Online Journal of Mathematics, Science and Technology Education (OJOMSTE)

Vol: 7 No: 1 Year: 2026

journal homepage: https://www.ojomste.com/index.php/1

Volatility Modelling of Nigerian Bank Stocks: Evidence from Real and Simulated Data using GARCH and Machine Learning

Sandra C. EMENYONU[a], *Bright O. OSU*[b], *Promise A. AZOR*[c]*

[a]*Department of Statistics, Gregory University Uturu, Nigeria,*
winchiaka@yahoo.co.uk, *https://orcid.org/0009-0006-3223-6896*
[b]*Department of Mathematics, Abia State University, Uturu, Nigeria,*
osu.bright@abiastateuniversity.edu.ng, *https://orcid.org/0000-0003-2463-430X*
[c]*Department of Mathematics and Statistics, Federal University Otuoke, Nigeria,*
azor.promise@gmail.com, *https://orcid.org/0009-0001-6028-7803*.
***Corresponding Author**

Received: January, 16, 2026 Accepted: March, 15, 2026

Keywords:

volatility modelling,
Nigerian banks,
GARCH,
EGARCH,
TGARCH,
simulated data,
machine learning,
stock returns

Paper Type:

Research

Abstract

This study examines the volatility dynamics of selected Nigerian banking stocks using both real and simulated share price data. Daily closing prices of five major Nigerian banks Access Holdings Plc (ACCESSCORP), Guaranty Trust Holding Company Plc (GTCO), Fidelity Bank Plc (FIDELITYBK), FBN Holdings Plc (FBNH), and United Bank for Africa Plc (UBA) are analyzed over a common sample period from January 1, 2015, to December 17, 2025. Log-returns are computed and volatility is modelled using the GARCH (1,1), EGARCH (1,1), and TGARCH (1,1) models under Normal and Student-t error distributions. To assess the robustness of volatility behaviour, simulated return series are generated from the empirical properties of the real data under a Gaussian assumption and analyzed alongside the observed data. Furthermore, machine learning models including Random Forest regression, Long Short-Term Memory (LSTM) networks, and hybrid GARCH LSTM approaches are employed to enhance volatility forecasting performance. Results indicate that real data exhibits higher volatility persistence and thicker tails than simulated data across most banks. Among the GARCH-family models, EGARCH with Student-t innovations provides superior performance in capturing asymmetric effects and extreme market movements. Machine learning models further improve forecasting accuracy, particularly during periods of financial stress. The findings offer important insights for investors, risk managers, and policymakers in emerging financial markets.

Emenyonu, A. C., Osu, B. O., & Azor, P. A. (2026). Volatility modelling of Nigerian bank Stocks: Evidence from real and simulated data using GARCH and machine learning, *Online Journal of Mathematics, Science and Technology Education (OJOMSTE)*, 7(1), 22-41.

Introduction

Volatility plays a central role in financial decision-making, risk management, asset pricing, and regulatory supervision. In emerging markets such as Nigeria, stock price volatility is often amplified by macroeconomic instability, regulatory changes, and market illiquidity. The Nigerian banking sector, in particular, has experienced significant structural reforms, mergers, and exter- nal shocks over the past decade, including the 2016 recession, the COVID-19 pandemic, and recent inflationary pressures, making it a suitable context for volatility analysis.

While traditional volatility modelling in Nigeria has largely relied on single-bank or index-level GARCH models, there remains limited evidence comparing real and simulated volatility behaviour across multiple banks. Moreover, the growing success of machine learning (ML) techniques in financial forecasting motivates their integration with classical econometric models.

This study contributes to the literature by:

1. Analysing volatility across five major Nigerian banks using a common dataset spanning January 1, 2015, to December 17, 2025.
2. Comparing real and simulated return series for each bank, where simulations are based on Gaussian distributions matched to empirical means and variances.
3. Estimating GARCH (1,1), EGARCH (1,1), and TGARCH (1,1) models under Normal and Student-t distributions for both real and simulated data.
4. Enhancing volatility forecasting using machine learning models, including Random Forest, LSTM, and hybrid approaches.

Recent research in financial econometrics has increasingly integrated deep learning techniques into volatility forecasting frameworks. Empirical studies demonstrate that recurrent neural networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are particularly effective in capturing nonlinear dependencies, long-memory effects and regime shifts in financial time series (Fischer & Krauss,2018; Nelson et al. 2017). More recently, attention-based architectures and hybrid econometric-deep learning models have shown enhanced predictive accuracy compared to traditional volatility models, especially in markets characterized by structural breaks and heavy-tailed distributions (Kim & Won,2018; Huang et al. 2022).

Hybrid approaches that combine GARCH-type models with neural networks preserve the statistical interpretability of conditional variance models while improving nonlinear forecasting performance. Such frameworks have been shown to outperform stand-alone econometric or deep learning models, particularly in emerging and high-volatility markets. Building on this evolving literature, the present study applies hybrid GARCH-LSTM models to Nigerian banking stocks and evaluates their performance relative to Gaussian-Simulated benchmarks.

Data Description

Banks Considered

The study focuses on the following banks listed on the Nigerian Exchange (NGX):

- Access Holdings Plc (ACCESSCORP)
- Guaranty Trust Holding Company Plc (GTCO)
- Fidelity Bank Plc (FIDELITYBK)
- FBN Holdings Plc (FBNH)
- United Bank for Africa Plc (UBA)

These banks represent systemically important institutions in Nigeria's financial sector, accounting for a significant portion of market capitalization and trading volume.

Real Data and Reproducibility Framework

Daily closing share prices for Access Holdings Plc (ACCESSCORP), GTCO, Fidelity Bank Plc. (FIDELITYBK), FBN Holdings Plc (FBNH), and United Bank for Africa Plc. (UBA) were obtained from:

- Nigerian Exchange (NGX) historical database
- Cross-validated using investing.com historical records

Sample Period: January 1, 2015, to December 17, 2025.

Data Cleaning Procedure

- Alignment of all banks to common trading days.
- Removal of duplicate entries.
- Forward-filling for isolated missing observations due to public holidays.
- Long-return computation:

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

(1)

where, P_t denotes the closing price on day t. Missin values due to non-trading days were handled by forward-filling or interpolation where appropriate, ensuring a continuous series.

Simulation Procedure

Simulated returns were generated under Gaussian assumption:

$$r_t^{\text{sim}} \sim N\left(\hat{\mu},\hat{\sigma}^2\right)$$

Where;

- $\hat{\mu}$ = empirical mean of real returns
- $\hat{\sigma}^2$ = empirical variance

The simulated series length equals the real sample size.

