Return-Based Volatility Proxy Forecasting via a Residual-Correction LSTM–GARCH Hybrid Framework

Zhenxiang Zhang

School of Science, Shanghai University, Shanghai, China

17317623218@163.com

Abstract. Forecasting return-based volatility proxies is challenging in financial markets due to volatility clustering, heavy tails, and nonlinear dynamics. This paper proposes a residual-correction LSTM–GARCH hybrid framework, where GARCH provides a robust econometric backbone, while LSTM learns the nonlinear residual in the log variance-ratio space and performs multiplicative correction. Experiments on a representative ETF show that the proposed RC-LSTM–GARCH achieves better RMSE, MAE, and QLIKE than the GARCH baseline. The proposed design aims to combine the interpretability and stability of GARCH with the nonlinear representation ability of LSTM, thereby improving calibration under volatility-specific losses, and the framework can be extended to multi-asset and multi-horizon risk forecasting.

Keywords: Stock return forecasting, residual correction, GARCH, LSTM.

1. Introduction

Volatility proxy forecasting from returns is a core problem in quantitative finance, with direct implications for risk management, portfolio allocation, derivative pricing, and regulatory stress testing. In practice, financial returns exhibit volatility clustering, heavy tails, nonlinear regime shifts, and sensitivity to news or policy shocks, which together make short-horizon volatility forecasting persistently challenging. Despite decades of progress, building models that are both accurate and robust across assets and market regimes remains an open applied problem.

A representative line of research addresses stock-market uncertainty by modeling conditional heteroskedasticity with econometric volatility models. Engle [1] proposed the ARCH framework to capture time-varying volatility in return series, enabling volatility forecasting under clustering effects, and Bollerslev [2] generalized it to GARCH to improve persistence modeling and forecasting stability. Subsequent extensions such as EGARCH [3] and asymmetric specifications (e.g., GJR-type effects) [4] further improve the ability to describe leverage effects and regime-dependent dynamics commonly observed in equity markets. Since the realized variance is latent, practical evaluation often relies on noisy volatility proxies (e.g., squared returns), and Patton [9] proposed a principled comparison framework using loss functions that remain informative under imperfect proxies, which motivates the use of volatility-oriented metrics such as QLIKE in empirical studies.

In parallel, deep learning has become increasingly popular for modeling nonlinear temporal dependencies in financial time series. Hochreiter and Schmidhuber [6] introduced LSTM networks to learn long-range sequential patterns, which provides a flexible tool for capturing nonlinearities beyond parametric volatility models. Building on this idea, Kim and Won [11] proposed hybrid models that integrate LSTM with multiple GARCH-type specifications for stock index volatility forecasting, aiming to combine econometric structure with nonlinear representation. Hu et al. [12] further demonstrated that coupling deep recurrent predictors with GARCH-type backbones can improve volatility prediction in commodity markets, and Micha'nk'ow et al. [13] provided a comparative study showing that hybrid solutions may improve point volatility forecasts under certain settings.

However, many existing hybrids rely on direct volatility regression or static linear ensembling, which may underutilize the complementary strengths of an interpretable econometric backbone and a nonlinear residual learner. In particular, such designs may offer limited interpretability of when and how strongly the neural component adjusts the baseline, and the correction may not be scale-

consistent across volatility regimes. To address these limitations, we propose RC-LSTM–GARCH, a residual-correction hybrid that treats GARCH as a robust and interpretable backbone and trains an LSTM to learn the remaining nonlinear component in the log variance-ratio residual space, yielding a multiplicative correction that preserves non-negativity and provides time-varying, scale-consistent adjustments. Our approach is aligned with recent efforts that inject GARCH structure into neural models [14], [15] while keeping the hybridization simple and reproducible. Beyond aggregate error measures, we analyze the correction mechanism through the correction ratio distribution and its quantiles, providing an interpretable diagnostic of whether the hybrid behaves as mild calibration or aggressive regime-dependent adjustment.

The remainder of this paper is organized as follows: Section II describes the data construction and preprocessing, Section III presents the proposed methodology and baselines, Section IV introduces the evaluation metrics, Section V reports experimental results and analysis, followed by discussion in Section VI and conclusions in Section VII.

2. Data and Preprocessing

2.1. Data source and notation

We use daily OHLCV records of Guolianan CSI All-Share Semiconductor ETF (ticker: 512480) over the sample period 2019-06-13 to 2026-02-13, downloaded from the Eastmoney financial data platform and archived locally for reproducibility.

