Volatility Forecasting in Financial Time-Series: A Comparative Study of GARCH, XGBoost, and LSTM Models

Preprint · October 2025

DOI: 10.13140/RG.2.2.16753.95848

5 authors, including:

Arsh Misra
Columbia University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Tiancheng Zhou
Columbia University
9 PUBLICATIONS **112** CITATIONS
SEE PROFILE

Volatility Forecasting in Financial Time-Series: A Comparative Study of GARCH, XGBoost, and LSTM Models

Arsh Misra, Xiang Si, Muqing Wen, Tiancheng Zhou, and Xinyao Han
Columbia University

Abstract

Forecasting market volatility is essential for pricing derivatives, managing risk, and designing trading strategies. This study compares traditional econometric and modern machine learning approaches to model S&P 500 volatility over the period 2018–2024, a period that includes several high-stress market regimes such as the COVID-19 pandemic. We evaluated a conventional GARCH(1,1) model against two data-driven architectures: XGBoost, a gradient-boosted ensemble method, and LSTM, a recurrent neural network capable of capturing temporal dependencies. All models are trained in rolling windows of daily returns derived from the price data of S&P 500 and evaluated using R^2, RMSE, and MAPE. Our results show that both deep learning models substantially outperform GARCH, with the LSTM achieving the strongest out-of-sample performance ($R^2 = 0.962$) and the lowest prediction error (MAPE $= 0.066$). These findings highlight the capacity of nonlinear sequence and ensemble methods to capture volatility clustering and structural regime shifts that traditional models fail to represent. We conclude that hybrid frameworks combining econometric interpretability with machine learning flexibility offer a promising direction for robust volatility forecasting in dynamic markets.

Keywords: Volatility forecasting, GARCH, XGBoost, LSTM, financial time series, deep learning

I. Introduction

Volatility is a key measure that serves as an essential tool for strategies in arbitrage, derivative pricing, portfolio management, and hedging. Financial institutions and funds have long sought ways to accurately forecast volatility in order to capitalize on market cycles or anticipate periods of financial turmoil. Traditional approaches such as the GARCH model [1] have been the conventional choice for decades. However, in recent years, deep learning models have entered the spotlight. These algorithms have increasingly been applied to enhance the accuracy and reliability of the prediction of stock market volatility [5], prompting discussion of their comparative performance against classical econometric models.

This study evaluates one traditional model (GARCH) and two deep learning models, XGBoost [2] and LSTM [3], in predicting S&P 500. The time horizon spans January 2018 to December 2024, encompassing multiple black-swan events such as the COVID-19 pandemic. This period provides a dynamic environment across distinct volatility regimes for robust comparative analysis.

II. Research Objectives and Data Preprocessing

This study aims to address the following research objectives:

1) Construct three robust predictive models to forecast volatility for the S&P 500 Index.
2) Evaluate the performance of the model using three key metrics: R^2, the root mean square error (RMSE) and the mean absolute percentage error (MAPE).
3) Compare the respective performance of the models to determine the most viable forecasting approach.

We collected historical daily price and volume data for the S&P 500 from January 4, 2018 to December 30, 2024, using the Tushare API[6]. Volatility, being directly related to return fluctuations, was modeled using daily returns defined as:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \tag{1}$$

rolling 22-day volatility computed as:

$$vol_t = \sqrt{\frac{1}{N-1}\sum_{i=t-N+1}^{t}(r_i - \bar{r})^2}. \tag{2}$$

We also derived features such as percentage change and swing to capture short-term price dynamics.

III. Models

A. GARCH

The distribution of S&P 500 daily returns resembles a Generalized Error Distribution (GED), deviating from normality with leptokurtosis and mild negative skewness. It features a sharp central peak, indicating frequent small fluctuations, and fat tails, suggesting a higher probability of extreme returns, particularly beyond ± 0.1. Traditional linear models such as ARIMA often struggle to capture these properties due to their assumption of homoskedasticity, whereas GARCH models effectively handle time-varying volatility by modeling conditional variance based on past squared returns and variances. This makes GARCH well-suited for forecasting volatility in

financial time series, accommodating both non-normality and volatility clustering inherent in SPX daily returns.