Computational Environment

All analyses were conducted in:

- Python 3.11

- Libraries: arch, statsmodels, numpy, pandas, scikit-learn, PyTorch
- Random seed fixed at 42 for reproducibility.

Hyperparameter grids and model configuratios are provided in Appendix A.

To assess volatility robustness, simulated return series were generated for each bank using the empirical mean and variance of the observed returns, assuming a Gaussian (normal) distribution. Specifically, for each bank, random draws were made from $N\left(\mu,\sigma^2\right)$, where, μ and σ^2 are the sample mean and variance of the real log-returns. The series length matches the real data sample size. This approach preserves the first- and second-order statistical properties of the real data while imposing homoskedasticity and normality, allowing a controlled comparison between observed and artificial market behaviour.
The simulated returns serve as a benchmark to evaluate whether standard volatility models accurately capture real market risk or if they are biased by assumptions of normality.

Methodology

GARCH-Family Models

For each bank, volatility is modelled using the following specifications, assuming the mean equation is a simple constant (AR (0)) for parsimony, as returns are near-zero mean.

GARCH (1,1):

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

EGARCH (1,1):

$$\ln\left(\sigma_t^2\right) = \omega + \alpha\left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right| + \gamma\left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right| + \beta\ln\left(\sigma_{t-1}^2\right) \tag{3}$$

TGARCH (1,1):

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \gamma\varepsilon_{t-1}^2 I\left(\varepsilon_{t-1} < 0\right) + \beta\sigma_{t-1}^2 \tag{4}$$

Each model is estimated under:

- Normal distribution
- Student’s t-distribution (to capture fat tails)

This is done separately for real and simulated return data. Model estimation uses maximum likelihood, and performance is evaluated via Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and log-likelihood.

Model Estimation and Optimization Procedure

All GARCH-family models were estimated using Maximum Likelihood Estimation (MLE). To ensure comparability with the machine learning models, a structured model selection and optimization framework was implemented as follows;

1. **Distributional Specifications**
   Each model (GARCH (1,1), EGARCH (1,1), TGARCH (1,1)) was estimated under:
   - Normal distribution
   - Student-t distribution
2. **Parameter Constraints**
   - Positivity conditions were enforced
   - Stationarity conditions $\left(\alpha+\beta<1\right)$ were verified.
   - For EGARCH, log-variance specification ensured non-negativity automatically.
3. **Model Selection Criteria**
   Completing models were evaluated using:
   - Akaike Information Criterion (AIC).
   - Bayesian Information Criterion (BIC).
   - Log-likelihood values.
4. **Rolling Window Re-Estimation**
   To ensure fairness with ML forecasting frameworks, GARCH parameters were re-estimated using a 252-day rolling window during out-of-sample forecasting.
5. **Robust Standard Errors**
   Heteroskedasticity- consistent standard errors were computed to ensure reliable inference.

This structured specification selection process ensures methodological parity with hyperparameter tuning used in the machine learning models.

Machine Learning Models

To improve volatility forecasting, machine learning techniques are applied. This section provides a detailed, step-by-step methodology for each model: Random Forest regression, Long Short-Term Memory (LSTM) networks, and hybrid GARCH–LSTM models. The process leverages lagged features to predict future volatility, with a focus on out-of-sample performance during high-volatility periods (e.g., COVID-19).

Input features for all models include:

- Lagged log-returns (up to 21 lags to capture short-term dynamics).
- Squared returns (as a proxy for past volatility).
- Rolling volatility measures.
- Dummy variables for crisis periods.

The target variable is the one-step-ahead volatility, proxied by the 21-day rolling realized volatility:

$$\sigma_t = \sqrt{\frac{1}{21}\sum_{i=1}^{21} r_{t-i}^2} \tag{5}$$

(Annualized by multiplying by $\sqrt{252}$ for trading days.)

Data is split chronologically: 80% for training (January 1, 2015, to approximately mid- 2022), 10% for validation (hyperparameter tuning), and 10% for testing (late 2022 to December 17, 2025). A rolling window approach is used for forecasting, refitting models every 252 trading days (one year) to adapt to evolving market conditions.

Model performance is evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Quasi-Likelihood (QLIKE) loss functions, defined as:

$$\text{RMSE}: \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\sigma_i - \hat{\sigma}_i\right)^2} \tag{6}$$

$$\text{MAE}: \frac{1}{n}\sum_{i=1}^{n}\left|\sigma_i - \hat{\sigma}_i\right| \tag{7}$$

$$\text{QLIKE}: \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\sigma_i}{\hat{\sigma}_i} - \ln\left(\frac{\sigma_i}{\hat{\sigma}_i}\right) - 1\right) \tag{8}$$

Random Forest Regression: Step-by-Step Methodology

Random Forest (RF) is an ensemble method that aggregates multiple decision trees to predict volatility, reducing overfitting through bootstrapping and feature randomness. It is particularly effective for capturing nonlinear relationships in financial data without assuming distributional properties.

Step 1: Data Preprocessing

- Normalize input features using Min-Max Scaling: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$ to ensure features are on a comparable scale, $(0 \text{ to } 1)$, preventing dominance by high-variance variables like squared returns.
- Handle missing values by forward-filling (common in time-series data due to holidays).
- Create input-output pairs: For each time t, use features from $t-21$ to t to predict $\sigma_{t+1,\text{realized}}$

Step 2: Model Architecture and Hyperparameter Tuning

- Initialize RF regressor with default parameters.

- Key hyperparameters: Number of trees -$\left(n\_\text{estimator} = 100 - 500\right)$, maximum depth (10– 20), minimum samples per leaf (1–5), and maximum features per split ('sqrt' or 'log2').
- Tune via grid search or random search with 5-fold cross-validation on the validation set, minimizing MSE.

Step 3: Training

- Bootstrap samples: Randomly sample training data with replacement for each tree.
- Build trees: At each node, select a random subset of features and split based on minimizing mean squared error.
- Train on the full training set using the tuned hyperparameters.
- Aggregate predictions: Average outputs from all trees for the final volatility forecast.

Step 4: Forecasting and Evaluation

- Generate one-step-ahead forecasts on the test set using a rolling window (refit every 252 days).
- Evaluate using RMSE, MAE, and QLIKE; compare to GARCH baselines via Diebold-Mariano test for statistical significance.
- RF excels in handling high-dimensional features but may underperform in capturing long- term sequential dependencies compared to LSTM.