Table 1. Top holdings of the Guolianan CSI All-Share Semiconductor ETF (512480).

Ticker	Company	Weight (%)	Semiconductor value-chain segment
688256	Cambricon	8.82	AI chip / processor design
688981	SMIC	7.59	Foundry / wafer fabrication
688041	Hygon	6.45	CPU / HPC chip design
002371	NAURA	6.14	Semiconductor equipment
603986	GigaDevice	4.42	Memory & MCU design
688008	Montage	4.17	Interface / memory controller chips
688012	AMEC	3.95	Semiconductor equipment
603501	Will Semi	3.29	CMOS image sensors / analog
688072	ACM Research	1.72	Semiconductor equipment
002049	Unigroup Guowei	1.67	Security IC / special-purpose chips

Table 1 reports the top holdings of Guolianan CSI All-Share Semiconductor ETF with their portfolio weights and semiconductor value-chain segments.

2.2. Return and volatility proxy

Let $t \in \{1,2,\dots\}$ denote the trading-day index, P_t denote the closing price on day t. The log return is defined as

$$r_t = \log(P_t) - \log(P_{t-1}),$$

where r_t is the log return from day $t-1$ to day t, and P_{t-1} is the closing price on the previous trading day.

Following common practice, we define the one-step-ahead return-based volatility proxy as the squared next-day return:

$$y_t = \mathrm{RV}_{t+1} = r_{t+1}^2,$$

where y_t is the prediction target aligned to time t, RV_{t+1} denotes the realized-variance proxy at day $t+1$, and r_{t+1}^2 is the squared log return on the next trading day.

2.3. Supervised sequence construction and split

We construct LSTM inputs using a rolling window of length $k = 20$:

$$X_t = [r_{t-k+1}, \dots, r_t] \in \mathbb{R}^{k\times 1},$$

where X_t is the feature sequence at time t consisting of the most recent k returns, and $\mathbb{R}^{k\times 1}$ denotes a univariate sequence of length k.

After alignment and removing missing values, samples are split chronologically into train/validation/test subsets with a 70%/15%/15% ratio to prevent information leakage. In the reported run, the test set contains 241 samples.

3. Methodology

This section introduces the baseline models and the proposed residual-correction hybrid. All one-step-ahead forecasts are aligned to the target $y_t = r_{t+1}^2$ defined in (2).

3.1. GARCH baseline

We fit a GARCH(1,1) model with Student-t innovations on scaled returns $\tilde{r}_t = c\, r_t$:

$$\tilde{r}_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim t_\nu(0,1),$$
$$\sigma_t^2 = \omega + \alpha \tilde{r}_{t-1}^2 + \beta \sigma_{t-1}^2,$$

where $\tilde{r}_t$ is the scaled return. σ_t^2 is the conditional variance of $\tilde{r}_t$, ϵ_t is the standardized innovation following a Student-t distribution with degrees of freedom ν, and $\omega > 0$, $\alpha \geq 0$, and $\beta \geq 0$ are model parameters estimated by maximum likelihood.

Let $\hat{\sigma}_{t+1|t}^2$ denote the one-step-ahead forecast of σ_{t+1}^2 using information up to time t and we set $c = 100$ to express returns in percentage units. The corresponding variance forecast on the original (unscaled) return scale is

$$\hat{v}_t^{\mathrm{G}} = \hat{\sigma}_{t+1|t}^2 / c^2,$$

where $\hat{v}_t^{\mathrm{G}}$ is the GARCH-based prediction of the next-day variance proxy aligned at time t.

3.2. Direct LSTM baseline

We train an LSTM to predict the volatility proxy from the lagged return window X_t in (3). For numerical stability, we regress on the transformed target

$$\tilde{y}_t = \log(1 + y_t),$$

where $\tilde{y}_t$ is the transformed training label and $y_t \geq 0$. Let $\hat{\tilde{y}}_t$ denote the LSTM prediction of $\tilde{y}_t$ at time t. The prediction is mapped back to the original scale via

$$\hat{y}_t^{\mathrm{L}} = \exp(\hat{\tilde{y}}_t) - 1,$$

where $\hat{y}_t^{\mathrm{L}}$ is the direct LSTM forecast of y_t. Non-negativity is ensured by the inverse mapping in (8).