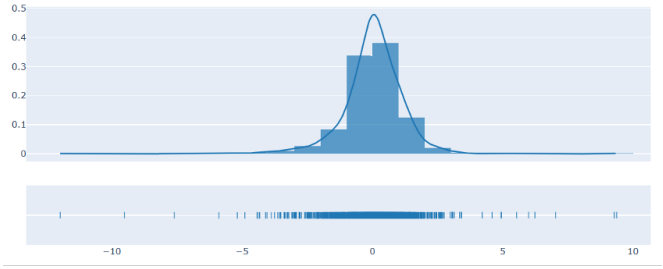

Fig. 1. Distribution of S&P 500 daily returns showing leptokurtosis and fat tails.

We selected GARCH(1,1) as the final model after balancing model complexity and predictive accuracy. To determine optimal lag orders, we adopted a two-step approach: first, analyzing the Partial Autocorrelation Function (PACF) of squared returns to identify volatility clustering and short-memory effects; and second, minimizing the Akaike Information Criterion (AIC) to evaluate model fit and complexity.

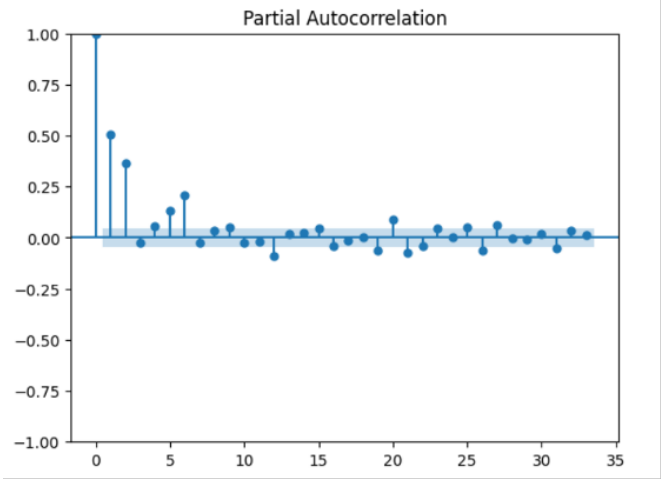

Fig. 2. Partial autocorrelation of squared returns used for lag selection in GARCH.

The PACF of squared returns showed significant spikes at low lags (particularly at lags 1 and 2), indicating short-memory effects and favoring lower-order GARCH terms. A grid search over GARCH(p, q) models ($1 \leq p, q \leq 5$) using AIC confirmed that GARCH(1,1) achieved the lowest AIC value, representing the best trade-off between fit and simplicity. While higher-order models such as GARCH(4,4) offered slightly better dynamic capture, residual diagnostics indicated only marginal improvements, reinforcing GARCH(1,1) as the more efficient model.

To adapt to changing market conditions, we implemented a rolling forecasting approach in which the GARCH model is re-estimated at each step using the most recent data within an expanding window. This dynamic updating allows the model to capture regime shifts more effectively, improving predictive accuracy in non-stationary environments.

B. XGBoost

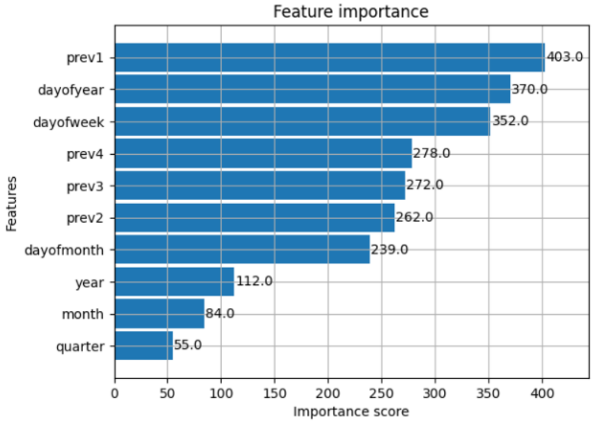

Fig. 3. Feature importance ranking for the XGBoost model.

