Stochastic vs. Machine Learning Methods

A comparative analysis of volatility forecasting approaches using EWMA, GARCH-family models, and LSTM neural networks on daily Microsoft (MSFT) returns from 2015 to 2025.

Thomas Fouilloux, Wai Lok Law, Luv Barnwal, Aayan Zangie

Data Science Society - 2025 AT Project Showcase

Contents

What is Volatility? σ

Volatility = how much an asset's price fluctuates over time

Three key facts:

1. **We can't observe it directly** - we only see prices, not "true" volatility
2. **It clusters** - turbulent periods follow turbulent periods, calm follows calm
3. **Bad news hits harder** - negative returns increase volatility more than positive returns of equal size (the "leverage effect")

Our challenge: forecast tomorrow's volatility using only today's information

Real-world applications:

- **Risk Management**: Banks calculate Value-at-Risk to set capital reserves
- **Option Pricing**: Black-Scholes and other models require volatility estimates
- **Portfolio Construction**: Investors size positions based on expected risk

The stakes:

- Underestimating volatility could lead to catastrophic unexpected losses
- Overestimating volatility could lead to missing profitable opportunities

Data

- Microsoft (MSFT) daily returns
- Period: 2015–2025
- Training: 2015–2021
- Testing: 2022–2025

Returns:

$$r_t = \ln\left(\frac{\text{Price}_t}{\text{Price}_{t-1}}\right)$$

Key Challenge

We never observe "true" volatility - only noisy proxies like squared returns.

Our Approach:

- Evaluate forecasts against **realised volatility** (average of squared returns over 5, 20, or 25 days)
- Use **QLIKE** loss function - robust even with noisy proxies

Lower QLIKE = Better Forecasting Performance

Exponentially Weighted Moving Average (EWMA)

The EWMA model estimates conditional variance using a recursive function:

$$\boxed{\hat{\sigma}_t^2 = \lambda \hat{\sigma}_{t-1}^2 + (1-\lambda) r_{t-1}^2}$$

Parameters:

- $\hat{\sigma}_t^2$: Variance forecast for time t
- $\hat{\sigma}_{t-1}^2$: Previous Variance Estimate for time $t-1$ (i.e. Variance Estimate for Today)
- r_{t-1}: Return at time $t-1$
- λ: Decay factor $\lambda \, \epsilon \, \{0, 1\}$ (typically between 0.85 and 0.99)
- **RiskMetrics (p.4) standard**: $\lambda = 0.94$

Key insight: Higher values of λ assign more weight to past observations, resulting in smoother volatility estimates.

Hyperparameter Tuning Methodology

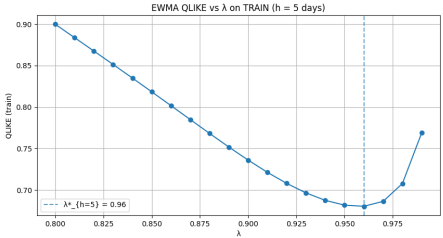
1. Calculate QLIKE for $\lambda \in \{0.80, 0.81, \ldots, 0.99\}$
2. Evaluate across three forecast horizons (5, 20, and 25 days)
3. Select optimal λ by minimising QLIKE on the whole training set

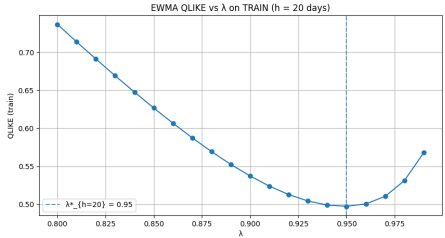
Optimal Lambda Selection Results:

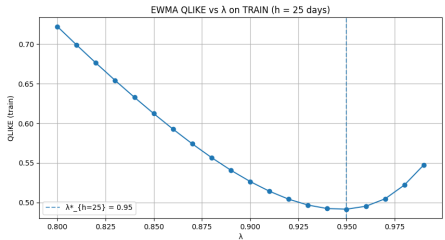
Horizon	Optimal Lambda
5-day	0.96
20-day	0.95
25-day	0.95

Note: Higher decay factors were consistently preferred across all horizons.

