\mathrm{res}} = W_{\mathrm{skip}} x + b_{\mathrm{skip}}, \quad W_{\mathrm{skip}} \in \mathbb{R}^{64 \times d} \tag{38}$$

The final hidden state is obtained by combining the two pathways:

$$h = h_4 + h_{\mathrm{res}}. \tag{39}$$

The output layer is given by:

$$\hat{y} = \mathrm{Softplus}(W_5 h + b_5), \tag{40}$$

where the Softplus activation ensures that predicted volatility is strictly positive, consistent with the definition of realized volatility.

Design Rationale. Several architectural choices are made deliberately:

- **ELU activation:** ELU is used instead of ReLU to avoid dead neuron problems and to provide smoother gradients, which improves optimization stability in financial time series.
- **Residual connection:** The skip connection allows the model to preserve linear relationships between inputs and output while learning nonlinear corrections. This is particularly important given that volatility dynamics often contain strong linear persistence components.
- **Softplus output:** Enforces positivity without imposing hard constraints, avoiding instability associated with unconstrained outputs.

Training Procedure. The model is trained using the Adam optimizer with learning rate $\eta = 10^{-3}$:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta), \tag{41}$$

where the loss function is the mean squared error (MSE):

$$\mathcal{L} = \frac{1}{N} \sum_{t=1}^{N} (\hat{y}_t - y_t)^2 . \tag{42}$$

Unlike the GARCH framework where QLIKE is preferred, MSE is intentionally used here to penalize large prediction errors more aggressively. This choice reflects the objective of forcing the network to closely fit the realized volatility surface, especially during high-volatility regimes.

The model is trained for 300 epochs with mini-batch size 8. No explicit dropout layer is applied in the baseline specification, although model generalization is evaluated through separate out-of-sample and stress-test periods. This design allows the network to serve as a high-capacity nonlinear benchmark. However, because the model contains substantially more flexibility than the GARCH specification, its results should be interpreted cautiously and assessed primarily through out-of-sample performance rather than in-sample fit.

Overfitting Considerations. The training configuration intentionally allows for high model capacity and minimal regularization. This design serves two purposes:

1. To evaluate the upper bound of predictive performance achievable with the given feature set.
2. To test whether richer nonlinear representations can extract signal from option microstructure variables beyond what is captured by econometric models.

However, this approach introduces the risk of overfitting. To assess generalization, the model is evaluated on multiple out-of-sample periods (OOS A and OOS B), representing different volatility regimes.

Economic Interpretation. The neural network can be interpreted as a flexible nonlinear mapping from market state variables to expected volatility. Unlike GARCH models, which impose a specific parametric structure, the neural network allows interactions between:

- option liquidity conditions (spreads, volume),
- volatility persistence (lagged RV),
- and macro expectations (VIX).

This flexibility enables the model to adapt across regimes, potentially capturing effects such as liquidity-driven volatility amplification and nonlinear responses to market stress.

Overall, the neural network serves as a high-capacity benchmark that tests whether additional predictive power can be extracted from the joint dynamics of option markets and underlying asset volatility.

3.5 Trading Strategy

Building on the neural-network volatility forecasting framework developed in the previous section, we construct a systematic options trading strategy that exploits discrepancies between model-forecasted realized volatility and market-implied volatility. The strategy translates the forecasted

short-horizon volatility into a directional volatility exposure using liquid SPY option contracts. Rather than forecasting the direction of the underlying index, the strategy focuses on whether the option market appears to overprice or underprice future volatility relative to the model forecast.

The core economic intuition is straightforward. If the model predicts that future realized volatility will exceed the volatility level implied by option prices, options are interpreted as relatively underpriced and the strategy takes a long-volatility position. Conversely, if the model forecast is sufficiently below implied volatility, options are interpreted as relatively expensive and the strategy takes a short-volatility position. The implementation therefore forms a two-sided volatility relative-value strategy.

