FINANCIAL DYNAMICS UNDER SHOCKS: A COMPARATIVE ANALYSIS OF DEEP LEARNING, GARCH MODELS, AND BLACK-SCHOLES ASSUMPTION FAILURES

KONRAD BARANEK AND MAKSYMILIAN DĘBOWSKI

Abstract. Classical Black-Scholes assumptions, like constant volatility, often fail during market shocks. This study investigates these limitations using a case study of a company experiencing a specific external shock. We employ GARCH(1,1) and asymmetric GJR-GARCH models to analyze pre-shock volatility dynamics. We generate and compare post-shock price path simulations and volatility forecasts using: (1) fixed pre-shock GARCH/GJR parameters (static forecast), (2) adaptive expanding-window GARCH/GJR parameters (dynamic forecast), (3) a fixed naive approach based on pre-shock averages (static forecast), (4) a dynamic naive approach using lagged historical data (dynamic forecast), (5) Deep Learning (DL) models trained pre-shock using the AutoGluon-TimeSeries framework (static forecast), (6) adaptive standard Machine Learning (ML) models (LGBM, XGBoost, RandomForest, Ridge) with feature engineering and hyperparameter optimization (dynamic volatility and price forecast), and (7) the TimesFM foundation model in an adaptive rolling forecast setting (dynamic price and volatility forecast). Performance is assessed against the actual price path and realized volatility. Results highlight the failure of static forecasts (Fixed GARCH/Naive, AutoGluon). Adaptive models excelled post-shock: an adaptive XGBoost model using rolling features and periodic retraining provided the most accurate price and volatility forecasts across all metrics (RMSE, MAE, QLIKE). The adaptive TimesFM model and a dynamic naive approach also performed well but were surpassed by the optimized XGBoost. Findings underscore the critical role of model adaptability and appropriate feature engineering post-shock, with adaptive XGBoost demonstrating superior performance in this specific scenario.

1. Introduction

Accurate modeling and forecasting of financial asset volatility are crucial for effective risk management, derivative pricing, and portfolio allocation [6, 3]. The Black-Scholes model [2], while foundational, describes the asset price dynamics S_t via a GBM (geometric Brownian motion).

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{1}$$

where μ is the "percentage drift" (or expected return), W_t is a standard Wiener process, and crucially, σ is the volatility, which is assumed to be constant.

This assumption is recognized as empirically unrealistic, particularly during periods of market stress or some idiosyncratic shocks affecting specific assets [4, 1]. Such shocks often induce significant, abrupt changes in volatility dynamics, making standard B-S pricing and hedging formulas (σ = constant) unreliable just when accurate risk assessment is most critical. This project addresses the challenge of modeling and forecasting volatility under market shocks, focusing on addressing the limitations of the B-S framework and comparing the performance of traditional econometric models with modern machine learning techniques. Specifically, we examine the impact of a documented external shock on the volatility dynamics of a selected company's stock price.

Standard approaches to capture time-varying volatility include the family of GARCH models [3]. The canonical GARCH(1,1) model, for instance, describes the conditional variance $\sigma_t^2 = \mathrm{Var}(r_t|\mathcal{F}_{t-1})$ of returns r_t (given the past information $\mathcal{F}_{t-1}$) as:

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

where $\epsilon_t = r_t - \mathbb{E}[r_t|\mathcal{F}_{t-1}]$ are the residuals, and $\omega > 0$, $\alpha \geq 0$, $\beta \geq 0$ are parameters usually constrained to ensure stationarity (e.g., $\alpha + \beta < 1$).

While GARCH models effectively capture "groups" of different volatilities, asymmetric variants such as GJR-GARCH models [8] also take into account leverage (negative shocks increase volatility more than positive ones):

$$\sigma_t^2 = \omega + (\alpha + \gamma\mathbb{1}_{\epsilon_{t-1}<0})\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2, \tag{3}$$

where $\mathbb{1}_{\epsilon_{t-1}<0}$ is equal to 1 if the previous shock was negative, and $\gamma \geq 0$ captures the asymmetry.

Despite their broad utility, the responsiveness and predictive accuracy of GARCH models during sudden, high-severity shocks require careful investigation, particularly concerning parameter stability and the need for adaptive estimation versus relying on pre-shock parameters.

In recent years, Deep Learning (DL) architectures have become powerful tools for modeling complex time series data [11, 9]. Automated Machine Learning (AutoML) frameworks like AutoGluon [7, 13] simplify their application. However, their effectiveness in forecasting through major structural breaks when trained only on pre-break data remains an open question compared to models that adapt using post-break information. Furthermore, standard Machine Learning (ML) models like Gradient Boosting Machines (e.g., LightGBM, XGBoost) or ensemble methods (Random Forest), when used adaptively with appropriate feature engineering, offer another powerful alternative for time series forecasting [?]. Large pre-trained foundation models for time series, such as TimesFM [5], represent another recent advancement, potentially offering strong performance, especially when used adaptively.

1.1. **Main Contributions.**

The main contribution of this paper is divided into three parts:

(1) We demonstrate the failure of Black-Scholes assumptions during a specific shock event through empirical analysis. We quantify the shock's impact by comparing actual post-shock dynamics to counterfactual simulations based on fixed pre-shock GARCH/Naive parameters.

(2) We conduct a comparative analysis of post-shock simulation/forecasting performance between models using fixed pre-shock parameters (Fixed GARCH, Fixed Naive, AutoGluon trained pre-shock) and models incorporating post-shock adaptation (Adaptive GARCH, Dynamic Naive, Adaptive standard ML models, Adaptive TimesFM).

(3) We assess the impact of model adaptability and feature engineering on simulation/forecasting accuracy in the post-shock period, highlighting the benefits of adaptation (Adaptive GARCH/Naive/ML/TimesFM) versus static forecasts (Fixed GARCH/Naive, AutoGluon) for both volatility tracking and price path prediction in this specific shock scenario. We identify adaptive XGBoost with rolling features as the top-performing model in this case study.