EWMA Lambda Selection (2/2)







GARCH(1,1) Model

Generalised Autoregressive Conditional Heteroskedasticity captures "volatility clustering":

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

- $\omega > 0$: Baseline variance (constant term)
- $\alpha \geq 0$: ARCH coefficient (reaction to recent shocks)
- $\beta \geq 0$: GARCH coefficient (persistence)
- $\varepsilon_{t-1} = r_{t-1} - \mu$: Return residual at $t-1$

Core Idea	Intuition
"Volatility persists"	Today's volatility depends on yesterday's volatility + yesterday's shock

GJR-GARCH(1,1) Model, Capturing the Leverage Effect

Following Glosten, Jagannathan, and Runkle (1993), we add an asymmetric term:

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

Additional Parameter:

- $\gamma \geq 0$: Leverage coefficient (asymmetric response)
- $\mathbb{I}_{[\varepsilon_{t-1}<0]}$: Indicator function (equals 1 when returns residuals are negative, otherwise 0)

Leverage Effect:

- Negative shocks impact: $(\alpha + \gamma)$
- Positive shocks impact: α

Note: the rational behind the leverage coefficient γ is that negative returns increase volatility more than positive returns of equal magnitude.

What is an LSTM?
A neural network designed for sequential data - it "remembers" patterns across time.

Pure LSTM

- **Input**: Past returns, squared returns, rolling volatilities
- **Output**: Direct forecast of future volatility
- *"Learn everything from scratch"*

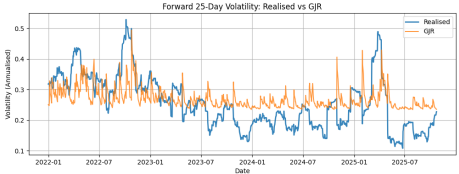
Hybrid LSTM

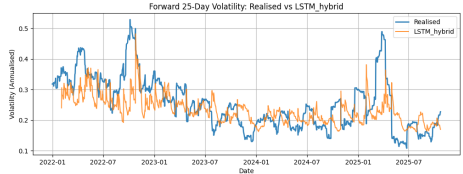
- **Input**: Same features + GJR-GARCH forecast
- **Output**: Correction factor for GJR
- *"Learn to fix GJR's mistakes"*

Final: $\hat{\sigma}^2_{\text{hybrid}} = \hat{\sigma}^2_{\text{GJR}} \times \hat{\rho}_{\text{LSTM}}$

Note: We only used price-derived features to ensure a fair comparison with GARCH models.

LSTM vs. GJR-GARCH 25-Day Forecast Comparison





GJR-GARCH (QLIKE: 0.177) - Best performer

Hybrid LSTM (QLIKE: 0.188) - 6% worse performance relative to GJR-GARCH

Blue = Realised volatility (what actually happened) *Orange = Model forecast*

Horizon	EWMA	GARCH	GJR	LSTM	Hybrid	Winner
5-day	**0.422**	0.424	0.425	0.583	0.584	EWMA
20-day	0.205	0.196	**0.192**	0.249	0.243	GJR
25-day	0.187	0.181	**0.177**	0.205	0.188	GJR

Table: Out-of-Sample QLIKE (Lower = Better)

Key Observations:

- **GJR-GARCH wins** at longer horizons - the leverage effect matters
- **LSTM underperforms** - but the gap narrows at longer horizons
- **Simple beats complex** - 4 parameters outperform thousands

Thank You! σ

Questions?

Thomas Fouilloux, Wai Lok Law, Luv Barnwal, Aayan Zangie

Data Science Society - 2025 AT Project Showcase

Appendix σ

Definition

Volatility measures the **dispersion of returns** for a given asset over time. It captures the degree of uncertainty or risk associated with the magnitude of price changes.

Key Characteristics:

- **Volatility is not directly observable** - we can only estimate it from price data
- **Volatility clusters** - periods of high volatility tend to follow high volatility (and vice versa)
- **Volatility is mean-reverting** - extreme levels eventually return toward long-run averages
- **Volatility exhibits asymmetry** - negative returns often increase volatility more than positive returns (the "leverage effect")

In finance, volatility is central to risk management, option pricing, and portfolio construction.

Real-World Applications

- **Risk Management**: Banks and funds use volatility forecasts to calculate Value-at-Risk (VaR) and set capital reserves
- **Option Pricing**: The Black-Scholes model and its variants require volatility estimates - better forecasts mean better pricing
- **Portfolio Optimisation**: Investors adjust position sizes based on expected volatility to manage risk-return trade-offs
- **Trading Strategies**: Volatility forecasts inform hedging decisions and timing of market entry/exit

Accurate volatility forecasting has direct financial implications: underestimating risk can lead to catastrophic losses, while overestimating can leave profitable opportunities on the table.