3.5.1 Signal Construction

At each decision time t, the neural network produces a forecast of annualized realized volatility over the next five trading days, denoted by

$$\widehat{RV}_{t,t+5}^{NN}.$$

This forecast is compared with the contemporaneous at-the-money implied volatility extracted from SPY options. The implied volatility input is constructed by identifying the strike closest to the current SPY price and averaging the implied volatility of the corresponding call and put contracts:

$$IV_t^{ATM} = \frac{IV_t^{call} + IV_t^{put}}{2}. \tag{43}$$

The main trading signal is a normalized volatility edge:

$$\text{VolEdge}_t = \frac{\widehat{RV}_{t,t+5}^{NN} - IV_t^{ATM}}{IV_t^{ATM}}. \tag{44}$$

This normalization is important because the economic meaning of a fixed volatility-point difference depends on the prevailing volatility regime. For example, a two-volatility-point difference is more meaningful when implied volatility is 10% than when it is 40%. The relative signal therefore makes the entry rule more comparable across low- and high-volatility environments.

A positive volatility edge indicates that the model expects future realized volatility to be higher than the level priced by the option market. This suggests that options may be relatively cheap. A negative volatility edge indicates that implied volatility is high relative to the model's forecast, suggesting that option premium may be expensive.

3.5.2 Threshold-Based Entry Rule

The strategy enters trades only when the magnitude of the volatility edge exceeds a pre-specified threshold. Specifically, the entry rule is:

$$\text{Enter long volatility if } \text{VolEdge}_t > 5\%, \tag{45}$$

and

$$\text{Enter short volatility if } \text{VolEdge}_t < -5\%. \tag{46}$$

No position is initiated when

$$-5\% \leq \text{VolEdge}_t \leq 5\%.$$

This no-trade region is designed to reduce excessive turnover and avoid entering trades when the estimated volatility mispricing is too small to reliably overcome bid–ask spreads, slippage, and model uncertainty.

The use of a relative threshold also makes the trading rule directly comparable across market regimes. In calm markets, the rule requires the model forecast to differ meaningfully from a low implied-volatility base. In stressed markets, the required absolute volatility difference becomes larger because implied volatility itself is elevated.

3.5.3 Position Direction and Volatility Exposure

The sign of the volatility edge determines the direction of the option position.

Long-volatility position. When

$$\text{VolEdge}_t > 5\%,$$

the neural network predicts realized volatility above market-implied volatility. The strategy therefore buys a strangle, consisting of one out-of-the-money call and one out-of-the-money put with the same expiration date. This position benefits from sufficiently large movements in SPY in either direction and has positive convexity. It is designed to profit when future realized volatility is higher than the level embedded in option prices.

Short-volatility position. When

$$\text{VolEdge}_t < -5\%,$$

the model forecast is below market-implied volatility. The strategy therefore sells a strangle using the same type of call and put contracts. This position collects option premium and benefits if SPY remains sufficiently range-bound, allowing realized volatility to remain below the level priced by the market.

This design creates a symmetric two-sided volatility trading framework. The long-volatility side attempts to monetize underpriced optionality, while the short-volatility side attempts to harvest expensive option premium. However, the two sides are not risk-symmetric. Long strangles have limited downside equal to the premium paid, whereas short strangles expose the strategy to nonlinear tail losses during large market moves.

3.5.4 Option Contract Selection

All trades are implemented using SPY options. SPY is selected because of its high liquidity, tight option markets, and direct connection to the broad U.S. equity market. The option universe

includes weekly options, allowing the strategy to select contracts whose maturities are reasonably aligned with the five-day volatility forecast horizon.

At each trading decision, the algorithm considers option contracts with remaining time to expiration between 7 and 30 calendar days:

$$7 \leq DTE \leq 30.$$

Within this eligible set, expirations are ranked by proximity to a target maturity of approximately 14 days. This target maturity balances two considerations. First, it is short enough for the option position to be sensitive to near-term realized volatility. Second, it avoids the most extreme gamma and liquidity risks associated with options that are only a few days from expiration.

The strategy constructs strangles using out-of-the-money call and put options. The preferred call strike range is:

$$1.05S_t \leq K_t^{call} \leq 1.12S_t, \tag{47}$$

where S_t denotes the current SPY price. The preferred put strike range is:

$$0.88S_t \leq K_t^{put} \leq 0.95S_t. \tag{48}$$

Thus, both legs are approximately 5%–12% out of the money. The strategy prioritizes contracts in these ranges because extremely far out-of-the-money options may be inexpensive but often have weak volatility sensitivity, poor liquidity, and large relative bid–ask spreads. By selecting moderately out-of-the-money contracts, the strategy aims to maintain meaningful gamma and vega exposure while avoiding the high premium cost of at-the-money straddles.