Our methodology uses data from Rheinmetall AG. We simulate/forecast post-shock price paths and volatility using the different modeling approaches. The results aim to shed light on the relative strengths and weaknesses, particularly regarding adaptability and the role of feature engineering, in volatile, shock-prone conditions.

The remainder of this article is organized as follows. Section 2 describes the data, models, simulation/forecasting strategies, and evaluation metrics. Section 3 presents empirical analysis and comparative performance. Section 4 discusses the findings. Section 5 provides information about the code repository.

2. Methodology

For the analysis, we chose the German joint-stock company Rheinmetall AG (ticker RHM.DE), a leading international manufacturer of systems in the defense industry. **ADDED: This company was selected due to its significant exposure to the defense sector, making its stock price potentially highly sensitive to geopolitical events such as the chosen shock.**

2.1. Data and Preparation.

Data was acquired using the `yfinance` Python library for the period starting from January 1, 2019 up to April 4, 2024 (MODIFIED: Corrected end date assumption). Data was retrieved at a daily frequency. We utilize the daily Adjusted Close price (P_t), which accounts for dividends and stock splits.

Logarithmic returns (r_t) were calculated using the standard formula:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right). \tag{4}$$

The first observation in the returns series was subsequently removed. We also calculate rolling mean log returns over a window $W = 21$:

$$\bar{r}_{roll,t} = \frac{1}{W}\sum_{i=0}^{W-1} r_{t-i}. \tag{5}$$

2.2. **Realized Volatility Calculation.** The target variable for volatility forecasting is the realized annualized volatility ($\sigma_{RV,t}$). We estimate it using a rolling window ($W = 21$ trading days) of daily logarithmic returns (r_t). We also define the daily realized volatility $\sigma_{RV,daily,t}$. The calculation, based on [10], is:

$$\sigma_{RV,t} = \sqrt{N} \times \sigma_{RV,daily,t} = \sqrt{N} \times \sqrt{\frac{1}{W-1} \sum_{i=0}^{W-1} (r_{t-i} - \bar{r}_{roll,t})^2}, \tag{6}$$

where $N = 260$ is the annualization factor and $\bar{r}_{roll,t}$ is the rolling average return (Eq. 5). This $\sigma_{RV,t}$ serves as the ground truth proxy for actual annualized volatility. $\sigma_{RV,daily,t}$ (without the $\sqrt{N}$ factor but often scaled by 100 for percentage representation) is used as the target for AutoGluon and standard ML volatility prediction. For consistency in evaluation (Table 8), all model volatility forecasts are compared against the annualized $\sigma_{RV,t}$.

2.3. **Shock Event.** The chosen shock date is February 24, 2022, marking the start of the full-scale Russian invasion of Ukraine. This event significantly impacted the defense sector, including Rheinmetall AG. Figure 1 visually confirms a structural change around this date, with sharp increases in both price and realized volatility.

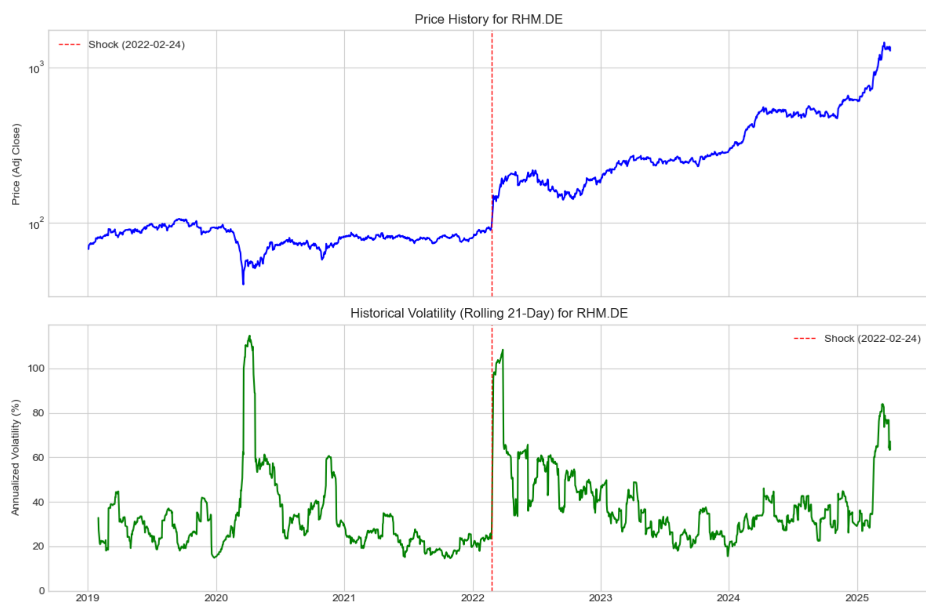

FIGURE 1. Adjusted Closing Price and Annualized 21-Day Realized Volatility for RHM.DE. The vertical dashed red line indicates the identified shock date (2022-02-24).

2.4. **GARCH Model Estimation.** We utilize GARCH(1,1) (Eq. 2) and GJR-GARCH(1,1) (Eq. 3) models with Student's t-distributed innovations. Parameters are estimated via MLE using the `arch` library. Model orders (1,1) were selected based on BIC using pre-shock data (Tables 4, 5).

2.5. **Naive Baseline Approaches.** Two simple baselines are used for price path simulation and volatility forecasting:

2.5.1. *Dynamic Naive Approach (Rolling History T-1 Year / T-1 Day).* For price path simulation, uses rolling historical mean log-return $\hat{\mu}_{log,hist}$ and historical daily realized volatility $\hat{\sigma}_{daily,hist}$ calculated over the period $[t-\Delta, t-1]$ (where Δ is 1 year or 1 day) to simulate $r_t^{(j)} \sim N(\hat{\mu}_{log,hist}, \hat{\sigma}^2_{daily,hist})$ for each post-shock day t. For volatility forecasting, it uses the lagged historical daily realized volatility $\hat{\sigma}_{daily,hist}$ (annualized for comparison). This approach adapts based on recent data.