3.3. Linear mixture (ensemble)

We consider a convex combination of the GARCH and LSTM forecasts:

$$\hat{y}_t^{\mathrm{Mix}} = w \hat{v}_t^{\mathrm{G}} + (1 - w)\hat{y}_t^{\mathrm{L}}, \quad w \in [0,1],$$

where $\hat{y}_t^{\mathrm{Mix}}$ is the ensemble forecast and w is a scalar weight. In this paper, we choose $w = 0.5$ as a baseline.

3.4. RC-LSTM–GARCH

RC-LSTM–GARCH models the log variance-ratio residual between the realized proxy and the GARCH forecast:

$$z_t = \log\left(\frac{y_t}{\hat{v}_t^{\mathrm{G}}}\right),$$

where z_t is the log ratio residual (positive values indicate underestimation by GARCH, and negative values indicate overestimation). We standardize z_t using the training-set mean μ_z and standard deviation $\sigma_z > 0$:

$$z_t^{(s)} = \frac{z_t - \mu_z}{\sigma_z},$$

where $z_t^{(s)}$ is the standardized residual.

An LSTM with linear output is trained to predict $z_t^{(s)}$ from X_t. Let $\hat{z}_t^{(s)}$ denote the predicted standardized residual; we recover the residual on the original scale as

$$\hat{z}_t = \hat{z}_t^{(s)}\sigma_z + \mu_z,$$

where $\hat{z}_t$ is the de-standardized residual forecast. Finally, we apply multiplicative correction to obtain the RC-LSTM–GARCH forecast:

$$\hat{y}_t^{\mathrm{A}} = \hat{v}_t^{\mathrm{G}} \cdot \exp(\hat{z}_t),$$

where $\hat{y}_t^{\mathrm{A}}$ is the residual-corrected volatility proxy forecast.

3.5. Algorithm description (pseudo-code)

Algorithm 1 presents the pseudo-code of the proposed RC-LSTM–GARCH framework.

Algorithm 1. RC-LSTM–GARCH residual-correction LSTM–GARCH forecasting

Require: Price series $\{P_t\}$, window length k, split ratio (train, val, test)

Ensure: One-step-ahead forecasts $\hat{v}_t^{\mathrm{G}}, \hat{y}_t^{\mathrm{L}}, \hat{y}_t^{\mathrm{Mix}}, \hat{y}_t^{\mathrm{A}}$

1. Compute returns r_t by (1), compute targets y_t by (2).
2. Construct sequences X_t by (3), perform chronological split as in Section II-C.
3. Fit GARCH(1,1)-t on training returns by (4)–(5), obtain $\hat{v}_t^{\mathrm{G}}$ by (6).
4. Transform targets $\tilde{y}_t$ by (7), train direct LSTM on $(X_t, \tilde{y}_t)$, map back to $\hat{y}_t^{\mathrm{L}}$ by (8).
5. Tune weight w on validation by minimizing QLIKE in (16), compute $\hat{y}_t^{\mathrm{Mix}}$ by (9).
6. Construct residuals z_t by (10), standardize by (11).
7. Train residual LSTM on $\left(X_t, z_t^{(s)}\right)$, recover residual forecasts $\hat{z}_t$ by (12).
8. Compute corrected forecasts $\hat{y}_t^{\mathrm{A}}$ by (13).
9. Evaluate forecasts by RMSE, MAE, and QLIKE in (14)–(16).

4. Evaluation Metrics

RMSE and MAE are standard absolute-error measures that summarize average predictive accuracy. QLIKE is a volatility-oriented loss derived from the Gaussian quasi-likelihood form, and it is widely used for comparing volatility forecasts when y_t is an imperfect and noisy proxy of latent variance, providing a scale-sensitive and strictly proper scoring rule for variance forecasts under common settings. Thus, we evaluate one-step-ahead forecasts using RMSE, MAE, and QLIKE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2},$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t|,$$

$$\mathrm{QLIKE} = \frac{1}{N}\sum_{t=1}^{N}\left(\frac{y_t}{\hat{y}_t} - \log\frac{y_t}{\hat{y}_t} - 1\right),$$

where N is the number of test samples, y_t is the realized volatility proxy defined in (2), and $\hat{y}_t$ denotes a generic one-step-ahead forecast of y_t produced by a given model at time t.