If no contract is available in the preferred strike range, the selection rule relaxes to the nearest available out-of-the-money call or put, provided the contract satisfies the liquidity filters described below. This prevents the algorithm from failing to trade solely because the preferred strike window is unavailable on a particular date.

3.5.5 Liquidity Filters

Before a contract can be selected, it must pass several liquidity filters. These filters are included to reduce execution noise and avoid contracts whose quoted prices are unreliable. A contract is considered tradable only if:

- its ask price is greater than \$0.05;
- its bid–ask spread ratio does not exceed 25%;
- its open interest is at least 5 contracts.

The bid–ask spread ratio is computed relative to the ask price:

$$\frac{Ask_t - Bid_t}{Ask_t}.$$

The spread filter is especially important for short-dated out-of-the-money options, where quoted spreads can represent a large fraction of the option price. Without this filter, apparent backtest profitability may be driven by unrealistic fills in illiquid contracts.

3.5.6 Position Sizing

Position size is determined by a fixed portfolio-budget rule. For each trade, the strategy allocates at most 15% of total portfolio value to the option structure:

$$Budget_t = 0.15 \times V_t, \tag{49}$$

where V_t is the total portfolio value at time t.

For a long strangle, the number of contracts is based on the total premium required to buy both legs:

$$q_t = \left\lfloor \frac{Budget_t}{100 \times (Ask_t^{call} + Ask_t^{put})} \right\rfloor. \tag{50}$$

For a short strangle, the analogous sizing rule uses the bid prices of the two legs:

$$q_t = \left\lfloor \frac{Budget_t}{100 \times (Bid_t^{call} + Bid_t^{put})} \right\rfloor. \tag{51}$$

In both cases, the number of contracts is capped at 500. This cap prevents the strategy from scaling excessively into very cheap options, where a fixed-budget rule could otherwise generate an unrealistically large number of contracts. The strategy therefore combines proportional capital allocation with a hard contract-level exposure limit.

Although this sizing rule controls premium exposure, it does not fully control the tail risk of short-volatility trades. In particular, the loss on a short strangle can exceed the initial premium collected. Therefore, the strategy requires explicit exit and risk-management rules.

3.5.7 Position Management and Exit Rules

The strategy manages open positions using profit-taking, stop-loss, time-based, and expiration-based exit rules. Only one tracked option structure is held at a time. If a position is already open, the algorithm first manages the existing trade and does not initiate a new one on the same day.

Stop-loss rule. A position is liquidated if the unrealized loss reaches 60% of the entry cost basis:

$$\frac{PnL_t}{EntryCost} \leq -60\%. \tag{52}$$

For long strangles, the entry cost basis corresponds to the premium paid for both legs. For short strangles, it corresponds to the initial premium collected. This rule is designed to prevent a losing volatility trade from remaining open after the original volatility view has moved substantially against the strategy.

Profit-taking rule. A position is closed when the unrealized profit reaches 40% of the entry cost basis:

$$\frac{PnL_t}{EntryCost} \geq 40\%. \tag{53}$$

This rule allows the strategy to realize gains when the volatility view materializes quickly. It is especially relevant for long-volatility trades, where option values can increase sharply after large underlying moves, and for short-volatility trades, where a portion of the premium can be captured before expiration risk becomes more concentrated.

Maximum holding period. The maximum holding period is 15 calendar days. This prevents the strategy from holding stale positions whose original volatility signal may no longer be relevant. The rule also keeps the trade horizon broadly aligned with the short-term nature of the volatility forecast and the selected option maturities.

Pre-expiration exit. All positions are liquidated when the remaining time to expiration falls to two days or less:

$$DTE \leq 2. \tag{54}$$

This rule reduces expiration-related risks, including rapidly increasing gamma exposure, declining liquidity, assignment risk, and unstable bid–ask spreads near maturity.