The Challenge: Volatility is Latent

True volatility (σ_t^2) is unobservable. We use **proxies** to estimate it:

Common Volatility Proxies:

- **Squared Returns**: $\hat{\sigma}_t^2 = r_t^2$ (very noisy, but unbiased)
- **Realised Variance**: $RV_t = \sum_{i=1}^{n} r_{t,i}^2$ (sum of intraday squared returns)
- **Rolling Window**: $\hat{\sigma}_t^2 = \frac{1}{n} \sum_{i=1}^{n} r_{t-i}^2$ (equally weighted historical)

Our Approach:

$$\sigma^2_{\text{realised},t} = \frac{1}{h} \sum_{i=1}^{h} r_{t+i}^2$$

We use the average of squared daily returns over a forward-looking horizon ($h = 5$, 20, or 25 days) as our volatility proxy, following the methodology when high-frequency data is unavailable.

Understanding QLIKE: Our Scoring Metric

σ

Why Not Just Use Mean Squared Error (MSE)?

When evaluating volatility forecasts, we face a unique problem: we never observe the "true" volatility, only noisy proxies like squared returns.

QLIKE's Advantage:

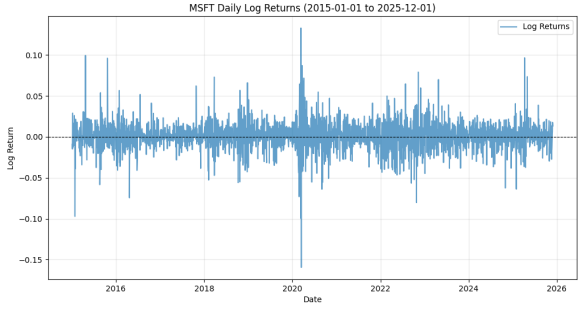
- **Robust to proxy noise** - correctly ranks models even when our volatility estimate is imperfect
- **Penalises underestimation** - missing a volatility spike is often more costly than overestimating
- **Less sensitive to outliers** - financial data has extreme events that can distort MSE

Interpretation:

Lower QLIKE = Better Forecast

A QLIKE of 0.20 means the model's forecasts are, on average, closer to realised volatility than a model with QLIKE of 0.40.

Visualising Volatility Clustering



Example: MSF daily returns showing periods of calm (small fluctuations) followed by turbulence (large swings)

Key Observations:

- Large price movements tend to be followed by large movements (of either sign)
- Small price movements tend to be followed by small movements
- This pattern is what GARCH models are designed to capture

Note: This is why assuming constant volatility (as in basic models) is unrealistic for financial data.

Dataset Overview

Dataset	Ticker	Period	Source
Microsoft	MSFT	2015-01-01 to 2025-12-01	Yahoo Finance

Chosen Splits Splits

Split	Period	Purpose
In-Sample (Training)	2015-01-01 to 2021-12-31	Model estimation & LSTM training
Validation	2020-01-01 to 2021-12-31	Testing (used occasionally in LSTM tuning)
Out-of-Sample (Test)	2022-01-01 to 2025-12-01	Final performance evaluation

Returns: Continuously compounded daily returns (S_t = the adjusted closing price at time t):

$$r_t = \ln\left(\frac{S_t}{S_{t-1}}\right)$$

Loss Function: QLIKE

Following Patton (2011), we use QLIKE as our evaluation metric:

- Only QLIKE and MSE preserve robustness when using noisy volatility proxies: $h_t^* = E_{t-1}[\hat{\sigma}_t^2] = \sigma_t^2$
- QLIKE yields higher statistical power in Diebold-Mariano-West tests when compared to MSE
- Less sensitive to extreme outliers common in financial data and it penalises underestimation of volatility

QLIKE Definition:

$$\text{QLIKE} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\sigma^2_{\text{realised},t}}{\sigma^2_{\text{forecast},t}} - \ln\left(\frac{\sigma^2_{\text{realised},t}}{\sigma^2_{\text{forecast},t}}\right) - 1\right)$$