2.5.2. *Fixed Naive Approach (Average Pre-Shock Parameters).* Uses fixed average mean log-return $\bar{\mu}_{log,fixed}$ and average daily realized volatility $\hat{\sigma}_{daily,fixed}$ calculated over the year `before` the shock ($[t_{shock} - \Delta_{1yr}, t_{shock} - 1]$) to simulate $r_t^{(j)} \sim N(\bar{\mu}_{log,fixed}, \hat{\sigma}^2_{daily,fixed})$ for all post-shock days t. The volatility forecast is the constant $\hat{\sigma}_{daily,fixed}$ (annualized). This is a static approach.

2.6. **AutoGluon-TimeSeries Setup.** We use AutoGluon-TimeSeries [13] version 1.2 to generate DL-based forecasts. Crucially, AutoGluon was trained `only` on data available `before` the shock date (up to Feb 23, 2022). Two separate predictors were trained:

(1) **Volatility Predictor:** Trained to forecast `Realized Volatility Daily` (non-annualized, scaled by 100) over the entire post-shock horizon.
(2) **Price Predictor:** Trained to forecast `Adj Close` over the same horizon.

Both used the `best_quality` preset, RMSE evaluation, a 3600s time limit, GPU acceleration, and available past covariates (e.g., price, volume, returns for volatility predictor). AutoGluon selected a `WeightedEnsemble` as the best model based on validation performance (using validation folds within the pre-shock data). This pre-trained ensemble was then used to generate a **single, static forecast** for the entire post-shock period (median quantile "0.5" for price, "mean" for volatility) without any updates using post-shock data. The mean daily volatility forecast was annualized for evaluation.

2.7. **Adaptive Standard Machine Learning Models.** Beyond Deep Learning, we also evaluate standard ML models in an adaptive forecasting framework for **both volatility and price**. We tested LightGBM, XGBoost, RandomForest, and Ridge regression.

- **Features:** We experimented with different feature sets, including basic lags and rolling window statistics (mean, std) of key variables (price, returns, volatility, volume) over multiple horizons (e.g., 5, 10, 21, 63 days). Rolling window features consistently yielded superior performance compared to basic lag features. Care was taken during feature engineering to use only information available prior to the forecast point, avoiding common sources of look-ahead bias. The validity of the features in relation to the observed model performance is further discussed in Section 3.6.
- **Adaptive Training:** Models were retrained periodically using all data up to $t-1$ to predict the target variable (volatility or price) at time t. We tested retraining steps

(step size) of 1, 5, and 10 days. For the best performing model, XGBoost, a step size of 5 days provided the optimal results for both price and volatility forecasting in the post-shock period.

- **Target:** Separate models were trained. For volatility forecasting, the target variable was the daily realized volatility (non-annualized, scaled by 100). For price forecasting, the target was the `Adj Close` price.
- **Optimization:** Hyperparameters for the best performing model configuration (XGBoost with rolling features) were optimized using Optuna based on time series cross-validation on the pre-shock data.
- **Evaluation:** The resulting daily volatility forecasts were annualized for comparison in Table 8. Price forecasts were directly compared to actual prices. Results for the overall best-performing standard ML configuration identified – **Adaptive XGBoost using rolling features and a 5-day retraining step** – are reported for both price and volatility in the consolidated comparison.

2.8. **Adaptive TimesFM Rolling Forecast.** We utilize the TimesFM foundation model [5] (specifically, `google/timesfm-2.0-500m-pytorch`) for forecasting in an adaptive, rolling framework. Separate experiments were conducted for price and volatility forecasting.

- **Strategy:** A rolling forecast simulation is performed starting from the shock date. At predefined intervals (`step size`), the model uses the most recent `context_len` days of historical data (either price or volatility, depending on the target) to forecast the next `horizon_len` days. Only the first `step_size` days of this forecast are kept and evaluated. The historical window then rolls forward by `step_size` days.
- **Parameters:** We used `context_len = 128` days and `horizon_len = 30` days. We tested `step_size` values of 1, 5, 10, 15, and 20 days. Performance generally degraded with increasing step size; therefore, results for the best performing `step_size=1` are reported.
- **Target:** For price forecasting, the model directly forecasts the price (**Adj Close**). For volatility forecasting, the model forecasts the **daily realized volatility** (non-annualized, scaled by 100).
- **Evaluation:** The median forecast (`timesfm` column from the model output) for the relevant `step_size` days is compared against the actual target (`y`) to calculate MAE and RMSE. Volatility forecasts were annualized for comparison. Results for `step_size=1` are reported in Table 8 for both price and volatility.

2.9. **Simulation and Forecasting Strategies.** We compare the following strategies post-shock, starting from $S_{shock-1}$:

(1) **Fixed GARCH/GJR:** Models estimated once on pre-shock data. Used for static price simulation ($n_{sims} = 20000$) and static volatility forecast over the entire post-shock period.

(2) **Adaptive GARCH/GJR:** Parameters re-estimated daily using an expanding window including post-shock data. Used for dynamic one-step-ahead price simulation ($n_{sims} = 20000$) and dynamic volatility forecast ($\hat{\sigma}_t$).
(3) **Fixed Naive:** Uses fixed pre-shock averages for static price simulation ($n_{sims} = 20000$) and static volatility forecast.
(4) **Dynamic Naive:** Uses rolling history (T-1 year or T-1 day) for dynamic price simulation ($n_{sims} = 20000$) and dynamic volatility forecast (lagged rolling volatility).
(5) **AutoGluon Forecast:** Uses the pre-trained AutoGluon ensembles (Sec 2.6) to generate a single, static forecast path for price and volatility over the entire post-shock horizon.
(6) **Adaptive ML Forecast (XGBoost):** Uses the best performing standard ML configuration identified (Sec 2.7): Adaptive XGBoost with rolling features and a 5-day retraining step, generating forecasts for both price and volatility.
(7) **Adaptive TimesFM Forecast:** Uses the TimesFM model in a rolling forecast simulation (Sec 2.8, step=1) to generate adaptive forecasts for both price and volatility.