3.5.8 Contribution of the Strategy Design

The main contribution of this trading framework is that it links a forward-looking machine-learning volatility forecast to a transparent and implementable options strategy. Instead of relying only on historical volatility or a static volatility risk premium assumption, the strategy forms a daily estimate of whether SPY options appear expensive or cheap relative to the model-implied realized-volatility forecast.

The use of a normalized volatility edge provides a scale-adjusted trading signal that can be applied across different volatility regimes. The two-sided strangle implementation allows the strategy to express both long-volatility and short-volatility views, while liquidity filters and rule-based exits make the backtest more realistic than an unconstrained signal-only strategy.

At the same time, the strategy deliberately remains simple and interpretable. The entry rule is determined by the sign and magnitude of the volatility edge, the option contracts are selected from a transparent DTE and moneyness range, and the exit rules are fixed before evaluation. This design provides a direct empirical test of whether the neural-network volatility forecast contains economically useful information for SPY option trading.

4 Results

4.1 Trading Strategy Performance by Period

We evaluate the economic performance of each model-driven trading strategy using QuantConnect backtests. Performance is reported separately for each period to assess robustness across different market environments.

For the OOS and stress-test periods, we also report Value-at-Risk (VaR) as a downside-risk diagnostic. VaR is computed from the empirical distribution of daily strategy returns within each

evaluation period:

$$VaR_{95} = -Q_{0.05}\left(r_t^{strategy}\right), \tag{55}$$

where $Q_{0.05}$ denotes the 5th percentile and $r_t^{strategy}$ is the daily portfolio return. VaR is reported as a percentage of portfolio value. A higher VaR therefore indicates a larger potential one-day downside loss under the historical return distribution of the strategy.

In-Sample (IS): 01/01/2016 – 01/01/2022

Table 5: Strategy Performance (IS)

Model	Total Return (%)	Sharpe Ratio	Max Drawdown (%)	VaR (%)
GARCH	-9.25	-1.0	12	0.24
Neural Network	252.5	0.9	33	1.9

Out-of-Sample A (OOS A): 05/01/2023 – 11/01/2023

Table 6: Strategy Performance (OOS A)

Model	Total Return (%)	Sharpe Ratio	Max Drawdown (%)	VaR (%)
GARCH	3.86	0.1	3	0.25
Neural Network	3.0	0.4	7	1.18

Out-of-Sample B (OOS B): 01/01/2025 – 07/01/2025

Table 7: Strategy Performance (OOS B)

Model	Total Return (%)	Sharpe Ratio	Max Drawdown (%)	VaR (%)
GARCH	8.1	1.1	2	0.28
Neural Network	16.8	0.7	31	4.15

Out-of-Sample C (OOS C): 10/01/2025 – 04/01/2026

Table 8: Strategy Performance (OOS C)

Model	Total Return (%)	Sharpe Ratio	Max Drawdown (%)	VaR (%)
GARCH	-2.35	-2.2	6	0.25
Neural Network	20.9	1.6	9	2.22

These results provide a period-by-period diagnostic of how each strategy behaves under different market regimes. Rather than demonstrating uniform robustness, the results highlight regime dependence. The GARCH strategy generally exhibits smaller drawdowns and lower VaR, indicating more conservative risk exposure, but its return generation is more modest and less consistently positive across periods. In contrast, the neural-network strategy produces substantially higher returns in some periods, especially in-sample and during OOS B, but this comes with materially larger drawdowns and higher VaR, suggesting greater sensitivity to market conditions and model specification.

4.2 Stress Testing

To further evaluate robustness, we conduct stress tests during historically adverse market conditions. We evaluate the ability of each strategy to withstand extreme volatility environments. In particular, the analysis focuses on drawdown control and stability during rapid market dislocations.