- $\sigma^2_{\text{realised},t}$: Realised variance (average squared returns over forecast horizon)
- $\sigma^2_{\text{forecast},t}$: Model's variance forecast
- **Lower QLIKE = Better forecasting performance**

Methodology Disclaimer

Due to the absence of easily accessible high-frequency intraday data (which is the academic standard):

- We evaluate 1-day forecasts against realised volatility over three horizons:
  - 5-days (one week of trading)
  - 20-days (one month of trading)
  - 25-days (one month of trading)
- This approach reduces the difficulty of forecasting single-day squared returns which is a very noisy estimate of realised volatility (Andersen and Bollerslev, 1998)
- Provides a more stable benchmark for model comparison

High Frequency Data is the academic standard, indeed, Hansen & Lunde, 2005 (p.13) explain that many utilise realised volatility computed from intraday returns. This requires high-frequency data not freely available in our case

Why We Use Multi-Day Realised Volatility

Due to the absence of freely available high-frequency intraday data:

- We evaluate forecasts against realised volatility over 5, 20, and 25 days
- This reduces the noise inherent in single-day squared returns (Andersen and Bollerslev, 1998)
- Provides a more stable benchmark for model comparison

Academic Standard:
Hansen & Lunde (2005, p.13) recommend using realised volatility from intraday returns, which requires high-frequency tick data not freely available.

Implication: Our EWMA forecasts are 1-day ahead evaluated against multi-day realised volatility, while GARCH and LSTM forecasts match the evaluation horizon.

How Do Our Models "Think"?

EWMA	*"Recent events matter more."*
	Applies exponentially decaying weights to past observations. Simple, fast, and reactive.
GARCH	*"Volatility persists and clusters."*
	Models volatility as depending on both recent shocks and its own past values.
GJR-GARCH	*"Bad news hits harder."*
	Adds asymmetry: negative returns increase volatility more than positive returns.
LSTM	*"Let the data reveal complex patterns."*
	Neural network that learns nonlinear relationships from sequences of data.

EWMA - Exponentially Weighted Moving Average

$$\underbrace{\hat{\sigma}_t^2}_{\text{Tomorrow's forecast}} = \underbrace{\lambda}_{\text{Memory}} \cdot \underbrace{\hat{\sigma}_{t-1}^2}_{\text{Today's estimate}} + \underbrace{(1-\lambda)}_{\text{Reactivity}} \cdot \underbrace{r_{t-1}^2}_{\text{Today's shock}}$$

GARCH(1,1) - Generalised Autoregressive Conditional Heteroskedasticity

$$\sigma_t^2 = \underbrace{\omega}_{\text{Baseline}} + \underbrace{\alpha \cdot \epsilon_{t-1}^2}_{\text{Shock reaction}} + \underbrace{\beta \cdot \sigma_{t-1}^2}_{\text{Persistence}}$$

GJR-GARCH(1,1) - Adds asymmetry for the leverage effect

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

γ = extra volatility boost when returns are negative

GARCH(1,1) Model

Generalised Autoregressive Conditional Heteroskedasticity captures "volatility clustering":

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

- $\omega > 0$: Baseline variance (constant term)
- $\alpha \geq 0$: ARCH coefficient (reaction to recent shocks)
- $\beta \geq 0$: GARCH coefficient (persistence)
- $\varepsilon_{t-1} = r_{t-1} - \mu$: Return residual at $t-1$

Unconditional (Long-run) Variance:

$$\sigma_{LR}^2 = \frac{\omega}{1 - \alpha - \beta} \quad \text{(requires } \alpha + \beta < 1 \text{ for stationarity)}$$

GJR-GARCH(1,1) Model, Capturing the Leverage Effect

Following Glosten, Jagannathan, and Runkle (1993), we add an asymmetric term:

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

Additional Parameter:

- $\gamma \geq 0$: Leverage coefficient (asymmetric response)
- $\mathbb{I}_{[\varepsilon_{t-1}<0]}$: Indicator function (equals 1 when returns residuals are negative, otherwise 0)

Leverage Effect:

- Negative shocks impact: $(\alpha + \gamma)$
- Positive shocks impact: α

Note: the rational behind the leverage coefficient γ is that negative returns increase volatility more than positive returns of equal magnitude.

