Zero : Jurnal Sains, Matematika, dan Terapan
E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 9, Number 1, 2025
DOI: 10.30829/zero.v9i1.24614
Page: 201-214

Using an LSTM Neural Network to Improve Symmetric and Asymmetric GARCH Volatility Forecast

[1] **Setya Budi Rahmawanto**
Master's Program in Data Science, Universitas Kristen Satya Wacana, Salatiga, Indonesia

[2] **Didit Budi Nugroho**
Master's Program in Data Science, Universitas Kristen Satya Wacana, Salatiga, Indonesia
Study Center for Multidisciplinary Applied Research and Technology (SeMARTy), Universitas Kristen Satya Wacana, Salatiga, Indonesia

[3] **Suryasatriya Trihandaru**
Master's Program in Data Science, Universitas Kristen Satya Wacana, Salatiga, Indonesia

Article Info

Article history:

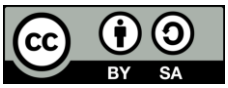
Accepted, 28 May 2025

Keywords:

Financial Time Series;
GARCH;
Hybrid Mode;
LSTM;
Volatility Forecasting

ABSTRACT

Volatility forecasting is crucial for financial risk management, yet traditional models like GARCH struggle with nonlinearities and asymmetric effects. This study leverages Long Short-Term Memory (LSTM) neural networks to enhance symmetric and asymmetric GARCH models, addressing these limitations. By integrating LSTM with GARCH, GARCH-X, and Realized GARCH frameworks, we propose hybrid models (Baseline and Extended versions) to improve forecasting accuracy. Using daily data from FTSE 100, Nikkei 225, and S&P 500 indices (2000–2020), we compared hybrid models against traditional models. Results show that the Extended LSTM hybrid model outperforms both traditional GARCH-type models and the Baseline LSTM, capturing complex volatility patterns more effectively. The Extended model's architecture, featuring ReLU, GRU, and dropout layers, mitigates over-smoothing and enhances responsiveness to market fluctuations. This research demonstrates LSTM's potential to refine volatility forecasting, offering valuable insights for investors and risk managers.



Corresponding Author:

Didit Budi Nugroho,
Master's Program in Data Science,
Universitas Kristen Satya Wacana, Salatiga, Indonesia,
Email: didit.budinugroho@uksw.edu

1. INTRODUCTION

The Generalised Autoregressive Conditional Heteroskedasticity (GARCH) model, see [1], has been recognized as one of the most popular volatility models. However, this model fails to account for asymmetric effects, where positive and negative returns exhibit different volatility behaviors. To address this limitation, asymmetric GARCH models such as the Exponential GARCH (EG) in [2] and GJR (Glosten-Jagannathan Runkle) in [3], [4] were developed, offering better alignment with the complexities of financial markets.

Despite these advancements, traditional GARCH models still face challenges in capturing nonlinearities and dynamic market patterns, particularly when high-frequency data is involved. For instance, while models like

GARCH with exogenous variables (GX) in [5], [6] and Realized GARCH (RG) in [7] integrated realized measures, which are calculated from intra-day data, to improve forecasting accuracy, they remain constrained by their linear frameworks. This limitation becomes evident in highly volatile or non-stationary market conditions, where traditional models often underperform.

The rise of deep learning, particularly Recurrent Neural Networks (RNNs) and their variants like Long Short-Term Memory (LSTM), has offered a new approach to time series forecasting. For example, in investigating the daily stock price prediction of the top five companies in Thailand 50 (SET50) index, study [8], [9] showed that LSTM is suitable for predict future stock market values. Furthermore, LSTM offers several advantages over traditional GARCH models, including the ability to capture non-linearity in time series, not requiring stationarity for modeling, and demonstrating superior performance in volatility forecasting compared to GARCH-type models. Study [10], [11] showed that volatility predictions using LSTM can outperform GARCH models. Meanwhile, study [12] introduced a Baseline LSTM model and an Extended LSTM model to predict stock price volatility. However, these studies have three notable gaps in previous studies that limit their practical applicability and generalizability: a limited empirical scope that often restricts analysis to short time frames or a single asset, inadequate feature engineering, and a lack of comprehensive comparative analysis against a wide range of traditional GARCH variants.