Strategies (1), (3), and (5) represent forecasts made *at the time of the shock* using only past information (static). Strategies (2), (4), (6), and (7) represent forecasts that **adapt** using information revealed after the shock (dynamic).

2.10. **Evaluation Metrics.** Performance is evaluated over the post-shock period T.

2.10.1. *Price Path Accuracy Metrics.* Median simulated price ($\hat{S}_{t,median}$ for GARCH/Naive), AutoGluon median price forecast ($\hat{S}_{t,AG,0.5}$), Adaptive XGBoost price forecast ($\hat{S}_{t,XGB}$), or TimesFM median price forecast ($\hat{S}_{t,TFM}$) compared against actual price (S_t). Lower is better.

(1) **RMSE**: $RMSE_{price} = \sqrt{\frac{1}{T}\sum_{t=t_0}^{t_0+T-1}(S_t - \hat{S}_t)^2}$
(2) **MAE**: $MAE_{price} = \frac{1}{T}\sum_{t=t_0}^{t_0+T-1}|S_t - \hat{S}_t|$

where $\hat{S}_t$ is the respective model's price forecast/simulation median.

2.10.2. *Volatility Forecast Accuracy Metrics.* Model's annualized forecast volatility ($\hat{\sigma}_{model,t}$) compared against actual annualized realized volatility ($\sigma_{RV,t}$) over T_{vol} days (from $t_0 + W$ to $t_0 + T - 1$). AutoGluon's daily mean forecast $\hat{\sigma}_{AG,daily,mean}$ is annualized: $\hat{\sigma}_{AG,t} = \sqrt{N} \times \hat{\sigma}_{AG,daily,mean}$. GARCH/Naive daily σ_t are similarly annualized. Adaptive ML (XGBoost) and TimesFM daily forecasts are also annualized. Lower is better.

(1) **RMSE**: $RMSE_{vol} = \sqrt{\frac{1}{T_{vol}}\sum(\sigma_{RV,t} - \hat{\sigma}_{model,t})^2}$
(2) **MAE**: $MAE_{vol} = \frac{1}{T_{vol}}\sum|\sigma_{RV,t} - \hat{\sigma}_{model,t}|$
(3) **QLIKE** [12]: $QLIKE_{vol} = \frac{1}{T_{vol}}\sum\left(\frac{\sigma^2_{RV,t}}{\hat{\sigma}^2_{model,t}} - \ln\left(\frac{\sigma^2_{RV,t}}{\hat{\sigma}^2_{model,t}}\right) - 1\right)$

3. Results and Comparisons

3.1. **Empirical Analysis of Black-Scholes Assumptions.** Before proceeding with volatility forecasting, we begin by empirically examining the key mathematical assumptions of the

Black-Scholes model, namely constant volatility and normality of logarithmic returns. This is crucial to demonstrate the context of our study and the need to go beyond the B-S framework, especially in a market environment highly susceptible to shocks.

3.1.1. *Volatility Dynamics Pre- and Post-Shock.* Descriptive statistics in Table 1 and Figure 2 clearly demonstrate a significant change in volatility dynamics after the shock. The mean annualized realized volatility increased significantly, rising from 31.37% pre-shock to 40.11% post-shock. The standard deviation also rose substantially, indicating wider volatility swings.

TABLE 1. Descriptive Statistics of Annualized Realized Volatility Before and After Shock (2022-02-24)

Statistic	Before Shock	After Shock
Mean (%)	31.37	40.11
Std Dev (%)	26.15	35.42
Median (%)	16.60	16.63
Min (%)	14.53	15.53
Max (%)	114.81	108.48

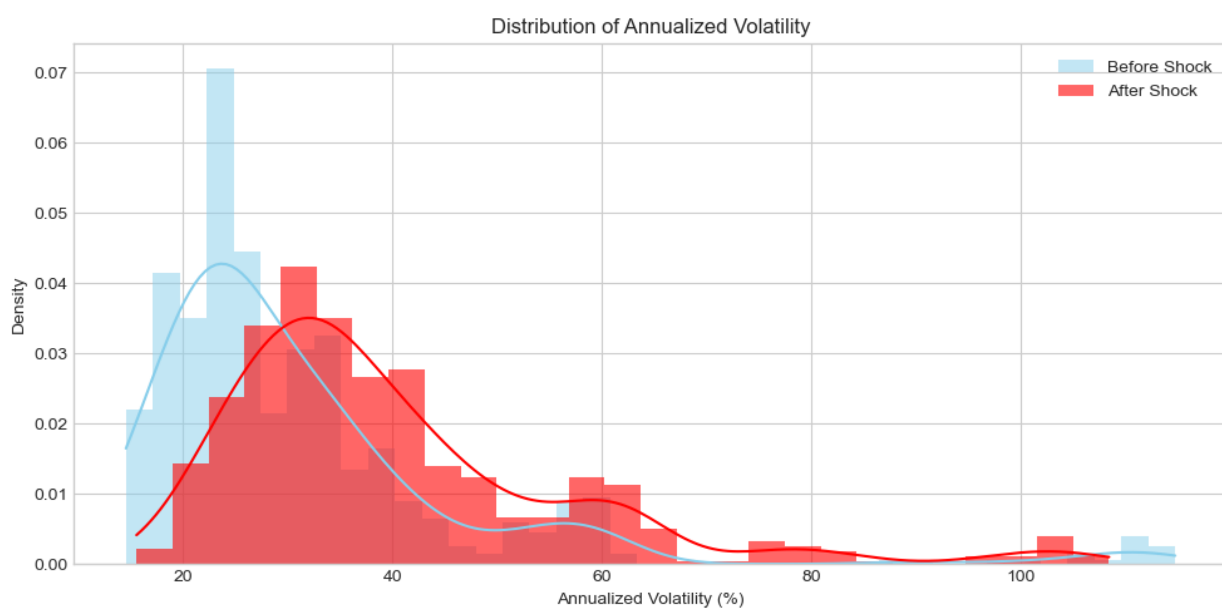

FIGURE 2. Distribution of Annualized Realized Volatility Before and After Shock (2022-02-24)