Information Criteria for Model Selection

Following Tsay (2005), we use information criteria rather than QLIKE for GARCH selection:

Akaike Information Criterion (AIC):

$$\text{AIC} = 2k - 2\ln(\hat{L})$$

Bayesian Information Criterion (BIC):

$$\text{BIC} = k\ln(n) - 2\ln(\hat{L})$$

- k: Number of parameters
- $\hat{L}$: Maximised likelihood
- n: Sample size

Why BIC? Penalises model complexity more heavily than AIC, favouring parsimonious specifications which are less prone to overfitting.

Model Selection Results

Estimated all combinations of GARCH(p, q) and GJR-GARCH(p, o, q) for $p, q, o \in \{1, 2, 3, 4, 5\}$:

Rank	Model	AIC	BIC
1 (AIC)	GJR-GARCH(5,5,1)	6332.23	6403.39
1 (BIC)	**GJR-GARCH(1,1,1)**	6358.02	**6385.39**
2 (AIC)	GJR-GARCH(5,5,5)	6334.15	6427.21
2 (BIC)	**GARCH(1,1)**	6366.19	**6388.08**

Selected Models:

- **Primary**: GJR-GARCH(1,1) - Best BIC
- **Benchmark**: GARCH(1,1) - Reference comparison (and second best BIC)

Our findings are consistent with Hansen & Lunde (2001) (p.27): "We do not find much evidence that the GARCH(1,1) model is outperformed."

How we compute forecasts beyond 1-day ahead:

GARCH:

$$E_t[\sigma^2_{t+h}] = \omega + (\alpha + \beta) \cdot E_t[\sigma^2_{t+h-1}] \quad \text{for } h \geq 2$$

GJR-GARCH:

$$E_t[\sigma^2_{t+h}] = \omega + \left(\alpha + \beta + \frac{\gamma}{2}\right) \cdot E_t[\sigma^2_{t+h-1}] \quad \text{for } h \geq 2$$

where $h \in \{5, 20, 25\}$ days.

Note: The $\frac{\gamma}{2}$ term in GJR-GARCH assumes symmetric distribution of positive/negative shocks in expectation.

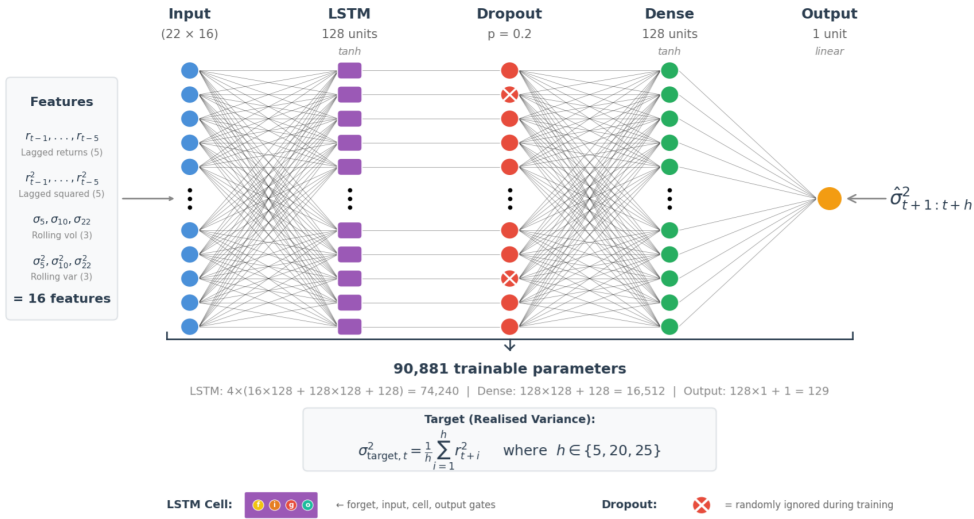
Literature-Based/In-sample Hyperparameter Selection

Parameter	Value	Justification
Lookback window	22 days	Roszyk & Slepaczuk (2024)
LSTM units	128	Roszyk & Slepaczuk (2024)
LSTM layers	1	In-sample testing
Dropout rate	0.2	Srivastava et al. (2014)
Dense hidden units	128	In-sample testing
Dense hidden layers	1	In-sample testing
Epochs	50 (early based on a validation set)	In-sample testing
Batch size	64	In-sample testing
Activation	tanh	Roszyk & Slepaczuk (2024)
Validation set early stopping condition	80/20% split on the training set	In-sample testing

Other key Literature used:

- Kim & Won (2018): A hybrid model integrating LSTM with multiple GARCH-type models
- Kakade et al. (2021): A Hybrid Ensemble Learning Garch-Lstm Based Approach
- Hu, Ni & Wen (2020): A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction

Pure LSTM Architecture



Forward-Looking Pure LSTM

Directly predicts future realised variance over different horizons.