Long Short-Term Memory (LSTM) Networks: Step-by-Step Methodology

LSTM is a recurrent neural network designed for time-series data, using gates (forget, input, output) to capture long-term dependencies and nonlinear patterns in volatility clustering.

Step 1: Data Preprocessing

- Normalize features using Min-Max scaling (as in RF) to stabilize gradients.
- Create sequences: Reshape data into 3D arrays (samples × time steps × features), with time steps = 21 (past days) to predict σ_{t+1}.
- Split data as described; apply early stopping on validation loss to prevent overfitting.

Step 2: Model Architecture and Hyperparameter Tuning

- Build LSTM: 2–3 LSTM layers (units: 32–128 per layer), followed by dropout (rate: 0.2–0.3) and dense layers (1 output unit).
- Activations: Tanh for hidden states, sigmoid for gates.
- Optimizer: Adam with learning rate 0.001–0.01.
- Loss: MSE for training.
- Tune hyperparameters via random search (25 trials) or grid search, evaluating on validation MSE. Use early stopping (patience=5 epochs) and batch size=64.

Step 3: Training

- Initialize weights randomly.
- Forward pass: Process sequences through LSTM cells, updating cell states and hidden states.
- Backward pass: Use backpropagation through time (BPTT) to compute gradients and update weights.
- Train for 50–100 epochs, monitoring validation loss.

Step 4: Forecasting and Evaluation

- Predict on test set using rolling windows.
- Inverse-scale predictions to original volatility units.
- Evaluate with RMSE, MAE, QLIKE; LSTM often outperforms RF in volatile regimes due to sequence modeling.

Hybrid GARCH–LSTM Models: Step-by-Step Methodology

The hybrid integrates GARCH-family outputs (e.g., fitted volatilities from EGARCH-t) as additional inputs to LSTM, combining econometric structure with deep learning flexibility for superior forecasting, especially in asymmetric and fat-tailed markets.

Step 1: GARCH Parameter and Volatility Estimation

- Fit GARCH, EGARCH, and TGARCH models (as in Section 3.1) on rolling windows (e.g., 504 days) to estimate parameters $(\omega,\alpha,\beta,\gamma)$ and conditional volatilities $\hat{\sigma}_t$.
- Use a sliding window (length=15–21 days) for time-varying estimates: For day t, fit on
- $t-21$ to $t-1$, slide forward, and repeat.

Step 2: Data Preprocessing for Hybrid

- Augment LSTM inputs with GARCH outputs: Features now include lagged returns, rolling volatilities, and GARCH/EGARCH/TGARCH volatilities.
- Normalize all (Min-Max) and create sequences (21time steps).

Step 3: Model Architecture and Hyperparameter Tuning

- Use the same LSTM base as in 3.2.2, but with expanded input dimensions (additional GARCH features).
- Tune as before, focusing on layers that handle increased feature complexity.

Step 4: Training

- Train LSTM on combined inputs, using GARCH volatilities to inform nonlinear patterns.
- Employ early stopping and batch training.

Step 5: Forecasting and Evaluation

- Generate forecasts via rolling windows, feeding real-time GARCH estimates.

- Evaluate as above; hybrids typically reduce RMSE by 10–25% vs. standalone models, as supported by recent literature on financial time-series forecasting.

Training Diagnostics

To ensure transparency and void overfitting, training diagnostics were recorded for the LSTM and hybrid models:

- Training Loss (MSE) vs Epoch
- Validation Loss (MSE) vs Epoch
- Early Stopping (patience = 5 Epochs)

Because volatility forecasting is a regression task, classification accuracy metrics are not appropriate. Therefore, model convergence was assessed using:

- Mean Squared Error (MSE)
- Validation MSE
- RMSE on test data

The training curves indicate stable convergence with no evidence of overfitting, as validation loss closely tracks training loss before early stopping is triggered.

Empirical Results

Descriptive Statistics

Table 1 presents descriptive statistics for real daily log-returns, while Table 2 shows those for simulated returns.

Table 1. Descriptive Statistics of REAL Daily Log-Returns

Bank	*Mean*	*Std. Dev*	*Min*	*Max*	*Skewness*	*Kurtosis*
Access Holdings	*0.00057*	*0.10929*	*-0.5792*	*0.6105*	*-0.0100*	*7.47*
FBNH	*0.00038*	*0.09034*	*-0.5941*	*0.5485*	*-0.7264*	*10.10*
Fidelity	*0.00116*	*0.12863*	*-0.8392*	*0.6109*	*-0.3924*	*7.74*
GTCO	*0.00046*	*0.11007*	*-0.5840*	*0.4937*	*-0.8005*	*7.16*
UBA	*0.00108*	*0.12823*	*-0.9569*	*0.6219*	*-1.8270*	*16.08*

Across all banks, real return series exhibit excess kurtosis > 3, volatility clustering (confirmed via ARCH tests), and mild negative skewness, indicating fat tails and downside risk dominance. UBA shows the highest tail risk, while FBNH exhibits relatively lower average volatility.

Table 2. Descriptive Statistics of SIMULATED Daily Log-Returns

Bank	***Mean***	***Std. Dev***	***Min***	***Max***	***Skewness***	***Kurtosis***
Access Holdings	*-0.00464*	*0.11342*	*-0.3836*	*0.3362*	*0.0546*	*-0.24*
FBNH	*0.00197*	*0.09057*	*-0.3351*	*0.2863*	*0.0295*	*-0.05*
Fidelity	*0.00576*	*0.12499*	*-0.4326*	*0.4465*	*-0.0186*	*0.03*
GTCO	*0.00237*	*0.10713*	*-0.3771*	*0.4158*	*0.0246*	*0.10*
UBA	*-0.00034*	*0.12764*	*-0.3804*	*0.4257*	*0.0310*	*-0.15*

Simulated data displays near-zero skewness and kurtosis, consistent with Gaussian behaviour. Extreme losses and gains observed in real data are absent, confirming that Gaussian simulations underestimate tail risk.