This study systematically addresses these limitations through three key innovations. First, we significantly expand the empirical scope by analyzing three major stock indices, including Financial Times Stock Exchange 100 (FTSE100—United Kingdom), the Nikkei Stock Average (N225—Japan), and the Standard and Poor's 500 (SP500—United States), representing economies ranked among the top 10 by nominal GDP as of 2024, thus providing a comprehensive representation of the global economy. Our dataset spans two decades (2000–2020), capturing multiple economic cycles including the dot-com bubble, global financial crisis, and periods of economic expansion, ensuring robust conclusions across diverse market conditions. Second, we implement sophisticated feature engineering by incorporating high-frequency realized volatility measures and developing novel hybrid architectures that combine LSTM with multiple GARCH variants (GX, EG, GJR, and RG). Third, we establish a rigorous comparative framework using three error metrics and four volatility proxies to provide unambiguous performance evaluations. These advancements not only deliver more accurate volatility forecasts but also create a standardized methodology for future research in financial machine learning, particularly in exploring advanced neural architectures and their application to emerging asset classes.

Building upon this, this study aims to answer key questions regarding the development of hybrid models. Specifically, we investigate how a hybrid model, particularly an Extended LSTM architecture, can be developed to improve volatility forecasting accuracy compared to traditional GARCH variants. We also seek to determine the extent to which the proposed hybrid model's performance surpasses that of traditional models in capturing complex volatility patterns and asymmetric effects across different market conditions and stock indices. Furthermore, we explore how the implementation of advanced feature engineering and a hierarchical architecture in the LSTM model can effectively mitigate over-smoothing and enhance responsiveness to market fluctuations.

The remainder of this paper is structured as follows. The Research Methods section details the methodology used in this analysis, including both traditional GARCH and LSTM hybrid models. The Result and Analysis section presents the performance comparison of the models, while the Conclusion section summarizes the key findings and provides recommendations for future study.

2. RESEARCH METHOD

2.1. Traditional models

Consider a time series $\{R_t\}_{t\geq 1}$ that represents a vector of returns on individual asset and is typically calculated as the natural logarithm of the price relative. This means $R_t = \log\left(\frac{P_t}{P_{t-1}}\right)$, where P_t is the asset price at time t and P_{t-1} is the price at the previous time step. This is then expressed as a percentage change by multiplying the result by 100. Assume that the return series R_t has a normal distribution with mean of zero and variance of σ_t^2 as follows:

$$R_t = \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_t^2), \tag{1}$$

where $\mathcal{N}$ denotes a Normal distribution. In the original GARCH model, the conditional variance σ_t^2 is indeed determined by three parameter: ω, α, β. These parameters govern the dynamics of volatility in the GARCH(1,1) process:

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

In a GARCH(1,1) model, the first 1 refers to the most recent lagged value of the squared return, and the second 1, refers to the most recent lagged value of the conditional variance. A more flexible dependence of return and volatility (square root of variance) shocks is offered by the EG(1,1) and GJR(1,1) models.

The increasing availability of high-frequency transaction data for financial assets has enabled researchers to develop a more accurate volatility estimator known as realized volatility. It results in a near-real-time estimate of "true" volatility, providing a more immediate and dynamic picture of market fluctuations [13], [14]. These improvements have allowed researchers to incorporate realized measures into volatility models directly. For example, GX-type models, such as the GX in [5], [6], GJRX in [4] and EGX in [2], [15]. Studies [2], [16] extended the EGX model to Log-linear Realized GARCH (LRG) specification. Study [17] confirmed the predictive advantage of LRG over the EG model. Meanwhile, studies [18], [19] developed the Realized Exponential GARCH (REG) model by incorporating multiple measurement equations. Recently, study [20] proposed the G@C (GARCH with Conditional AutoRegressive Realized volatility structure) model using a multiplicative error model structure to the measurement equation. The G@C model stands out for its simpler specification than the other variants and provide better forecasts. Table 1 summarizes the traditional GARCH-family models examined in this study, while Figure 1 depicts the evolutionary progression of these models—from symmetric variance specifications to asymmetric and realized-measure variants.