COVID Market Shock: 02/01/2020 – 05/01/2020

Table 9: Stress Test Performance (COVID Shock)

Model	Total Return (%)	Sharpe Ratio	Max Drawdown (%)	VaR (%)
GARCH	-3.92	-1.6	6	0.6
Neural Network	24.7	2.4	9	2.4

Global Financial Crisis: 08/01/2008 – 02/01/2009

Table 10: Stress Test Performance (Global Financial Crisis)

Model	Total Return (%)	Sharpe Ratio	Max Drawdown (%)	VaR (%)
GARCH	-0.25	-4.9	1	0.0
Neural Network	2.9	0.5	11	0.8

4.3 Summary

Overall, the results highlight the trade-off between model complexity and robustness. Simpler models such as GARCH provide more stable and consistent results across periods, and more expressive models like MLP shows great potential for long term profitability.

It should be noted that both of the models are constructed under resource constraint. MLP can be expected to perform reliable better with higher quality training data.

From a trading perspective, consistent performance across out-of-sample and stress periods is critical. The ability to control drawdowns and maintain reasonable Sharpe ratios under adverse conditions is a key determinant of strategy viability.

5 Risk Management

Robust risk management is central to the implementation of any volatility trading strategy, particularly those that rely on model-based forecasts. Since option-based strategies involve nonlinear payoffs and exposure to tail risk, the framework incorporates multiple layers of risk control to ensure stability across market conditions and to mitigate model uncertainty.

5.1 Robustness Across Time Periods

To evaluate the stability of the strategy, performance is tested across multiple market regimes, including in-sample (IS), out-of-sample (OOS), and stress periods. In particular, the strategy is applied to historically volatile environments such as the 2008 Global Financial Crisis and the COVID-19 market shock in 2020.

Importantly, the same model parameters are used across all testing periods, without re-optimization. This constraint ensures that performance is not driven by overfitting to specific market conditions, but instead reflects the model's ability to generalize. By maintaining parameter consistency, the analysis provides a more realistic assessment of how the strategy would perform in a live trading environment.

5.2 Cross-Asset Considerations

Although the primary implementation focuses on SPY options, the design of the strategy is asset-agnostic. The use of a highly liquid underlying ensures that the observed implied volatility reflects genuine market expectations rather than noise from illiquidity.

From a risk management perspective, concentrating on index options such as SPY reduces exposure to idiosyncratic risk associated with individual stocks, including earnings announcements and firm-specific shocks. This allows the strategy to isolate aggregate market volatility and the volatility risk premium more effectively.

Future extensions may consider applying the framework to other liquid index products or asset classes to evaluate cross-sectional robustness and diversification benefits.

5.3 Threshold-Based Risk Filtering

To mitigate the impact of estimation error and market microstructure noise, the strategy employs threshold-based filters when generating trading signals. Positions are initiated only when the spread between forecasted realized volatility and implied volatility exceeds both an absolute and relative threshold.

These thresholds serve two key purposes. First, they reduce excessive trading by filtering out small, noisy signals that are unlikely to be economically meaningful. Second, they improve the risk-return profile by focusing capital on higher-confidence opportunities where the discrepancy between model forecasts and market prices is more pronounced.

In addition, the thresholds indirectly control exposure during periods of elevated uncertainty, as volatility forecasts and implied volatility tend to converge during extreme market conditions. This helps prevent over-allocation during unstable regimes.

5.4 Position-Level Risk Controls

Beyond signal generation, the strategy incorporates several position-level risk management mechanisms. These include fixed holding periods aligned with the forecasting horizon, stop-loss constraints to limit downside risk, and profit-taking rules to lock in gains.

Together, these controls ensure that risk is managed dynamically throughout the life of each trade, rather than solely at the entry stage. By combining signal filtering with disciplined exit rules, the framework aims to balance return generation with controlled exposure to volatility and tail risk.

5.5 Summary

Overall, the risk management framework operates along three dimensions: temporal robustness, structural design of the underlying asset, and disciplined signal filtering. These components work jointly to reduce model risk, improve stability across market regimes, and enhance the practical viability of the strategy in real-world trading conditions.

6 Limitations

Several limitations should be considered when interpreting the results of this study. Although the strategy provides a structured way to compare model-predicted realized volatility with market-implied volatility, its performance is still hindered by several limitations. Therefore, the results should be interpreted as evidence that the framework shows great potential, rather than as proof of a consistently profitable trading rule.