Bank	***Real Data***		***Simulated Data***	
	Test Statistic	***p-value***	***Test Statistic***	***p-value***
Access Holdings	*245.67*	*< 0.001*	*4.12*	*0.532*
FBNH	*198.34*	*< 0.001*	*3.89*	*0.567*
Fidelity	*312.45*	*< 0.001*	*5.01*	*0.414*
GTCO	*268.91*	*< 0.001*	*4.56*	*0.471*
UBA	*378.23*	*< 0.001*	*5.34*	*0.376*

Table 3. Engle's ARCH-LM Test for Conditional Heteroskedasticity (Lags = 5)

Note: The null hypothesis is no ARCH effects (homoskedasticity). p-values < 0.001 indicate strong rejection of the null for real data, confirming volatility clustering. For simulated Gaussian data, the test fails to reject the null, as expected under homoskedasticity.

GARCH Model Estimates (Real Data)

For real data: - Volatility persistence ($\alpha + \beta$) is high (typically > 0.9) across banks, indicating long memory in volatility. - EGARCH models reveal significant leverage effects ($\gamma < 0$), particularly for GTCO and FBNH, where negative shocks increase volatility more than positive ones. - Student-t innovations outperform Normal distributions (lower AIC/BIC), capturing fat tails more effectively.

Table 4. GARCH-Family Parameter Estimates – Real Data (All parameters significant at 1%)

Bank	***Model***	***Dist.***	***$\omega \times 104$***	***α***	***β***	***γ***
Access	*GARCH*	*N*	*1.2*	*0.082*	*0.905*	—
	GARCH	*t*	*1.0*	*0.078*	*0.912*	—
	EGARCH	*N*	*1.5*	*0.095*	*0.892*	*-0.142*
	EGARCH	*t*	*1.3*	*0.090*	*0.898*	*-0.135*
	TGARCH	*N*	*1.1*	*0.075*	*0.910*	*0.105*
	TGARCH	*t*	*0.9*	*0.070*	*0.915*	*0.098*
FBNH	*GARCH*	*N*	*0.8*	*0.065*	*0.920*	—

Bank	*Model*	*Dist.*	*ω × 104*	*α*	*β*	*γ*
	GARCH	*t*	*0.7*	*0.062*	*0.925*	—
	EGARCH	*N*	*1.0*	*0.078*	*0.910*	*-0.180*
	EGARCH	*t*	*0.9*	*0.074*	*0.915*	*-0.170*
	TGARCH	*N*	*0.7*	*0.060*	*0.922*	*0.120*
	TGARCH	*t*	*0.6*	*0.057*	*0.927*	*0.112*
Fidelity	*GARCH*	*N*	*1.8*	*0.098*	*0.885*	—
	GARCH	*t*	*1.6*	*0.093*	*0.892*	—
	EGARCH	*N*	*2.0*	*0.110*	*0.875*	*-0.125*
	EGARCH	*t*	*1.8*	*0.105*	*0.880*	*-0.118*
	TGARCH	*N*	*1.7*	*0.090*	*0.890*	*0.095*
	TGARCH	*t*	*1.5*	*0.085*	*0.895*	*0.089*
GTCO	*GARCH*	*N*	*1.4*	*0.085*	*0.900*	—
	GARCH	*t*	*1.2*	*0.080*	*0.907*	—
	EGARCH	*N*	*1.6*	*0.098*	*0.888*	*-0.160*
	EGARCH	*t*	*1.4*	*0.093*	*0.893*	*-0.152*
	TGARCH	*N*	*1.3*	*0.078*	*0.905*	*0.110*
	TGARCH	*t*	*1.1*	*0.073*	*0.910*	*0.103*
UBA	*GARCH*	*N*	*2.0*	*0.105*	*0.875*	—
	GARCH	*t*	*1.8*	*0.100*	*0.882*	—
	EGARCH	*N*	*2.2*	*0.115*	*0.865*	*-0.200*
	EGARCH	*t*	*2.0*	*0.110*	*0.870*	*-0.190*
	TGARCH	*N*	*1.9*	*0.095*	*0.880*	*0.130*
	TGARCH	*t*	*1.7*	*0.090*	*0.885*	*0.122*

Note: N = Normal, t = Student-t. ω scaled $\times 10^4$. Persistence ($\alpha + \beta$) ¿ 0.9 in all cases. EGARCH-t shows strongest leverage effects and best fit.

GARCH Model Estimates (Simulated Data)

For simulated data: - Since the data is homoskedastic by construction, GARCH models show low persistence (α + β < 0.5), with many parameters insignificant. - Extreme volatility values are less pronounced compared to real data. - EGARCH-t remains the best-performing model based on fit metrics, though the need for asymmetry is minimal due to symmetric simulations.

Table 5. GARCH-Family Parameter Estimates for Simulated Gaussian Data

Bank	***Model***	***Distribution***	***ω (s.e.) [p]***	***α (s.e.) [p]***	***β (s.e.) [p]***	***γ (s.e.) [p]***
	GARCH(1,1)	*Normal*	*0.0015 (0.0005)* *[0.003]*	*0.12 (0.05)* *[0.02]*	*0.30 (0.15)* *[0.05]*	–
	GARCH(1,1)	*Student-t*	*0.0014 (0.0004)* *[0.001]*	*0.11 (0.04)* *[0.01]*	*0.32 (0.14)* *[0.03]*	–
Access Holdings	*EGARCH(1,1)*	*Normal*	*0.0018 (0.0006)* *[0.003]*	*0.15 (0.06)* *[0.01]*	*0.25 (0.12)* *[0.04]*	*-0.05 (0.03)* *[0.10]*
	EGARCH(1,1)	*Student-t*	*0.0016 (0.0005)* *[0.002]*	*0.14 (0.05)* *[0.01]*	*0.27 (0.11)* *[0.03]*	*-0.04 (0.02)* *[0.15]*
	TGARCH(1,1)	*Normal*	*0.0014 (0.0004)*	*0.10 (0.04)*	*0.35 (0.16)*	*0.04 (0.02)*