Table 1. Overview of the GARCH-family Models

Model	Short Description & Formula	Parameters to be Estimated & Conditions
EG	Take the natural logarithm of squared volatility and capture the asymmetric volatility effect α_2. $$\log(\sigma_t^2) = \omega + \alpha_1\left\lvert\frac{R_{t-1}}{\sigma_t}\right\rvert + \alpha_2\frac{R_{t-1}}{\sigma_t} + \beta\log(\sigma_{t-1}^2)$$	No restrictions on parameters $\omega, \alpha_1, \alpha_2, \beta$
GJR	Capture the asymmetric volatility effect α_2. $$\sigma_t^2 = \omega + \left(\alpha_1 + \alpha_2 1_{[R_{t-1}<0]}\right)R_{t-1}^2 + \beta\sigma_{t-1}^2$$	$\omega, \alpha_1, \alpha_2, \beta > 0$, $0 < \alpha_1 + \alpha_2 < 1$, $0 < \alpha_1 + \alpha_2 + \frac{1}{2}\beta < 1$; $\alpha_2 = 0$: GARCH
GX	Not considers the asymmetric effect, but include the past exogenous variable. $$\sigma_t^2 = \omega + \alpha R_{t-1}^2 + \beta\sigma_{t-1}^2 + \gamma X_{t-1}$$	$\omega, \alpha, \beta, \gamma > 0$, $\alpha + \beta < 1$; $\gamma = 0$: GARCH
EGX	Not only considers the asymmetric effect, but also include the past exogenous variable. $$\log(\sigma_t^2) = \omega + \alpha_1\left\lvert\frac{R_{t-1}}{\sigma_t}\right\rvert + \alpha_2\frac{R_{t-1}}{\sigma_t} + \beta\log(\sigma_{t-1}^2) + \gamma X_{t-1}$$	No restrictions on parameters $\omega, \alpha_1, \alpha_2, \beta, \gamma$; $\gamma = 0$: EG
GJRX	Not only considers the asymmetric effect, but also include the past exogenous variable. $$\sigma_t^2 = \omega + \left(\alpha_1 + \alpha_2 1_{[R_{t-1}<0]}\right)R_{t-1}^2 + \beta\sigma_{t-1}^2 + \gamma X_{t-1}$$	$\omega, \alpha_1, \alpha_2, \beta, \gamma > 0$, $0 < \alpha_1 + \alpha_2 < 1$, $0 < \alpha_1 + \alpha_2 + \frac{1}{2}\beta < 1$; $\gamma = 0$: GJR
LRG	Add a measurement equation that relates the observed realized measure to latent volatility. $$\begin{cases} \log(\sigma_t^2) = \omega + \beta\log(\sigma_{t-1}^2) + \delta(\varepsilon_{t-1}) + \gamma\log(X_{t-1}), \\ \log(X_t) = \xi + \phi\log(\sigma_t^2) + \tau(\varepsilon_t) + u_t, \quad u_t \sim N(0, \sigma_u^2), \\ \text{where } p(\varepsilon_t) = p_1\frac{R_t}{\sigma_t} + p_2\left(\frac{R_t^2}{\sigma_t^2} - 1\right) \end{cases}$$	$\sigma_u^2 > 0$, no restrictions on parameters $\omega, \beta, \gamma, \xi, \phi, \delta_1, \delta_2, \tau_1, \tau_2$; $\xi, \phi, \tau_1, \tau_2, \sigma_u^2 = 0$: EGX
REG	Utilize multiple realized measures. $$\begin{cases} \log(\sigma_t^2) = \omega + \beta\log(\sigma_{t-1}^2) + \delta(\varepsilon_{t-1}) + \gamma u_{t-1}, \\ \log(X_t) = \xi + \phi\log(\sigma_t^2) + \tau(\varepsilon_t) + u_t, \quad u_t \sim N(0, \sigma_u^2) \end{cases}$$	$\sigma_u^2 > 0$, no restrictions on parameters $\omega, \beta, \gamma, \xi, \phi, \delta_1, \delta_2, \tau_1, \tau_2$; $\gamma, \xi, \phi, \tau_1, \tau_2, \sigma_u^2 = 0$: EG
G@C	Specify a multiplicative error model to the measurement equation. $$\begin{cases} R_t = \rho z_t \varepsilon_t, \\ \log(z_t) = \omega + \beta\log(z_{t-1}) + \delta(\varepsilon_{t-1}) + \gamma\log(q_{t-1}), \\ q_t = z_t\epsilon_t, \epsilon_t \sim LN\left(-\frac{\sigma_q^4}{2}, \sigma_q^2\right), q_t = \sqrt{X_t} \end{cases}$$	$\sigma_q^2 > 0$, no restrictions on parameters $\rho, \omega, \beta, \gamma, \xi, \phi, \delta_1, \delta_2$ G@C model is a restricted REG.