Figure 2 visually confirms this shift, showing the post-shock volatility distribution shifted towards higher levels and exhibiting greater dispersion compared to the pre-shock period. This evidence points to a clear structural break in volatility behavior coinciding with the shock, directly challenging the Black-Scholes assumption of constant volatility.

3.1.2. *Normality of Log-Returns.* Analysis shows logarithmic returns fail to meet the Black-Scholes normality assumption. Statistics in Table 2 indicate positive skewness and high kurtosis (leptokurtosis) in both periods (pre and post-shock), demonstrating a right-skewed, heavy-tailed distribution.

TABLE 2. Descriptive Statistics of Daily Log-Returns Before and After Shock (2022-02-24)

Statistic	Before Shock	After Shock
Mean	0.000338	0.003321
Std Dev	0.022061	0.027382
Skewness	0.375239	0.738859
Kurtosis	9.976945	9.294488

Histograms (Figure 3) and Q-Q plots (Figure 4) confirm significant deviations from normality, especially in the tails. The Shapiro-Wilk test (Table 3) firmly rejects normality (p $\approx$ 0.0000) for both periods. Increased volatility after the shock and persistent non-normality weaken the Black-Scholes model's applicability, particularly during shocks when extreme price movements are more likely.

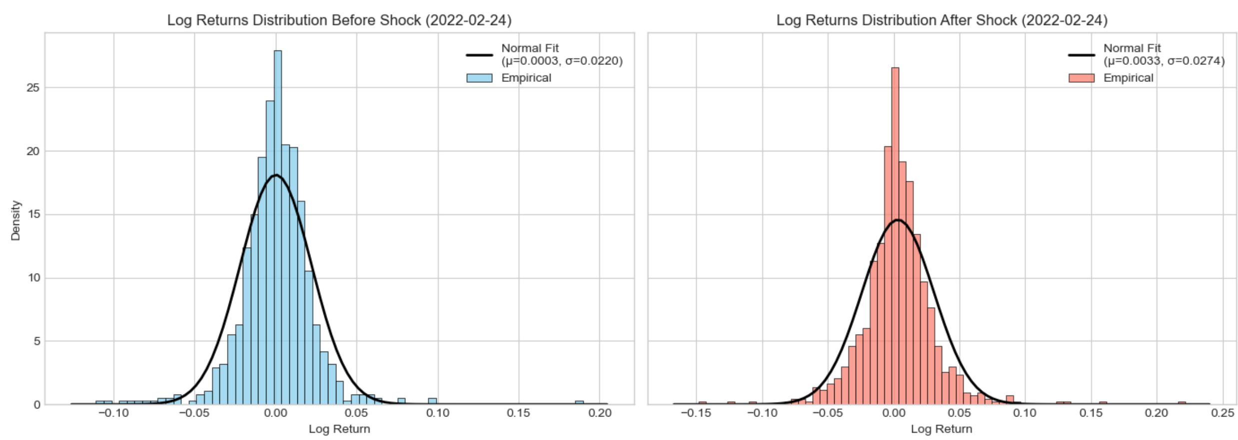

FIGURE 3. Histograms of Daily Log Returns Before and After Shock (2022-02-24) with Fitted Normal Distributions

TABLE 3. Shapiro- Wilk Normality Test for Daily Log Returns

Test Period	Statistic	p-value
Before Shock	0.9062	< 0.0001
After Shock	0.9055	< 0.0001

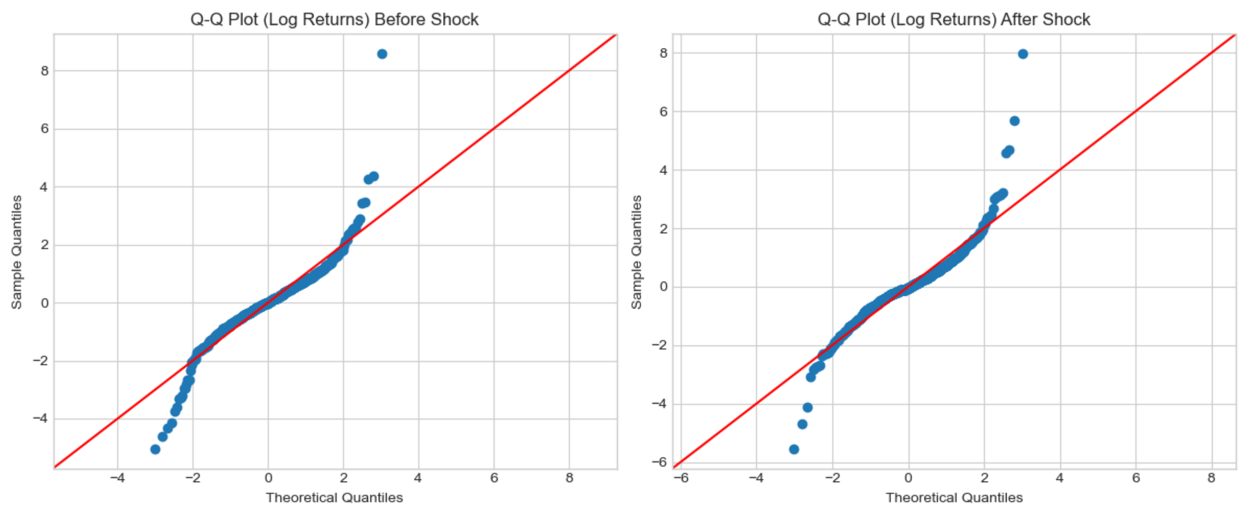

FIGURE 4. Q-Q Plots of Daily Log Returns Against Normal Distribution Before and After Shock (2022-02-24)

3.1.3. *Implications for Modeling.* Empirical evidence confirms Black-Scholes assumptions fail for the analyzed asset, especially during the market shock. Volatility shows a structural shift post-shock, while log-returns display non-normality and heavy tails throughout. These findings highlight the standard model's limitations for pricing and risk management during market stress, necessitating more sophisticated approaches like GARCH, standard ML, and potentially Deep Learning (like AutoGluon or TimesFM) to capture time-varying volatility and non-normality.