First, the strategy relies heavily on the accuracy of the realized volatility forecast. Models such as GARCH, Random Forest, LSTM, and MLP can capture volatility persistence and clustering, but short-horizon volatility remains difficult to predict. During abrupt market moves, the models are shown to react with delay. This issue is particularly profound in our project with limited access to financial data on QuantConnect, especially alternative data that are increasingly important in forecasting.

Second, model performance shows great sensitivity to the chosen sample period and market regime. Models tend to show preference for high-volatility periods and low-volatility periods depending on its engineering architecture. This regime dependence means that strong performance in one out-of-sample period does not necessarily imply robust performance in future market conditions.

Third, the position structure is simplified. The baseline strategy uses symmetric call and put positions to approximate long- or short-volatility exposure, but this does not perfectly isolate volatility risk. Even when the options are selected near the money, the position may still contain directional exposure, skew exposure, vega exposure, gamma exposure, and theta decay.

Fourth, the trade-entry thresholds are fixed throughout the backtest. Fixed thresholds improve simplicity and interpretability, but they may not be optimal across different volatility regimes. Similarly, an 8% relative spread may filter out noise in some environments but may be too restrictive or too loose in others. Future versions of the strategy could use regime-dependent thresholds based on VIX levels, realized volatility, or option liquidity conditions.

Fifth, the analysis focuses mainly on SPY options. SPY is highly liquid and economically important, which makes it a reasonable starting point, but the results may not generalize to individual equity options, sector ETFs, futures options, or other asset classes. Individual stocks contain more idiosyncratic risk, earnings-event risk, and liquidity variation, which could significantly change the behavior of implied volatility and the profitability of volatility-based signals.

In addition, the reported VaR estimates should be interpreted cautiously. Because each evaluation window is relatively short and the option strategy may trade infrequently, the empirical distribution of daily strategy returns may contain limited tail observations. As a result, VaR can understate downside risk, especially for option strategies with nonlinear payoffs and gap risk. VaR is therefore used as a descriptive risk metric rather than a complete measure of tail exposure.

Overall, these limitations suggest that the proposed strategy should be viewed as a research framework rather than a completely production-ready trading system. Future work could improve the analysis by adding more realistic transaction-cost modeling, better model engineering with more extensive access to data, dynamic delta hedging, regime-aware thresholds, and broader cross-asset testing.

7 Conclusion

This paper studies whether model-based volatility forecasts can be used to construct an implementable options trading strategy based on the volatility risk premium. The central idea is to compare forecasted realized volatility with market-implied volatility from SPY options. When the model predicts realized volatility to be meaningfully higher than implied volatility, the strategy takes long-volatility exposure. When implied volatility is meaningfully higher than the model forecast, the strategy takes short-volatility exposure in an attempt to capture the volatility risk premium.

Overall, the results suggest that volatility forecasting models can provide useful information for identifying potential mispricing in options markets, but the effectiveness of the strategy depends strongly on model choice, market regime, and execution assumptions. The GARCH-family models provide a transparent and interpretable framework for capturing volatility clustering, while machine learning models such as Random Forest and LSTM offer more flexibility in modeling nonlinear relationships. However, more complex models do not automatically translate into better trading performance, since they may introduce overfitting, lagged responses, or instability across regimes.

A key finding of this study is that volatility model performance is highly regime-dependent. Models that perform better during high-volatility periods may not perform as well during calmer markets, and models that are stable in normal periods may underreact during sudden volatility spikes. This highlights the importance of evaluating volatility strategies across multiple market environments rather than relying on a single backtest period. It also suggests that a successful volatility trading system should balance predictive accuracy with robustness and practical implementability.

The trading framework also shows the importance of disciplined signal construction and risk management. By using both absolute and relative thresholds, the strategy avoids trading on small forecast differences that may not be economically meaningful after transaction costs. In addition, position sizing, stop-loss rules, profit-taking rules, and time-based exits help control downside risk and align the holding period with the model's forecast horizon. These features are especially important because options are convex instruments and can experience rapid changes in value.

Future work could extend this study in several directions. First, the strategy could incorporate more realistic transaction-cost assumptions, including bid–ask spreads, slippage, commissions, and partial fills. Second, future versions could add dynamic delta hedging to better isolate volatility exposure from directional exposure. Third, the fixed trade-entry thresholds could be replaced with regime-aware thresholds that adjust based on VIX levels, realized volatility, or liquidity conditions. Fourth, the strategy could be tested across a broader universe of assets, including sector ETFs, individual equities, index options, and futures options, to evaluate whether the results generalize beyond SPY.