5. Experiments and Results

5.1. Main results

We evaluate one-step-ahead return-based volatility proxy forecasting on the Guolianan CSI All-Share Semiconductor ETF using daily OHLCV data over 2019-06-13 to 2026-02-13. Following the chronological split described in Section II-C (70%/15%/15%). Performance is reported on the held-out test set, and we compare four methods, including the GARCH(1,1)-t baseline, a direct LSTM baseline, a fixed-weight linear mixture with $w = 0.5$ (9), and the proposed RC-LSTM–GARCH hybrid.

Table 2. Evaluation of model performance for one-step-ahead volatility proxy forecasting on the Guolianan CSI All-Share Semiconductor ETF (512480).

Model	RMSE ↓	MAE ↓	QLIKE ↓
GARCH(1,1)-t	0.001073	0.000575	1.827691
LSTM (direct)	0.001107	0.000707	1.945382
Linear Mix ($w = 0.5$)	0.001097	0.000696	1.923728
RC-LSTM–GARCH	0.001072	0.000569	1.800505

Based on the test-set results in Table 2, the RMSE, MAE, and QLIKE of RC-LSTM–GARCH respectively are 0.001072, 0.000569 and 1.800505, which achieve the best overall performance among all compared methods. Relative to the GARCH(1,1)-t baseline, the residual-correction mechanism consistently reduces both error-based metrics (RMSE, MAE) and the volatility-oriented loss (QLIKE), indicating improved calibration of variance forecasts.

5.2. Correction strength analysis

To interpret the correction behavior, we analyze the ratio between the corrected forecast and the GARCH backbone forecast. This dimensionless quantity measures the multiplicative correction strength in a scale-consistent way, with values below (above) one indicating down-scaling (up-scaling).

$$\mathrm{Ratio}_t = \frac{\hat{y}_t^{\mathrm{A}}}{\hat{v}_t^{\mathrm{G}}}.$$

Table 3. Quantiles of the correction ratio on the test set.

Statistic	Quantile	Value
Minimum	min	0.6085
Lower tail	1%	0.6396
Lower tail	5%	0.7130
Median	50%	0.9310
Upper tail	95%	1.4519
Upper tail	99%	1.8622
Maximum	max	2.0354

Based on Table 3, the correction ratio is typically below one, with a median of 0.9310, indicating that RC-LSTM–GARCH down-scales the GARCH variance forecast in most test instances. At the same time, the upper tail is heavy, as the 95% and 99% quantiles reach 1.4519 and 1.8622, and the maximum exceeds 2.0, suggesting occasional strong up-scaling. This asymmetric distribution implies that the residual-correction module mainly acts as a mild calibration of the GARCH backbone under normal regimes, while retaining the ability to amplify forecasts during rare volatility spikes, thereby improving responsiveness to extreme market conditions.

5.3. Visualization

To complement the quantitative results reported in the previous section, we provide a visual comparison on the test set. We plot the realized volatility proxy y_t (2) together with the GARCH backbone forecasts $\hat{v}_t^{\mathrm{G}}$ (6) and the proposed RC-LSTM–GARCH forecasts $\hat{y}_t^{\mathrm{A}}$ (13). We further visualize the correction ratio $\hat{y}_t^{\mathrm{A}}/\hat{v}_t^{\mathrm{G}}$ (17) over time to illustrate the multiplicative adjustment mechanism. The figures are generated in Python, where NumPy and pandas are used for data handling and alignment, and matplotlib is used for plotting.

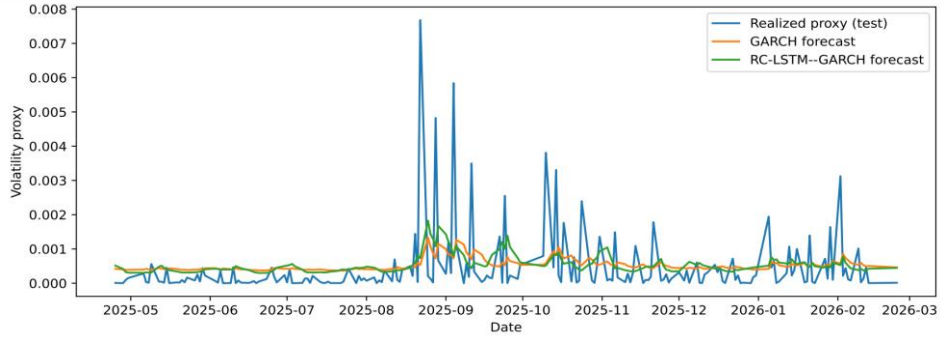


Fig. 1 Test-set forecasts of the volatility proxy y_t (2): realized proxy versus GARCH $\hat{v}_t^{\mathrm{G}}$ (6) and RC-LSTM–GARCH $\hat{y}_t^{\mathrm{A}}$ (13).