3.2. **Post-Shock Simulation with Fixed Pre-Shock Parameters.** We first examine simulations based on models and parameters derived solely from pre-shock data to represent a counterfactual "no-shock adaptation" scenario. This includes GARCH(1,1) and GJR-GARCH(1,1,1) models estimated on data up to Feb 23, 2022 (BIC selection in Tables 4, 5), and the Fixed Naive approach using average parameters from the year before the shock. These models generate static forecasts for the entire post-shock period.

Visual inspection of the simulation paths (not shown here) and quantitative metrics (Table 6) confirm the failure of these static approaches. Actual prices diverge sharply upwards from the simulated paths, and actual volatility remains significantly elevated compared to the low, decaying, or constant volatility forecasts generated by these models using only pre-shock parameters. This demonstrates the inadequacy of relying solely on pre-shock information after a major structural break has occurred.

3.3. **Post-Shock Simulation/Forecast with Adaptive/Dynamic Parameters.** Next, we examine models that adapt using post-shock data. This includes Adaptive GARCH/GJR (re-estimated daily), Dynamic Naive (using rolling T-1 year or T-1 day data), and the best

TABLE 4. GARCH(p,q) Model Comparison (Pre-Shock Data, Sorted by BIC)

p	q	AIC	BIC	LogLikelihood	$\alpha + \beta$
1	1	3252.34	3275.75	−1621.17	0.9676
1	2	3254.26	3282.35	−1621.13	0.9509
2	1	3254.28	3282.36	−1621.14	0.9662
2	2	3255.79	3288.55	−1620.89	0.9450
1	3	3256.26	3289.03	−1621.13	0.9509
3	1	3256.28	3289.04	−1621.14	0.9662
3	2	3257.26	3294.71	−1620.63	0.9415
2	3	3257.64	3295.09	−1620.82	0.9425
3	3	3258.71	3300.84	−1620.35	0.9186

Models fitted using pre-shock data ($N = 797$) for RHM.DE returns, constant mean, Student's t-distribution.

TABLE 5. GJR-GARCH(p,1,q) Model Comparison (Pre-Shock Data, Sorted by BIC)

p	o	q	AIC	BIC	γ
1	1	1	3244.49	3272.57	0.0793
1	1	2	3246.20	3278.97	0.1213
2	1	1	3246.25	3279.02	0.0801
2	1	2	3247.72	3285.17	0.1253
3	1	1	3248.11	3285.56	0.0829
1	1	3	3248.20	3285.65	0.1213
3	1	2	3249.67	3291.79	0.1238
2	1	3	3249.72	3291.85	0.1252
3	1	3	3251.67	3298.47	0.1238

Models fitted using pre-shock data ($N = 797$) for RHM.DE returns, constant mean, GJR-GARCH($o = 1$), Student's t-distribution.

Table 6. Evaluation Metrics for Fixed Pre-Shock Simulations/Forecasts

Model (Fixed Pre-Shock Params)	Price Path		Volatility		
	RMSE	MAE	RMSE	MAE	QLIKE
GARCH(1,1) Fixed	320.8753	235.1205	20.3088	13.3936	0.7943
GJR-GARCH(1,1,1) Fixed	336.6622	248.7592	21.4038	14.4542	0.9942
Fixed Naive (Avg Pre-Shock)	343.7507	255.4850	24.3017	17.9503	1.7356

Lower values indicate better performance. Price metrics compare median simulated price to actual price. Volatility metrics compare model's implied/used volatility to actual realized volatility (proxy). QLIKE is for variance/volatility forecast quality. Volatility metrics are for annualized volatility (%).

Adaptive standard ML model identified (Adaptive XGBoost, Sec 2.7). We analyze the simulation paths (where applicable) and forecasts generated by these adaptive approaches against the actual data.

Analysis of the underlying data (plots not shown) reveals improved volatility tracking by adaptive GARCH/GJR compared to fixed models. Their conditional volatility follows the actual realized volatility more closely after incorporating post-shock information. The Dynamic Naive volatility also adapts but might appear lagged compared to the GARCH models. The adaptive XGBoost model, benefiting from rolling features and periodic retraining, demonstrates exceptionally strong tracking capabilities for both price and volatility according to the metrics. Price path simulations/forecasts generated by adaptive GARCH/Naive/XGBoost models tend to be closer to reality than those from the fixed models, with XGBoost showing remarkably low errors. Quantitative performance metrics are provided in Table 7.

Table 7. Evaluation Metrics for Adaptive/Dynamic Simulations/Forecasts (Excluding TimesFM and Adaptive ML)

Model (Adaptive/Dynamic Params)	Price Path		Volatility		
	RMSE	MAE	RMSE	MAE	QLIKE
GARCH(1,1) Adaptive	306.4631	220.1101	7.1723	5.0485	0.0518
GJR-GARCH(1,1,1) Adaptive	301.9481	197.8874	8.0955	5.3340	0.0611
Dynamic Naive (T-1 Year)	221.7467	160.5942	27.7763	21.9810	1.5855
Dynamic Naive (T-1 Day)	14.8315	7.3015	3.6707	1.5967	0.0167

Lower values indicate better performance. Price metrics compare median simulated price to actual price. Volatility metrics compare model's conditional/used volatility to actual realized volatility (proxy). QLIKE is for variance/volatility forecast quality. Volatility metrics are for annualized volatility (%). This table excludes TimesFM and Adaptive XGBoost results shown in Table 8.