Another promising direction is to combine multiple forecasting models into an ensemble framework.

Since GARCH, Random Forest, and LSTM models capture different aspects of volatility dynamics, an ensemble approach may improve forecast stability and reduce dependence on any single model. Future research could also explore alternative targets, such as variance risk premium, implied-realized volatility spreads across different maturities, or delta-hedged option returns.

In conclusion, this paper demonstrates that the volatility risk premium can be studied through a structured framework that connects volatility forecasting, options pricing, trading signals, and risk management. While the strategy is not a complete production-ready trading system, it provides a useful foundation for understanding how predictive volatility models can be translated into practical options trading decisions.

References

Bakshi, G. and Kapadia, N. (2003). Delta-hedged gains and the negative market volatility risk premium. *Review of Financial Studies*, 16(2):527–566.

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

Bollerslev, T., Tauchen, G., and Zhou, H. (2009). Expected stock returns and variance risk premia. *Review of Financial Studies*, 22(11):4463–4492.

Broadie, M., Chernov, M., and Johannes, M. (2009). Understanding index option returns. *Review of Financial Studies*, 22(11):4493–4529.

Carr, P. and Wu, L. (2009). Variance risk premia. *Review of Financial Studies*, 22(3):1311–1341.

Cho, K. et al. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. *arXiv preprint*.

Coval, J. D. and Shumway, T. (2001). Expected option returns. *Journal of Finance*, 56(3):983–1009.

Demeterfi, K., Derman, E., Kamal, M., and Zou, J. (1999). More than you ever wanted to know about volatility swaps. *Goldman Sachs Quantitative Strategies Research Notes*.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*.

Jiang, G. J. and Tian, Y. S. (2005). The model-free implied volatility and its information content. *Review of Financial Studies*, 18(4):1305–1342.

Nelson, D. M. Q., Pereira, A. C. M., and de Oliveira, R. A. (2017). Stock market's price movement prediction with lstm neural networks. *International Joint Conference on Neural Networks*.

Zhang, G. P., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*.

Backtest Links

- **GARCH In-Sample:** https://www.quantconnect.cloud/backtest/01513cb107eef925f3f4bbb400a3d72e/?theme=chrome
- **GARCH OOS A:** https://www.quantconnect.cloud/backtest/b11637ee059a7c21d064909a79108f54/?theme=chrome
- **GARCH OOS B:** https://www.quantconnect.cloud/backtest/80d6b31087d05357ea7c02cbf714cf30/?theme=chrome
- **GARCH OOS C:** https://www.quantconnect.cloud/backtest/3cc8e40ff3665e4ba2b5ba6e3e042370/?theme=chrome
- **GARCH STRESS_2020:** https://www.quantconnect.cloud/backtest/16096037df1e5b993b1f615a33f4b57c/?theme=chrome
- **GARCH STRESS_2008:** https://www.quantconnect.cloud/backtest/c3568269502e26bfdc496e2b2d2b40e0/?theme=chrome
- **Neural Network In-Sample:** https://www.quantconnect.cloud/backtest/479cf5e94110df7d36c5e454360a9503/?theme=chrome
- **Neural Network OOS A:** https://www.quantconnect.cloud/backtest/8fe95ea7ead61711bb6f273d46675bf6/?theme=chrome
- **Neural Network OOS B:** https://www.quantconnect.cloud/backtest/407bdc73e5a0251365f3564087f53b34/?theme=chrome
- **Neural Network OOS C:** https://www.quantconnect.cloud/backtest/ed04c8202c9b9d5b755ac9d2fe1803cd/?theme=chrome
- **Neural Network STRESS_2020:** https://www.quantconnect.cloud/backtest/0d6b1f58135ef86b16160b1aeb9d7d94/?theme=chrome
- **Neural Network STRESS_2008:** https://www.quantconnect.cloud/backtest/fecd934bdf1d6e8d90d6760402148da7/?theme=chrome