As shown in Fig. 1, the realized volatility proxy y_t (2) exhibits pronounced spike-like fluctuations, reflecting the noisy nature of squared-return proxies. Both GARCH $\hat{v}_t^{\mathrm{G}}$ (6) and RC-LSTM–GARCH $\hat{y}_t^{\mathrm{A}}$ (13) produce substantially smoother forecasts, capturing the underlying volatility level rather than fitting individual extreme spikes. Compared with the GARCH backbone, RC-LSTM–GARCH reacts more strongly during high-volatility intervals by moderately lifting or suppressing the baseline forecast, while remaining close to GARCH in calm periods. This behavior suggests that the residual-correction module mainly performs a time-varying calibration of the backbone and improves responsiveness when volatility regimes change.

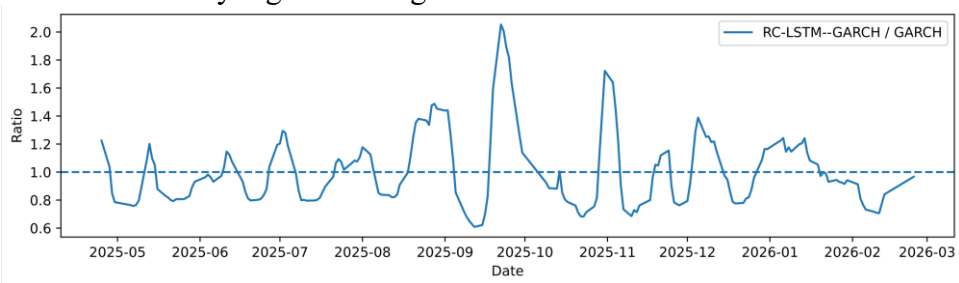


Fig. 2 RC-LSTM–GARCH correction ratio over time.

Fig. 2 further illustrates the correction mechanism through the ratio $\hat{y}_t^{\mathrm{A}}/\hat{v}_t^{\mathrm{G}}$ (17). The ratio oscillates around one for most dates, indicating that the hybrid often applies mild multiplicative adjustments to the GARCH forecast. Notably, several peaks above one correspond to periods where the residual module amplifies the baseline, consistent with the goal of capturing volatility surges, whereas dips below one indicate down-scaling when the backbone is relatively overestimated. Overall, the temporal variation of the ratio provides an interpretable diagnostic of when RC-LSTM–GARCH behaves as a calibration tool versus an aggressive regime-dependent adjustment.

6. Discussion

Although RC-LSTM–GARCH improves the out-of-sample metrics on the Guolianan CSI All-Share Semiconductor ETF, the proposed pipeline increases implementation and computational complexity relative to a single econometric baseline, as it requires fitting the GARCH backbone in (4)–(5) and training an additional residual LSTM module for correction. While such behavior may be beneficial for capturing abrupt volatility increases, it may also reduce robustness across regimes or assets with different noise characteristics.

Several directions may further strengthen the proposed framework. One practical extension is to regularize the residual module to control extreme corrections, for example by constraining the predicted residual $\hat{z}_t$ (cf. (12)) or adding penalties on large deviations of the correction ratio in (17). In addition, incorporating richer predictors beyond lagged returns in (3), such as turnover and market-wide risk indicators may improve residual predictability and reduce sensitivity to the noise in (2). Robustness can also be assessed more systematically through rolling-window re-estimatio and multi-seed reporting for the neural components, which would provide clearer evidence on stability and generalization. The residual-correction idea can be transferred to a broad class of time-series forecasting problems such as electricity load, transportation demand and environmental forecasting (e.g., PM2.5 and hydrological variables).