2.2. MCMC

Although Maximum Likelihood Estimation (MLE) is the most commonly used method for estimating the parameters of GARCH-type models, the Markov Chain Monte Carlo (MCMC) method has emerged as a strong alternative due to its ability to sample randomly from unknown distributions [21]. The MCMC method is based on Bayes' rule, where the model parameters (e.g. θ) are estimated from the posterior distribution:

$$\text{posterior}(\theta|\text{data}) \propto \text{likelihood}(\text{data}|\theta) \times \text{prior}(\theta). \quad (3)$$

One of the MCMC methods that can be applied is the Adaptive Random Walk Metropolis (ARWM), as described in detail in [2], [4], [5]. This method has been statistically proven to be efficient for estimating GARCHX, RG, GJRX models.

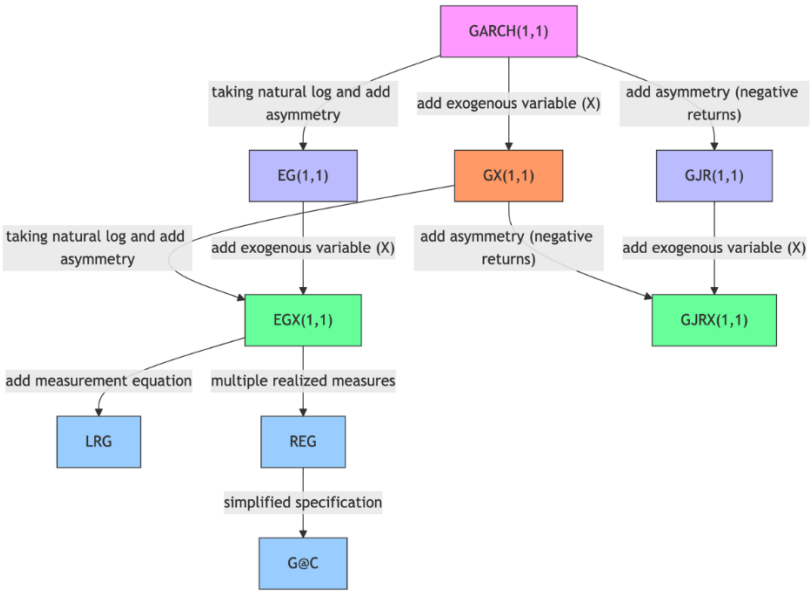


Figure 1. Development Flow of GARCH Volatility Models

2.3. Motivation for LSTM Integration

While GARCH models provide theoretically grounded volatility estimates, their linear parametric forms struggle with three key challenges: (1) nonlinear market dynamics during extreme events, (2) complex interactions between volatility drivers, and (3) structural breaks in financial time series. Traditional approaches often address these issues through ad-hoc modifications—manual adjustments to model specifications (e.g., adding asymmetry terms like α_2 in EGARCH or threshold effects in GJR) that are case-specific and lack generalizability. LSTM networks address these limitations through:

- Nonlinear function approximation: Ability to model complex patterns without predefined equations (unlike GARCH's fixed σ_t^2 formulations).
- Memory cells: Selective retention of long-term dependencies through forget/input gates.
- Adaptive feature extraction: Automatic feature extraction from raw inputs, eliminating manual specification of asymmetry terms.

2.4. LSTM Architecture

LSTM is a variant of RNN designed to capture temporal dependencies in time series data [15], [22]. The basic LSTM architecture, as illustrated in Figure 2, consists of memory cells that include three main components: the Forget Gate (f_t), the Input Gate (i_t), and the Output Gate (o_t). LSTM manages information through the Cell State (C_t), which stores long-term information, and the Hidden State (h_t).