XGBoost is a tree-based ensemble learning algorithm optimized for speed and performance. It models nonlinear relationships between engineered time features and future volatility. Our XGBoost model adopts several key hyper-parameters to balance model complexity and generalization performance. The most important are:

- `learning_rate` = 0.3 — controls the contribution of each tree to the final prediction; a higher rate speeds up learning but risks overfitting.
- `max_depth` = 6 — limits tree complexity; deeper trees model more interactions but may overfit, while six provides a practical trade-off.

We constructed features that capture both auto-regressive structure and seasonal calendar patterns, enabling the model to learn temporal dependence from recent fluctuations. Lagged returns and volatility capture short-term memory effects by incorporating recent fluctuations, reflecting volatility clustering. Calendar features (*dayofweek*, *dayofyear*) capture systematic, seasonally driven influences such as heightened uncertainty on Mondays or month-end.

Feature-importance analysis (Figure 3) identifies "pre_1", "dayofyear", and "dayofweek" as the top predictors. This indicates that the model relies heavily on recent market conditions, confirming its ability to incorporate both short-term memory and cyclical market behaviors.

C. LSTM

Long Short-Term Memory (LSTM) networks belong to the recurrent neural network (RNN) family and are designed to learn long-term temporal dependencies via a system of gates: forget, input, and output. Unlike traditional RNNs, which suffer from the vanishing-gradient problem, LSTMs can retain and discard information selectively, making them effective for time-series volatility prediction.

The architecture was tuned by experimenting with layer depth, neuron count, dropout rate, and input feature combinations. The best-performing configuration used:

- Features: Four previous days of true lagged volatility.

- Architecture: One LSTM layer (64 neurons) and one dense output layer.
- Dropout rate: 0.2.

We employed rolling true 22-day volatility with a 10-day look-back under a walk-forward validation scheme to prevent data leakage. Training and test sets were split before scaling; applying the scaler afterward would introduce training data into the test set and invalidate results. The 10-day look-back was constructed carefully to prevent the LSTM from absorbing patterns from previous look-backs. In essence, the model uses the 10-day look-back to predict day 11, but the memory of the 11th prediction does not carry over to the 12th, thereby avoiding circular dependencies and ensuring valid, independent forecasts.

IV. Results and Evaluation

After training and testing the models and calculating the relevant evaluation metrics, we obtained the following results.

TABLE I
Model Training / Fitting Scores

Metric	GARCH	XGBoost	LSTM
R^2	0.8194	0.9925	0.9820
RMSE	0.2944	0.0689	0.0270
MAPE	0.1890	0.4440	0.0569

TABLE II
Model Prediction Scores

Metric	GARCH	XGBoost	LSTM
R^2	0.3116	0.9484	0.9619
RMSE	0.1728	0.0633	0.0127
MAPE	0.1704	0.0463	0.0660

The GARCH model performed worse than expected, with a fitting R^2 of 0.819 and a prediction R^2 of 0.312. This may be because the model struggles to differentiate between one-off shocks and sustained volatility periods, hence the lower training score. The out-of-sample performance further highlights this limitation, as the GARCH model showed limited alignment with actual volatility, especially during regime shifts. Its forecasts are based on static, one-step-ahead dynamics that fail to adjust rapidly to structural breaks, resulting in a mismatch.

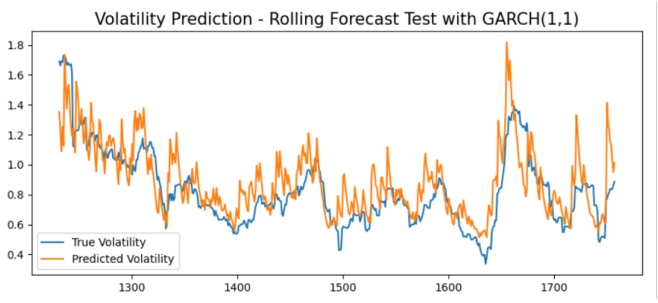

Fig. 4. Out-of-sample GARCH forecast vs. realized volatility.

The best-performing models turn out to be of the deep-learning variety, with the LSTM achieving training and test scores of 0.982 and 0.962, respectively, and the XGBoost model trailing closely behind with scores of 0.993 and 0.948. These are quite strong results; given the high degree of auto-correlation in volatility data, it is not surprising that the LSTM effectively captured these temporal dependencies. The MAPE corroborates this finding as the predicted versus actual values differ by only 0.066 on average. As illustrated in Figure 6, the LSTM output closely tracks the volatility realized, even between regime changes.