Features (all backward-looking, available at time t):

- **Lagged returns** (last 5 days): $r_{t-1}, r_{t-2}, \ldots, r_{t-5}$
- **Lagged squared returns** (ARCH effect): $r^2_{t-1}, r^2_{t-2}, \ldots, r^2_{t-5}$
- **Rolling volatilities**: windows of 5, 10, and 20 days
- **Rolling variances**: windows of 5, 10, and 20 days

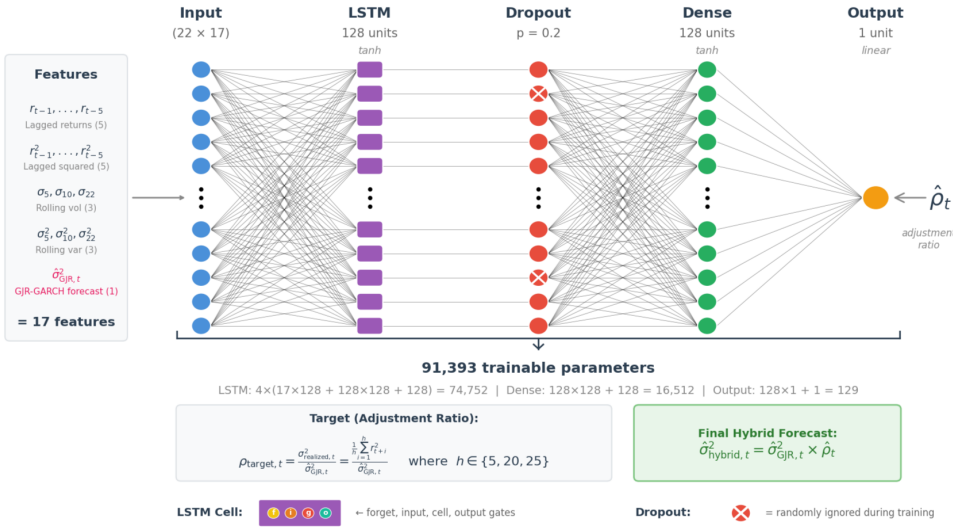
Target: is the realised variance over horizon h defined as per the below

$$\sigma^2_{\mathrm{target},t} = \frac{1}{h}\sum_{i=1}^{h} r^2_{t+i}$$

where h is the forecast horizon (5, 20, or 25 days).

The model learns to predict average squared returns over the specified horizon directly from historical features exclusively derived from returns (i.e. no external data is added to the LSTM).

Hybrid LSTM Architecture



GARCH-LSTM Hybrid Model

We propose a novel GARCH-LSTM hybrid model: our model's objective is to predict how "wrong" GJR-GARCH forecasts is compared to realised volatility.

- This also draws from Hu, Ni & Wen (2020)'s research as they demonstrated that GARCH forecasts can serve as informative features to increase LSTM predictive power.

Features:

- **Same features as the pure LSTM**
- **GJR-GARCH forecasts** (persistence signal)

Target: Adjustment Ratio, i.e. the LSTM predicts how much to "fix" GJR-GARCH's forecast:

$$\rho_t = \frac{\sigma^2_{\text{realised},t}}{\hat{\sigma}^2_{\text{GJR},t}}$$

Final Forecast:

$$\boxed{\hat{\sigma}^2_{\text{hybrid},t} = \hat{\sigma}^2_{\text{GJR},t} \times \hat{\rho}_{\text{LSTM},t}}$$

Results and Findings

Results:

Horizon	EWMA	GARCH	GJR	LSTM	Hybrid	Best
5-day	**0.4222**	0.4242	0.4250	0.5828	0.5842	EWMA
20-day	0.2049	0.1958	**0.1922**	0.2494	0.2425	GJR
25-day	0.1866	0.1814	**0.1771**	0.2050	0.1881	GJR

Table: Out-of-Sample QLIKE Loss Comparison (Lower is Better)

Main Findings:

- Parsimonious GARCH-family models, specifically **GJR-GARCH(1,1)**, generally outperform complex LSTM architectures for daily equity volatility forecasting, particularly at longer horizons. Interestingly, EWMA marginally outperforms all models at the 5-day horizon.
- Our findings align with Hansen and Lunde (2005), who found that simple GARCH(1,1) is difficult to beat in out-of-sample forecasting.
- LSTM underperformance was likely caused by the constraint of exclusively using daily returns as a basis for features. Indeed, contrary to GARCH and EWMA, LSTMs perform best with higher dimensional data. Excluding exogenous variables like VIX, volume, or sentiment may significantly limit the model's predictive ability.