3.4. **Deep Learning Approach using AutoGluon (Static Forecast).** We employed AutoGluon-TimeSeries (Sec 2.6), training models solely on pre-shock data to generate static forecasts for the entire post-shock period. The best validation model (WeightedEnsemble) was used for prediction.

3.4.1. *AutoGluon Volatility Forecasting.* The AutoGluon mean volatility forecast, evaluated against actual realized volatility, yielded the metrics presented later in Table 8. Textually, the forecast captured the generally elevated post-shock level but was smoother and less reactive than adaptive GARCH and adaptive ML models. Its performance was better than the naive models based on QLIKE but worse than adaptive GARCH and significantly worse than the optimized adaptive XGBoost across all metrics (RMSE, MAE, QLIKE). Notably, the AutoGluon leaderboard indicated **ChronosFineTuned** had a better test RMSE (14.42) than the chosen **WeightedEnsemble** (test RMSE 17.33 based on calculation, 58.03 on leaderboard), suggesting pre-shock validation was not optimal for selecting the best model for the post-shock regime.

3.4.2. *AutoGluon Price Forecasting.* The AutoGluon median price forecast significantly underestimated the post-shock trend. Performance metrics (Table 8) show higher errors than adaptive GARCH and substantially higher than Dynamic Naive (T-1 day), TimesFM, and especially the adaptive XGBoost model. Similar to volatility, the test leaderboard suggested **ChronosFineTuned** (test RMSE 213.61) performed better on the post-shock data than the validation-selected **WeightedEnsemble** (test RMSE 328.92 calculated, 335.45 on leaderboard), again highlighting the challenge of forecasting through the regime shift using only pre-shock data for model selection and training.

3.5. **Deep Learning Approach using TimesFM (Adaptive Forecast).** We employed the TimesFM foundation model in an adaptive rolling forecast setting (Sec 2.8). Unlike the static AutoGluon approach, TimesFM used updated historical context windows to generate forecasts periodically throughout the post-shock period. We evaluated its performance in forecasting both the price path and volatility. As noted in the methodology, performance degraded as the retraining step size increased, with daily updates (step size=1 day) providing the best results for TimesFM. The quantitative results for this best configuration are presented in Table 8. This adaptive approach allowed the model to incorporate post-shock information, leading to significantly better performance compared to the static AutoGluon forecast, although it was ultimately outperformed by the adaptive XGBoost model in this case study.

3.6. **Comparative Analysis of Simulation/Forecast Performance.** Table 8 consolidates the evaluation metrics for all strategies, grouping them by whether they used adaptation post-shock.

Key observations from Table 8:

Table 8. Consolidated Evaluation Metrics Post-Shock

Model / Strategy	Price Path Accuracy		Volatility Forecast Accuracy		
	$RMSE_{price}$	MAE_{price}	$RMSE_{vol}$ (%)	MAE_{vol} (%)	$QLIKE_{vol}$
Static Forecasts (Trained/Calibrated Pre-Shock Only)					
GARCH(1,1) Fixed	320.88	235.12	20.31	13.39	0.794
GJR-GARCH(1,1,1) Fixed	336.66	248.76	21.40	14.45	0.994
Fixed Naive (Avg Pre-Shock)	343.75	255.49	24.30	17.95	1.736
AutoGluon (WeightedEnsemble)	328.92	262.49	17.33	11.26	0.084
Adaptive / Dynamic Forecasts (Using Post-Shock Data)					
GARCH(1,1) Adaptive	306.46	220.11	7.17	5.05	0.052
GJR-GARCH(1,1,1) Adaptive	301.95	197.89	8.10	5.33	0.061
Dynamic Naive (T-1 Year)	221.75	160.59	27.78	21.98	1.586
Dynamic Naive (T-1 Day)	14.83	7.30	3.67	1.60	0.017
Adaptive XGBoost (Rolling Feats, step=5d)	**2.20**	**1.49**	**1.09**	**0.80**	**0.002**
TimesFM (Rolling, step=1d)	14.90	7.77	4.10	2.05	0.004

Lower values better. Price/Volatility: Forecast vs Actual (best in **bold**). AutoGluon: static pre-shock `WeightedEnsemble`. Adaptive XGBoost: adaptive forecast using rolling features and 5-day retraining step. TimesFM: adaptive rolling forecast (1-day step) for both price and volatility. Volatility metrics are for annualized volatility (%).

(1) **Failure of Static Forecasts:** All models relying solely on pre-shock information (Fixed GARCH/GJR, Fixed Naive, AutoGluon) performed poorly compared to adaptive alternatives, especially in price path simulation and volatility forecasting accuracy. This underscores the danger of using models calibrated on pre-shock regimes after a structural break.

(2) **Benefit of Adaptability:** Introducing adaptability via daily updates (Adaptive GARCH/GJR), rolling history (Dynamic Naive), periodic retraining (Adaptive XGBoost), or rolling context windows (Adaptive TimesFM) yielded significant improvements over static models.

(3) **Adaptive XGBoost Dominance:** The **Adaptive XGBoost model using rolling features and a 5-day retraining step** emerged as the clear best performer in this specific case study.

- **Volatility Forecasting:** XGBoost achieved the lowest RMSE (1.09%), MAE (0.80%), and QLIKE (0.002), significantly outperforming all other models, including Dynamic Naive (T-1d), TimesFM, and Adaptive GARCH. Its QLIKE score is particularly indicative of superior density forecasting.
- **Price Path Simulation/Forecast:** XGBoost demonstrated remarkably high accuracy, achieving by far the lowest RMSE (2.20) and MAE (1.49). This level of accuracy significantly surpasses the next best models (Dynamic Naive T-1d and TimesFM, both with RMSE around 15).