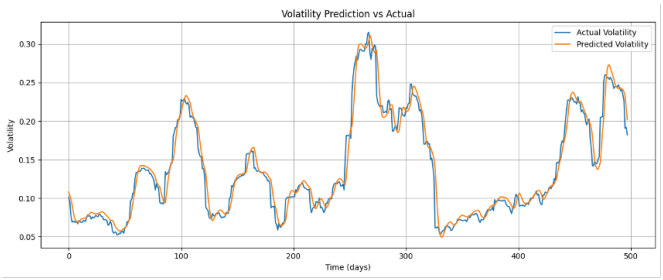

Fig. 5. LSTM forecast performance.

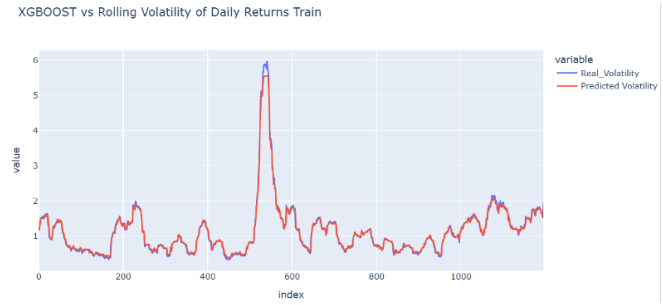

Fig. 6. XGBoost vs. rolling volatility of daily returns.

The XGBoost model also demonstrates strong training performance, closely tracking real volatility throughout the training period. It accurately replicates both baseline and spiking regimes, including extreme market shocks, reflecting the model's effectiveness in capturing short-term dependencies and seasonality patterns. Although slight discrepancies emerge around inflection points, preventing it from outperforming the LSTM, the predicted path remains stable and responsive. This indicates solid short-term generalization, especially in low-volatility environments where structural patterns persist.

V. Conclusion

Overall, this study reveals that modern sequence and ensemble methods can substantially enhance predictive accuracy under varied market conditions. By integrating temporal memory through LSTM and capturing nonlinear interactions via XGBoost, we achieved more resilient forecasts than the traditional GARCH, particularly during abrupt regime shifts. This performance gap underscores the importance of flexible model architectures that adapt to both persistent autocorrelation and extreme events. Our findings also reinforce prior

research such as Campisi *et al.* (2023), which found that machine-learning models significantly outperform classical architectures for volatility prediction.

However, several limitations should be noted. Our results hinge on a specific sample period (Jan. 2018–Dec. 2024) and may not generalize to other crisis regimes or longer horizons. The high accuracy of XGBoost and LSTM raises concerns about overfitting despite walk-forward validation. Deep-learning approaches also demand extensive hyperparameter tuning and computational resources, while limiting inputs to lagged volatility and calendar features may omit critical exogenous drivers such as market sentiment or liquidity. Future work should address these limitations by testing hybrid GARCH–neural architectures, expanding to intraday and alternative asset classes, and incorporating exogenous variables to improve both interpretability and robustness.

Our results prompt several directions for further study: Can hybrid GARCH–neural frameworks (e.g., Dessie *et al.*, 2025) blend economic interpretability with deep learning's adaptability? How would these models scale to intraday data or alternative assets such as commodities and cryptocurrencies? Additionally, incorporating exogenous signals, like market sentiment or liquidity measures, may further refine volatility forecasts and bridge the gap between theoretical advances and robust real-world applications.

REFERENCES

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

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

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

[4] G. Campisi, S. Muzzioli, and B. De Baets, "A comparison of machine learning methods for predicting the direction of the US stock market on the basis of volatility indices," *Int. J. Forecasting*, vol. 40, pp. 869–880, 2023.

[5] J. Zhang, "Some application of machine learning in quantitative finance: Model calibration, volatility formation mechanism and news screening," Ph.D. dissertation, Institut Polytechnique de Paris, 2023.

[6] Tushare, "Tushare Pro API Documentation," available at: https://tushare.pro/document/2?doc_id=211, accessed Jan. 2025.