Key Takeaways

What We Learned

1. **Simple models can win**: GJR-GARCH(1,1), despite having only 4 parameters, outperformed complex neural networks with thousands of parameters.

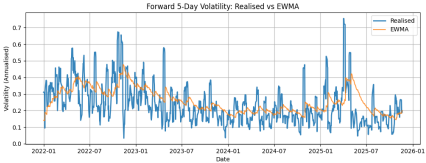
2. **The leverage effect matters**: Accounting for asymmetry (bad news → higher volatility) consistently improved forecasts.

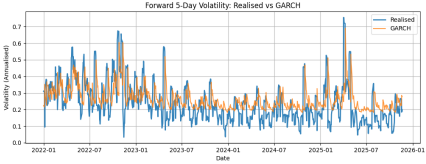
3. **Deep learning isn't magic**: LSTMs need more data, richer features, and better targets to outperform classical approaches.

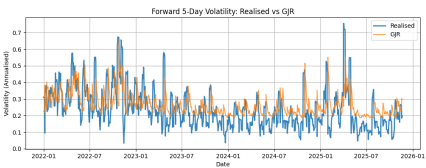
4. **Horizon matters**: Different models excel at different forecast horizons - there's no single "best" model.

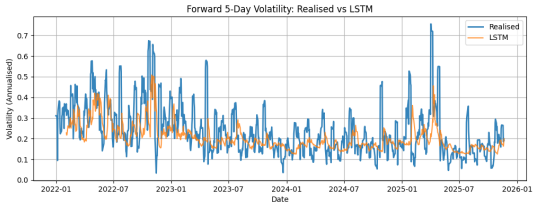
Practical Implication:
For daily equity volatility forecasting with limited data, start with GJR-GARCH before reaching for complex machine learning solutions.