  Furthermore, an analysis of the input data columns and the feature engineering *process confirmed that readily available, highly correlated information (like the unadjusted Close price) or other intra-day prices (High, Low, Open) were not directly used in the feature generation functions (add lag features, add rolling features) in a way that would constitute obvious data leakage.*

(4) **Deep Learning Performance (Static vs Adaptive):** The static AutoGluon baseline, trained only pre-shock, did not match the performance of adaptive models post-shock. In contrast, the adaptive TimesFM rolling forecast delivered strong performance, particularly for volatility QLIKE (0.004) and competitive price path accuracy (RMSE 14.90), although it was surpassed by the adaptive XGBoost model on all metrics in this comparison. This reinforces that adaptive strategies are likely necessary for DL models to be effective through structural breaks.

(5) **Relative Performance of Other Adaptive Models:** While overshadowed by XGBoost, the Dynamic Naive (T-1 day) and Adaptive TimesFM (step=1) models still performed well, significantly outperforming adaptive GARCH and all static models. They represent strong adaptive benchmarks. Adaptive GARCH models showed improvement over their fixed counterparts but lagged behind the ML, TimesFM, and T-1d Naive approaches.

(6) **Feature Engineering and Adaptation Frequency:** The success of XGBoost highlights the potential benefits of combining a powerful gradient boosting algorithm with informative rolling features and an appropriate adaptation frequency (5-day steps in this case, which performed slightly better than 1-day steps for XGBoost).

3.7. **Discussion on Dynamic Naive (T-1 Day) Performance.** Prior to incorporating the optimized XGBoost results, the Dynamic Naive model using the T-1 day lag showed remarkable competitiveness, particularly for price path forecasting and volatility RMSE/MAE. However, the results from the Adaptive XGBoost model (Table 8) place the Naive model's performance in a different perspective for this specific case study.

While the T-1 Naive model significantly outperformed static models and adaptive GARCH, it was clearly surpassed by the Adaptive XGBoost approach across all reported metrics for both price and volatility. For instance, XGBoost's volatility RMSE (1.09

This comparison highlights several points:

- **Value as Benchmark:** The T-1 Naive model remains an essential and challenging benchmark. Its strong performance relative to GARCH and static models underscores the high degree of short-term autocorrelation present in the post-shock period.
- **Limits of Simplicity:** In this instance, the extreme simplicity of the Naive model, while allowing rapid adaptation, was ultimately less effective than a more complex model (XGBoost) capable of leveraging engineered features (rolling statistics) to capture more intricate dynamics.
- **Power of Feature Engineering:** The substantial performance gap between Naive T-1d and Adaptive XGBoost strongly suggests that the rolling features provided significant predictive power that the simple lagged value alone could not capture.

Therefore, while the T-1 Naive model's effectiveness should not be dismissed, especially given its simplicity, this analysis demonstrates that well-configured adaptive ML models with relevant features can achieve substantially higher forecasting accuracy in challenging post-shock environments.

4. Conclusions

5. Code and Implementation

The analysis was implemented in Python (versions 3.12), using `pandas`, `numpy`, `arch`, `yfinance`, `autogluon.timeseries` (v1.2), `timesfm` (v2.0), `statsmodels`, `xgboost`, `lightgbm`, `scikit-learn` (for RandomForest, Ridge), `optuna`, and `matplotlib/seaborn`. The code, including GARCH modeling, naive approaches, AutoGluon setup, adaptive ML implementations (with XGBoost identified as the top performer), TimesFM rolling forecast simulation, feature engineering experiments (particularly rolling features), and Optuna hyperparameter optimization, potentially covering analyses beyond the scope of this paper, is available at:

https://github.com/maksdebowski/mathematical_foundations_of_ML_DL.git

in the **project** directory. Details might be found in the codebase and associated README files.

Exclusive footage of one of the authors deep into a late-night coding session.

References

1. Jean-Noël Barrot and Julien Sauvagnat, *Input specificity and the propagation of idiosyncratic shocks in production networks*, The Quarterly Journal of Economics **131** (2016), no. 3, 1543–1592.
2. Fischer Black and Myron Scholes, *The pricing of options and corporate liabilities*, The Journal of Political Economy **81** (1973), no. 3, 637–654.
3. Tim Bollerslev, *Generalized autoregressive conditional heteroskedasticity*, Journal of Econometrics **31** (1986), no. 3, 307–327.
4. Rama Cont, *Empirical properties of asset returns: stylized facts and statistical issues*, Quantitative Finance **1** (2001), no. 2, 223–236.
5. Anshul Das, Shounak Garg, Luca Stella, Vijay Chalumuri, Daniel C Maddix, Andrew M Dai, Syama Sundar Rangapuram, Dmitry Kuzin, and David Dohan, *A decoder-only foundation model for time-series forecasting*, 2023.
6. Robert F. Engle, *Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation*, Econometrica **50** (1982), no. 4, 987–1007.
7. Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola, *AutoGluon-Tabular: Robust and accurate automl for structured data*, Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 11700–11710.
8. Lawrence R. Glosten, 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 **48** (1993), no. 5, 1779–1801.
9. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT Press, 2016.
10. John C. Hull, *Options, futures, and other derivatives*, 10th ed., Pearson, 2018.
11. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, *Deep learning*, Nature **521** (2015), no. 7553, 436–444.
12. Andrew J. Patton, *Volatility forecast comparison using imperfect volatility proxies*, Journal of Econometrics **161** (2011), no. 2, 244–256.
13. Oleksandr Shchur, Caner S. Aussignac, Jan Gasthaus, and Syama Sundar Rangapuram, *AutoGluon-TimeSeries: Automl for probabilistic time series forecasting*, 2023.