For each time step t, the LSTM computes as follows:

$$\left.\begin{aligned} f_t &= s\left(W_f \cdot [h_{t-1}, y_t] + b_f\right) \\ i_t &= s(W_i \cdot [h_{t-1}, y_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, y_t] + b_c) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= s(W_o \cdot [h_{t-1}, y_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned}\right\} \quad (4)$$

In the equations above, f_t decides which part of the previous information to retain, while W_f and b_f are the weight and bias of the forget gate, respectively. The term $[h_{t-1}, y_t]$ represents the concatenation of the current input and the previous hidden state. The function s is the sigmoid activation function. The symbol $\otimes$ denotes point-wise multiplication, and the symbol $*$ denotes element-wise multiplication.

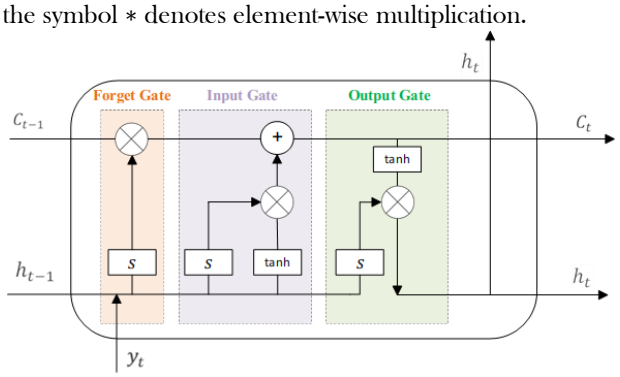


Figure 2. Standard Architecture of LSTM. Recreated Based on [23]

2.5. Hybrid Model

The LSTM architecture in this study is based on two architectures proposed in [12], see Figure 3. Such models are referred to as hybrid models. For example, a Baseline LSTM prediction model based on GARCH is named Baseline Neural-GARCH.

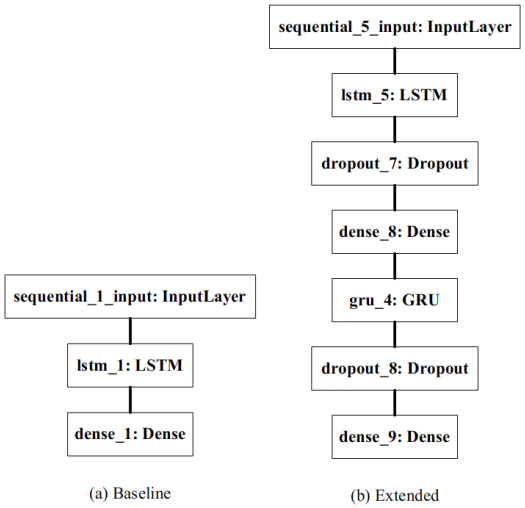


Figure 3. LSTM Architecture [12]

In this study, the Baseline LSTM model in Figure 3(a) employs a streamlined structure beginning with an InputLayer that receives two key features: (1) predicted volatility from a GARCH-type model (e.g., $\bar{\sigma}_1^2, \bar{\sigma}_2^2, \ldots, \bar{\sigma}_T^2$) and (2) exogenous realized volatility measures (e.g., $X_1, X_2, \ldots, X_T$). These are formatted as a 2D input matrix $U_t = [\bar{\sigma}_1^2 \quad X_t]$. The LSTM layer processes these inputs with 4 hidden units (where [12] uses five hidden units), generating hidden states:

$$h_t = \text{LSTM}\left(U_t; W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_c\right),$$

through gating mechanisms (forget, input, and output gates) to capture temporal dependencies in volatility patterns. This feeds directly into a final Dense layer (a fully connected neural layer) with linear activation that generates the volatility forecast as the output layer:

$$\hat{\sigma}_t^2 = W_d h_t + b_d$$

where W_d and b_d are learned during training via backpropagation. While computationally efficient, this minimalist design may oversimplify complex market dynamics.