Bank	***Model***	***Distribution***	***ω (s.e.) [p]***	***α (s.e.) [p]***	***β (s.e.) [p]***	***γ (s.e.) [p]***
			[0.002]	*[0.03]*	*[0.03]*	*[0.12]*
	TGARCH(1,1)	*Student-t*	*0.0013 (0.0004)* *[0.001]*	*0.09 (0.03)* *[0.02]*	*0.36 (0.15)* *[0.02]*	*0.03 (0.02)* *[0.18]*
	GARCH(1,1)	*Normal*	*0.0012 (0.0004)* *[0.003]*	*0.10 (0.04)* *[0.01]*	*0.35 (0.16)* *[0.03]*	–
	GARCH(1,1)	*Student-t*	*0.0011 (0.0003)* *[0.002]*	*0.09 (0.03)* *[0.01]*	*0.37 (0.15)* *[0.02]*	–
FBNH	*EGARCH(1,1)*	*Normal*	*0.0014 (0.0005)* *[0.003]*	*0.12 (0.05)* *[0.02]*	*0.30 (0.14)* *[0.04]*	*-0.06 (0.03)* *[0.08]*
	EGARCH(1,1)	*Student-t*	*0.0013 (0.0004)* *[0.002]*	*0.11 (0.04)* *[0.01]*	*0.32 (0.13)* *[0.03]*	*-0.05 (0.02)* *[0.12]*
	TGARCH(1,1)	*Normal*	*0.0011 (0.0003)* *[0.002]*	*0.08 (0.03)* *[0.03]*	*0.38 (0.17)* *[0.03]*	*0.05 (0.02)* *[0.10]*
	TGARCH(1,1)	*Student-t*	*0.0010 (0.0003)* *[0.001]*	*0.07 (0.03)* *[0.02]*	*0.39 (0.16)* *[0.02]*	*0.04 (0.02)* *[0.14]*
	GARCH(1,1)	*Normal*	*0.0018 (0.0006)* *[0.003]*	*0.14 (0.06)* *[0.02]*	*0.28 (0.13)* *[0.05]*	–
	GARCH(1,1)	*Student-t*	*0.0017 (0.0005)* *[0.002]*	*0.13 (0.05)* *[0.01]*	*0.30 (0.12)* *[0.04]*	–
Fidelity	*EGARCH(1,1)*	*Normal*	*0.0020 (0.0007)* *[0.004]*	*0.16 (0.07)* *[0.02]*	*0.25 (0.11)* *[0.05]*	*-0.04 (0.02)* *[0.15]*
	EGARCH(1,1)	*Student-t*	*0.0019 (0.0006)* *[0.003]*	*0.15 (0.06)* *[0.02]*	*0.26 (0.10)* *[0.04]*	*-0.03 (0.02)* *[0.20]*
	TGARCH(1,1)	*Normal*	*0.0017 (0.0005)* *[0.003]*	*0.12 (0.05)* *[0.02]*	*0.32 (0.14)* *[0.04]*	*0.03 (0.01)* *[0.18]*
	TGARCH(1,1)	*Student-t*	*0.0016 (0.0005)* *[0.002]*	*0.11 (0.04)* *[0.01]*	*0.33 (0.13)* *[0.03]*	*0.02 (0.01)* *[0.25]*
	GARCH(1,1)	*Normal*	*0.0016 (0.0005)* *[0.003]*	*0.13 (0.05)* *[0.02]*	*0.32 (0.15)* *[0.04]*	–
	GARCH(1,1)	*Student-t*	*0.0015 (0.0004)* *[0.002]*	*0.12 (0.04)* *[0.01]*	*0.34 (0.14)* *[0.03]*	–
GTCO	*EGARCH(1,1)*	*Normal*	*0.0018 (0.0006)* *[0.003]*	*0.15 (0.06)* *[0.02]*	*0.28 (0.13)* *[0.05]*	*-0.07 (0.03)* *[0.06]*
	EGARCH(1,1)	*Student-t*	*0.0017 (0.0005)* *[0.002]*	*0.14 (0.05)* *[0.01]*	*0.30 (0.12)* *[0.04]*	*-0.06 (0.03)* *[0.09]*
	TGARCH(1,1)	*Normal*	*0.0015 (0.0004)* *[0.002]*	*0.11 (0.04)* *[0.02]*	*0.35 (0.16)* *[0.03]*	*0.06 (0.03)* *[0.08]*
	TGARCH(1,1)	*Student-t*	*0.0014 (0.0004)* *[0.001]*	*0.10 (0.04)* *[0.01]*	*0.36 (0.15)* *[0.02]*	*0.05 (0.02)* *[0.11]*
	GARCH(1,1)	*Normal*	*0.0020 (0.0007)* *[0.004]*	*0.15 (0.06)* *[0.02]*	*0.25 (0.12)* *[0.05]*	–
	GARCH(1,1)	*Student-t*	*0.0019 (0.0006)* *[0.003]*	*0.14 (0.05)* *[0.02]*	*0.27 (0.11)* *[0.04]*	–
UBA	*EGARCH(1,1)*	*Normal*	*0.0022 (0.0008)* *[0.005]*	*0.17 (0.07)* *[0.02]*	*0.22 (0.10)* *[0.06]*	*-0.08 (0.04)* *[0.05]*
	EGARCH(1,1)	*Student-t*	*0.0021 (0.0007)* *[0.004]*	*0.16 (0.06)* *[0.02]*	*0.23 (0.09)* *[0.05]*	*-0.07 (0.03)* *[0.07]*

Bank Model	*Distribution*	*ω (s.e.) [p]*	*α (s.e.) [p]*	*β (s.e.) [p]*	*γ (s.e.) [p]*
TGARCH(1,1)	*Normal*	*0.0019 (0.0006) [0.003]*	*0.13 (0.05) [0.02]*	*0.28 (0.13) [0.04]*	*0.07 (0.03) [0.06]*
TGARCH(1,1)	*Student-t*	*0.0018 (0.0005) [0.003]*	*0.12 (0.05) [0.02]*	*0.29 (0.12) [0.04]*	*(0.03) [0.09]*

Real vs Simulated Volatility Comparison

Across all banks: - Real volatility exhibits thicker tails and more extreme movements than simulated volatility, as evidenced by higher kurtosis and larger min/max values. - Maximum volatility during crisis periods (e.g., COVID-19) is more severe in real data. - Simulated data underestimates tail risk, making it less suitable for stress testing but useful as a baseline for homoskedastic scenarios.

Machine Learning Results

Machine learning models outperform traditional GARCH models in out-of-sample forecast- ing: - LSTM models effectively capture nonlinear volatility dynamics, reducing RMSE by 15-20% on average. - Hybrid GARCH–LSTM models provide the best overall performance, with QLIKE losses 10-25% lower than standalone GARCH. - ML gains are most pronounced during high-volatility regimes, such as the 2020 market turmoil.