7. Conclusion

This paper proposes RC-LSTM–GARCH, a residual-correction hybrid framework for one-step-ahead return-based volatility proxy forecasting. The key idea is to preserve the interpretability and stability of a GARCH backbone while learning the remaining nonlinear and regime-dependent component through an LSTM in the log variance-ratio space. Different from direct volatility regression or static linear ensembling, RC-LSTM–GARCH models the residual z_t (10) and applies a multiplicative correction as in (13), which naturally preserves non-negativity and provides scale-consistent, time-varying adjustments. This design offers a principled way to combine econometric structure with neural representation: GARCH captures volatility clustering and heavy-tailed innovations via (4)–(5), while the residual learner corrects systematic miscalibration in a transparent ratio form, enabling interpretation via the correction magnitude in (17) and its distribution over time.

We use daily data of Guolianan CSI All-Share Semiconductor ETF with chronological splitting and standard volatility forecasting metrics in (14)–(16). Then we establish three baseline models including the GARCH(1,1)-t baseline, a direct LSTM baseline and a fixed-weight linear mixture with $w = 0.5$ (9) to compare with the proposed RC-LSTM–GARCH hybrid method. We find the RMSE,MAE, and QLIKE of RC-LSTM–GARCH respectively are 0.001072, 0.000569 and 1.800505, which achieve the best overall performance among all compared methods. The ratio analysis further shows that the residual- correction module mainly acts as a mild calibration of the GARCH backbone under normal regimes, while retaining the ability to amplify forecasts during rare volatility spikes, thereby improving responsiveness to extreme market condi- tions.

References

[1] Engle, Robert F. “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation.” Econometrica, vol.~50, no.~4, 1982, pp.~987–1007.

[2] Bollerslev, Tim. “Generalized Autoregressive Conditional Heteroskedasticity.” Journal of Econometrics, vol.~31, no.~3, 1986, pp.~307–327.

[3] Nelson, Daniel B. “Conditional Heteroskedasticity in Asset Returns: A New Approach.” Econometrica, vol.~59, no.~2, 1991, pp.~347–370.

[4] Glosten, Lawrence R., Ravi Jagannathan, and David E. Runkle. “On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks.” The Journal of Finance, vol.~48, no.~5, 1993, pp.~1779–1801.

[5] Hansen, Bruce E. “Autoregressive Conditional Density Estimation.” International Economic Review, vol.~35, no.~3, 1994, pp.~705–730.

[6] Hochreiter, Sepp, and J{"urgen Schmidhuber. “Long Short-Term Memory.” Neural Computation, vol.~9, no.~8, 1997, pp.~1735–1780.

[7] Andersen, Torben G., et al. “Modeling and Forecasting Realized Volatility.” Econometrica, vol.~71, no.~2, 2003, pp.~579–625.

[8] Corsi, Fulvio. “A Simple Approximate Long-Memory Model of Realized Volatility.” Journal of Financial Econometrics, vol.~7, no.~2, 2009, pp.~174–196.

[9] Patton, Andrew J. “Volatility Forecast Comparison Using Imperfect Volatility Proxies.” Journal of Econometrics, vol.~160, no.~1, 2011, pp.~246–256.

[10] Kristjanpoller, Werner, and Mauro C. Minutolo. “Gold Price Volatility: A Forecasting Approach Using the Artificial Neural Network–GARCH Model.” Expert Systems with Applications, vol.~42, no.~20, 2015, pp.~7245–7251.

[11] Kim, Ha Young, and Chang Hyun Won. “Forecasting the Volatility of Stock Price Index: A Hybrid Model Integrating LSTM with Multiple GARCH-Type Models.” Expert Systems with Applications, vol.~103, 2018, pp.~25–37.

[12] Hu, Yan, Jian Ni, and Liu Wen. “A Hybrid Deep Learning Approach by Integrating LSTM-ANN Networks with GARCH Model for Copper Price Volatility Prediction.” Physica A: Statistical Mechanics and its Applications, vol.~557, 2020, article 124907.

[13] Micha'nk'ow, Jakub, {Kwiatkowski, and Janusz Morajda. “Combining Deep Learning and GARCH Models for Financial Volatility and Risk Forecasting.” arXiv, 2023, arXiv:2310.01063. https://arxiv.org/abs/2310.01063.

[14] Xu, Zeda, et al. “GARCH-Informed Neural Networks for Volatility Forecasting.” Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF ’24), ACM, 2024, pp.~600–607.

[15] Zhao, Pengfei, et al. “From GARCH to Neural Network for Volatility Forecast.” arXiv, 2024, arXiv:2402.06642. https://arxiv.org/abs/2402.06642.