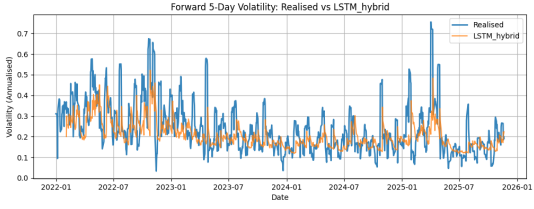
5-Day Volatility Forecasts



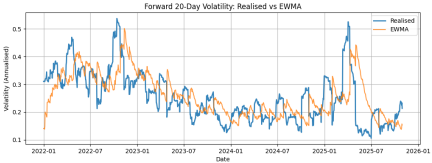


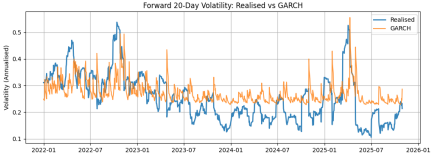


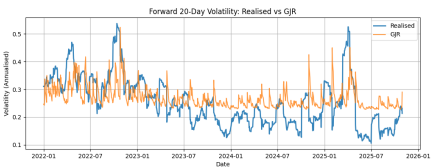


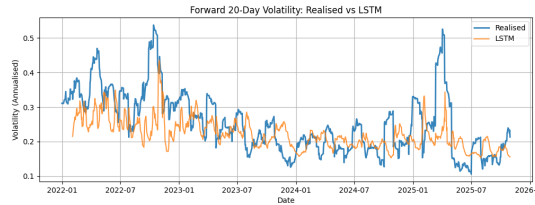


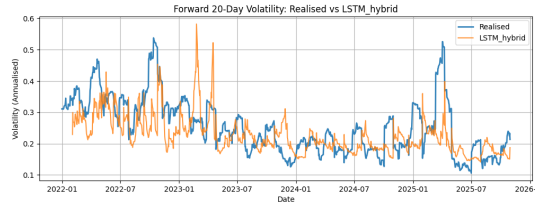
20-Day Volatility Forecasts



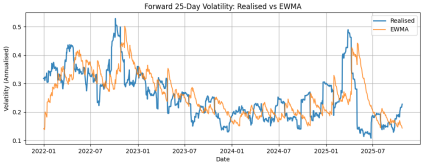


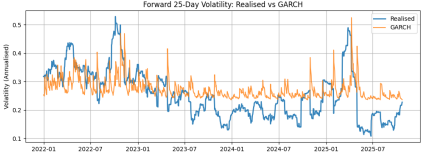


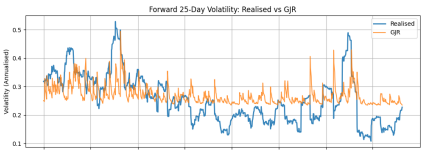


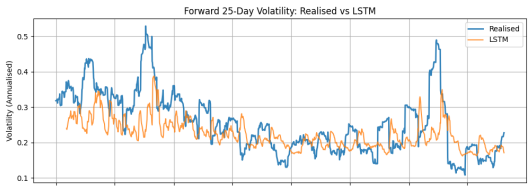


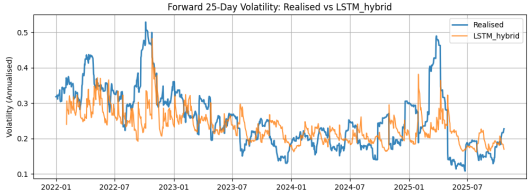
25-Day Volatility Forecasts











- **GARCH Models Dominate:** The leverage effect captured by the asymmetric term in GJR-GARCH models provides consistent forecasting improvements across horizons. EWMA marginally outperforms at the 5-day horizon.
- **LSTM Underperformance:** Both pure and hybrid architectures fail to beat GJR-GARCH. The gap is particularly evident at the 5-day horizon (QLIKE: 0.58/0.58 vs 0.42).
- **Hybrid Approach Limitations:** Incorporating GJR forecasts as features into the LSTM degraded performance rather than improving it, suggesting the neural network failed to leverage the econometric signal effectively.
- **Horizon Sensitivity:** The performance gap between GARCH and LSTM widest at shorter horizons (5-day) and narrows at longer horizons (20-day and 25-day). Indeed, the Hybrid LSTM performs closest to GJR-GARCH at the 25-day horizon with only a 6% performance gap in relative terms.

Conclusion

Main Findings:

- Parsimonious GARCH-family models, specifically **GJR-GARCH(1,1)**, outperform complex LSTM architectures for daily equity volatility forecasting in terms of QLIKE , with the performance gap largest at shorter horizons. Interestingly, EWMA marginally outperforms all models at the 5-day horizon, suggesting simpler approaches may suffice for short-term forecasting.
- Our findings align with Hansen and Lunde (2005), who found that simple GARCH(1,1) is difficult to beat in out-of-sample forecasting.

Interpreting the LSTM Results

We do not consider the negative LSTM results as evidence of fundamental limitations of deep learning models for volatility forecasting, indeed, they highlight specific challenges:

1. **Noise & Sample Size:** Daily returns are noisy, and $\approx$2,700 observations may be insufficient for deep learning pattern recognition.
2. **Feature Engineering:** Using only price-derived features limits the LSTM. It likely requires exogenous variables (Volume, VIX, Sentiment, etc.) to add value beyond GARCH.
3. **Evaluation Proxy:** Using squared daily returns as a target introduces measurement error that disadvantages models trying to learn subtle non-linear patterns.

Limitations and Future Work

- **High-Frequency Data**: Future work should use intraday data to construct **Realised Volatility**. Andersen and Bollerslev (1998) showed this dramatically improves evaluation accuracy compared to squared daily returns.
- **Alternative Architectures**: Transformer-based models with attention mechanisms have been used in recent literature (Soroka and Arzyn, 2025).
- **Broader Scope**: Robustness should be tested across multiple assets, sectors, and asset classes rather than a single equity (MSFT).
- **Regime Dependence**: The test period (2022-2025) covers extreme volatility shifts. Regime-switching models or separate training for high/low volatility periods could improve accuracy.
- **Data Volume**: Extending the training period or using a panel of assets would provide the larger dataset deep learning models typically require to generalise effectively.