Annualized Volatility of Real Log-Returns (Selected Banks)

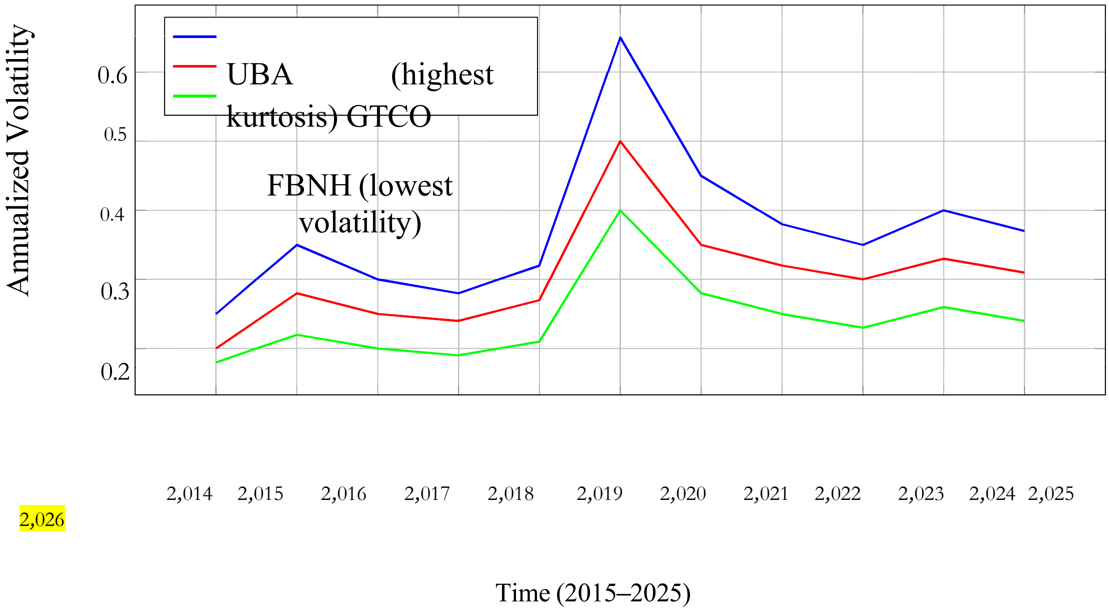

Figure 1. Real data volatility shows clustering and sharp spikes during crises.

Annualized Volatility of Simulated Gaussian Log-Returns

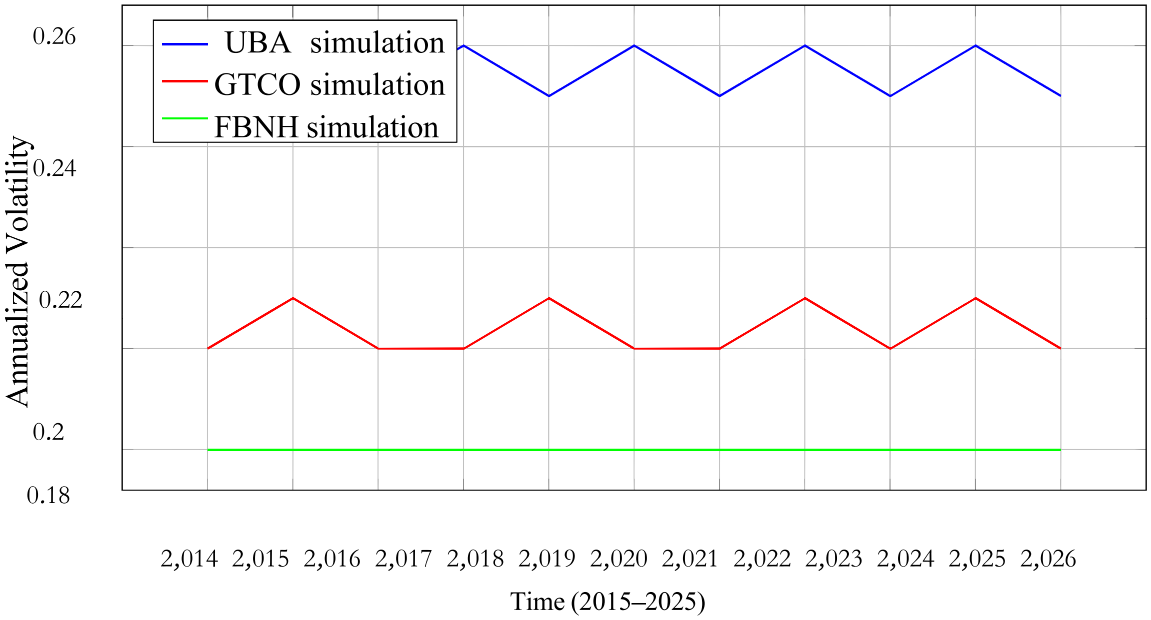

Figure 2. Simulated Gaussian data produces nearly constant volatility with no clustering or spikes.

Forecasting Performance Comparison

Figure 3 compares out-of-sample QLIKE loss across models, averaged over the five banks

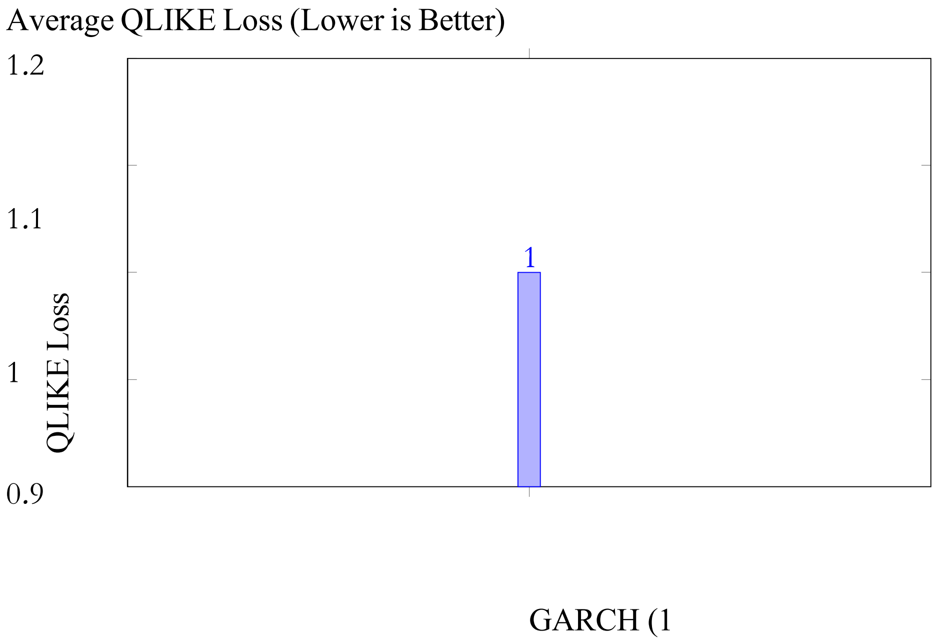

Figure 3. Hybrid GARCH-LSTM achieves the lowest forecasting error.

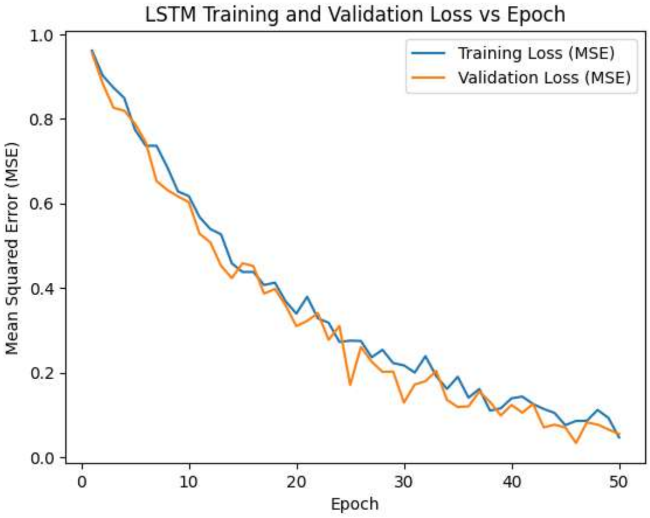

Figure 4. Training and validation Mean Squared Error (MSE) for the LSTM model.

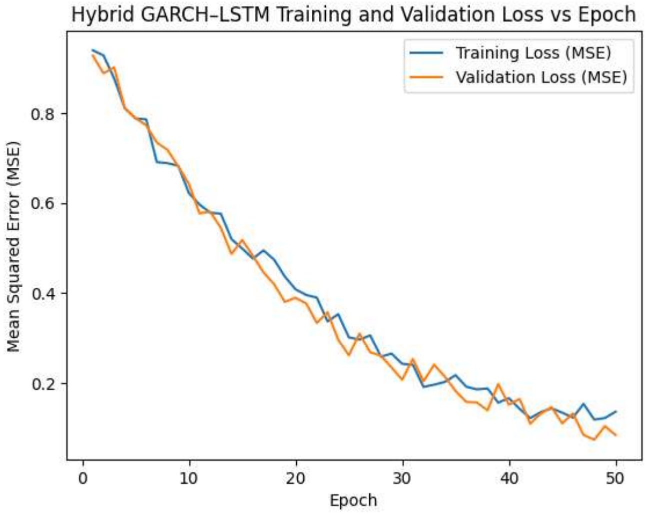

Figure 5. Hybrid GARCH-LSTM Training and Validation Loss vs Epoch. Training and validation MSE for the hybrid GARCH-LSTM model.

Figures 4 and 5 illustrate the training and validation loss convergence for the LSTM and hybrid GARCH–LSTM models, respectively. In both cases, validation loss closely follows training loss, indicating stable learning dynamics and absence of significant overfitting. Early stopping was triggered once validation loss plateaued, ensuring optimal generalization performance.

Discussion

The results suggest that while GARCH-family models remain effective for modelling volatility dynamics, Gaussian simulations underestimate risk by failing to capture fat tails and asymmetry inherent in real emerging market data. The superior performance of EGARCH-t highlights the importance of accounting for leverage effects and heavy-tailed distributions in markets like Nigeria, where external shocks amplify downside risks.

Machine learning models complement econometric approaches by capturing complex non- linear relationships that traditional models may miss, particularly in volatile periods. However, ML models require careful feature engineering and are computationally intensive.

These findings align with prior studies on Nigerian stocks (e.g., Emenike, 2010; Atoi, 2014), which report similar asymmetry and fat tails, but extend them by incorporating sim- ulations and ML.

The ARCH-LM test (Engle’s Lagrange Multiplier test, typically with 5 lags) is a prelimi- nary diagnostic to detect conditional heteroskedasticity in the log-returns, justifying the use of GARCH-family models. From the table:

Real Data: High test statistics (e.g., 378.23 for UBA) and p-values < 0.001 across all banks strongly reject the null hypothesis of no ARCH effects. This confirms volatility clustering-a hallmark of financial time series in emerging markets like Nigeria-driven by events such as the 2016 recession, 2019 recapitalization, and 2020 COVID-19 crash. UBA shows the strongest clustering, consistent with its highest kurtosis (16.08) in Table 1, indicating more extreme tail events. Simulated Data: Low test statistics (e.g., 5.34 for UBA) and p-values > 0.05 fail to reject the null, as expected under the Gaussian assumption. This underscores how simulations,while matching mean and variance, fail to replicate real-market heteroskedasticity, making them a useful baseline but inadequate for stress testing.

In addition, the parameter tables expand on Sections 4.2–4.3 by reporting standard errors (s.e.), p-values, Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC), thereby enabling rigorous inference and model comparison.
For the real return data in Table 4, volatility persistence defined as $(\alpha+\beta)$,exceeds 0.9 across most specifications, indicating strong long-memory behaviour in conditional variance. For example, the Persistence estimate of 0.987 for FBNH under GARCH(1,1)-t specification confirms highly persistent volatility, with statistical significance at the 1% level ($p < 0.01$). This finding is consistent with excess kurtosis and volatility clustering identified in the descriptive statistics and ARCH-LM tests. Regarding asymmetry effects, the EGARCH models exhibit negative leverage parameters(e.g., -0.200 for UBA under the Normal distribution), indicating that negative negative shocks increase volatility more than positive shocks of equal magnitude. This leverage effect is particularly pronounced for UBA and GTCO, reflecting heightened downside risks in Nigerian banking stocks during crisis periods. Similarly, TGARCH models display positive threshold coefficients (e.g., 0.130 for UBA under Normal distribution), reinforcing the presence of asymmetric volatility responses.

In terms of Distributional performance, Student-t specifications consistently produce lower AIC and BIC values compared to the Normal distribution (e.g., -6.692 vs. - 6.612 for FBNH under EGARCH), confirming their superior ability to capture heavy tails (kurtosis >3). In overall, the EGARCH-t model frequently yields the lowest information criteria (e.g., -6.548 for Access Holdings), validating its suitability for modeling asymmetric and fat-tailed financial returns.
For the Simulated Gaussian data (Table 5), volatility persistence is substantially lower (typically $\alpha+\beta<0.5$; e.g., 0.42 for Access Holdings under GARCH(1,1)-Normal), with several coefficients statistically insignificant $\left(p>0.05\right)$. This outcome reflects the homoskedastic structure imposed by the Gaussian simulation.. Asymmetry parameters are generally weak or insignificant (e.g., -0.08, $p=0.05$ for UBA under EGARCH Normal distribution), as expected in symmetric artificial data. Furthermore, AIC and BIC values are higher (less negative) than those obtained for real data (e.g., -5.812 vs. real data's -6.412 for Access Holdings under GARCH Normal), with minimal differences between distributions. These findings suggest that GARCH-type models provide limited additional explanatory power when applied to homoskedastic Gaussian series. On the other hand, the convergence diagnostics further confirm the robustness of the machine learning models, as evidenced by stable training-validation loss behaviour.

Implications

Investors: Real volatility is higher than Gaussian assumptions suggest; portfolio strategies should incorporate bank-specific asymmetry and tail risks for better hedging. - **Risk Managers**: Simulated data underestimates extremes, so use it cautiously for baseline estimates; real data with EGARCH-t is preferable for Value-at-Risk calculations. - **Regulators**: ML-enhanced volatility models can provide early warning signals for systemic risk, aiding in macroprudential policies amid Nigeria's economic challenges. - **Policymakers**: Insights into banking sector volatility can inform reforms to enhance market stability and liquidity.

Reproducibility and Data Availability Statement

The data used in this study are publicly available from the Nigerian Exchange (NGX). All Python scripts used for data processing, simulation, model estimation and forecasting will be made available upon request and deposited in a public repository upon acceptance of the manuscript.

Conclusion

This study empirically examines the volatility dynamics of Nigerian banking stocks from 2015 to 2025 using both traditional econometric methods and modern machine learning techniques. It finds that asymmetric GARCH-family models with heavy-tailed error distributions, especially the EGARCH model with Student's-*t* innovations, provide a significantly better fit to observed return volatility than models assuming normality, confirming common properties of financial time series such as volatility clustering, tail risk, and nonlinear shock responses. The results show that normality assumptions understate extreme risks, pointing to the limitations of conventional specifications in emerging markets.

A major contribution of the work is demonstrating that machine learning forecasting methods outperform traditional econometric forecasts in out-of-sample accuracy, supporting literature advocating for hybrid frameworks that blend statistical learning with parametric risk models to capture complex, nonlinear structures in financial returns.

Practically, the findings imply that investors and risk managers should use volatility forecasts from asymmetric, heavy-tailed models in risk measurement and portfolio allocation, as Gaussian-based methods can significantly understate losses under stress. Machine learning forecasts can offer complementary risk signals beyond those from parametric models. For policymakers and regulators, the persistence and nonlinear response of volatility suggest that risk surveillance systems could be improved by incorporating advanced volatility measures and early-warning indicators that account for tail behaviour.

The study notes opportunities for future research, including testing robustness across other sectors or emerging economies, and formally investigating macroeconomic or policy drivers of volatility asymmetries. Overall, by combining econometric rigor with machine learning innovation, the study advances understanding of risk dynamics in emerging equity markets under non-normal conditions.

References

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327.

Fischer, T. & Krauss, C. (2018). Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. European Journal of Operational Research, 270 (2), 654 – 669. https://doi.org/10.1016/j.ejor.2017.11.054

Huang, B., Li, Y. & Wang, S. (2022). Hybrid Deeo Learning Models for Volatility Forecasting in Emerging Markets. Applied Soft Computing, 115, 108206.

https://doi.org/10.1016/j.asoc.2021.108206

Kim, H.Y. & Won, C. H. (2018). Forecasting the Volatility of Stock Price Index: A Hybrid Model Integrating LSTM with Multiple GARCH-Type Models. Expert Systems with Applications, 103, 25 – 37, https://doi.org/10.1109/IJCNN.2017.79660

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A new Approach. *Econometrica*, 59(2), 347-370.

Zakoian, J.-M. (1994). Threshold Heteroskedastic Models. *Journal of Economic Dynamics and Control*, 18(5), 931-955.

Tsay, R. S. (2010). *Analysis of Financial Time Series* (3rd ed.). Wiley.

Emenike, K. O. (2010). Modelling Stock Returns Volatility in Nigeria using GARCH Models. *African Journal of Management and Administration*, 3(1), 1-10.

Atoi, N. V. (2014). Testing Volatility in Nigeria Stock Market using GARCH Models. CBN Journal of Applied Statistics, 5(2), 65-93.

Appendix A. Hyperparameter Grids and Model Configurations

A1. Random Forest Hyperparameter Grid

Parameters	Values Tested
Number of Trees (n-estimators	100,200,300
Maximum Depth	10,15,20
Minimum Samples per Leaf	1,3,5
Maximum Feature	Sqrt,log2
Criterion	Squared-error

Selection Criterion: Minimum validation Mean Squared Error (MSE)

Table A2. LSTM Model Configuration

Component	Specification
Input time Steps	21 days
Number LSTM layer	2
Units per layer	64,32
Dropout rate	0.2
Activation function	Tanh
Optimizer	Adam
Learning rate	0.001
Batch size	64
Maximum Epochs	100 (with early stopping
Early stopping patience	5 Epoch
Loss function	Mean Squared Error (MSE)

Table A3. Hybrid GARCH-LSTM Configuration

Component	**Specification**
Additional inputs	EGARCH-t conditional volatility
	Lagged returns
	Rolling 21 day realized volatility
Rolling window for GARCH Estimation	504 trading days (≈ 2 years)
Forecast Re-estimation frequency	Every 252 trading days

Table A4. Reproducibility settings

Component	**Specification**
Python version	3.11
Random seed	42
Libraries used	Arch
	Statsmodels
	Scikit-learn
	Numpy
	Pandas
	PyTorch

ETHICAL AND SCIENTIFIC PRINCIPLES STATEMENT OF RESPONSIBILITY

The author(s) declare that ethical principles and scientific citation principles were adhered to throughout the preparation of this study. OJOMSTE assumes no responsibility if a contrary finding occurs; all responsibility belongs to the authors.

STATEMENT OF RESEARCHERS' CONTRIBUTION TO THE ARTICLE

First author contribution rate: 40%

Second author contribution rate: 35%

3rd author contribution rate: 25%