In contrast, the Extended model in Figure 3(b) features a more sophisticated pipeline: its InputLayer normalizes the same inputs. The first LSTM layer employs 20 hidden to capture long-term dependencies, outputting sequential hidden states for first layer:

$$h_t^{(1)} = \text{LSTM}\left(U_t; W_f^{(1)}, W_i^{(1)}, W_o^{(1)}, W_c^{(1)}, b_f^{(1)}, b_i^{(1)}, b_o^{(1)}, b_c^{(1)}\right)$$

These features then undergo nonlinear transformation through a 10-unit Dense layer with ReLU activation (Rectified Linear Unit, which introduces nonlinearity while maintaining gradient flow):

$$z_t = \text{ReLU}\left(W_d h_t^{(1)} + b_d\right).$$

A subsequent 10-unit GRU (Gated Recurrent Unit) layer further refines temporal features into multiscale representations:

$$h_t^{(2)} = \text{GRU}\left(z_t; W_f^{(2)}, W_i^{(2)}, W_o^{(2)}, W_c^{(2)}, b_f^{(2)}, b_i^{(2)}, b_o^{(2)}, b_c^{(2)}\right)$$

before final dropout regularization and output. The output layer maps $h_t^{(2)}$ to $\hat{\sigma}_t^2$ through a final dense unit:

$$\hat{\sigma}_t^2 = W_o h_t^{(2)} + b_o.$$

This hierarchical design—particularly the ReLU-activated Dense layer's ability to model complex interactions and the GRU's multi-scale processing—proves essential for accurately reproducing volatility spikes while filtering market noise, as quantitatively demonstrated in our forecasting error metrics (Tables 2-4) and qualitatively shown in the trajectory comparisons (Figure 5). The architectural differences directly explain the Extended model's superior performance in handling asymmetric shocks and structural breaks that challenge traditional GARCH specifications.

The choice of 4 hidden units for the Baseline LSTM and 20 hidden units for the Extended LSTM was based on preliminary experiments and a trade-off between model complexity and computational efficiency. It should be noted that the variation in hidden units is not the primary focus of this study. Rather, this study aims to demonstrate that hybrid models, even with a specific number of hidden units, can outperform traditional GARCH-type models in volatility forecasting. The Baseline LSTM serves as a simple benchmark, while the Extended LSTM, with its increased capacity, shows improved performance in capturing complex volatility patterns to learn intricate nonlinear relationships, as evidenced by our empirical results.

2.6. Error Metrics

No universal loss function is suitable for all machine learning model. Since the type of problem worked on in LSTM is regression, the forecasting is evaluated based on regression losses as defined in[24], [25]. The loss functions used in this study are loss functions that always give non-negative values, namely Sum of Squared Errors (SSE), Mean Absolute Errors (MAE), and Root Mean Squared Error (RMSE). Lower error values indicate better model performance, representing smaller differences between predicted and actual values.

3. RESULT AND ANALYSIS

This section studies the performance of the Neural-GARCH(1,1)-type models and compares it to traditional GARCH-type using three stock indices, including the FTSE100, N225, and SP500, spanning the daily time period from January 2000 to December 2020. The selection of the 2000–2020 study period was carefully chosen to ensure comprehensive market coverage, encompassing multiple structural breaks including the dot-com bubble (2000–2002), global financial crisis (2007–2009), European debt crisis (2010–2012), and the COVID-19 market shock (2020). This extended timeframe spans both pre- and post-implementation periods of the Markets in Financial Instruments Directive (MiFID), the European Union's regulatory framework that significantly altered market transparency and trading practices when introduced in 2007. According to classifications by the International Monetary Fund, this period captures three complete business cycles, providing a balanced representation of various market conditions while avoiding potential recency bias. The inclusion of major indices from different economic systems further strengthens the study's robustness.

For the exogenous variables in our traditional models, we primarily employ 5-minute RV as it optimally balances accuracy and computational efficiency. Higher frequency data (e.g., 1-minute RV) tends to introduce microstructure noise that can distort volatility estimates, lower frequency measures (e.g., 10-minute RV10) may fail to capture important short-term market fluctuations. This choice aligns with established market microstructure theory and empirical findings demonstrating RV5's superior performance in volatility forecasting across different market conditions and asset classes [26].

3.1. Forecasting Construction

We evaluated the one-day-ahead conditional return volatility forecasting ability of competing GARCH models using a 500-day (approximately 2-year) test set for each daily return. Models were fitted to training data using the ARWM method in an MCMC algorithm, and the resulting parameter estimates were used for forecasting on the test data. All implementation was done in MATLAB.

Next, the Neural-GARCH-type models were compiled with an ADAM optimizer and a Mean Squared Error (MSE) loss function. The hyperparameters for both the Baseline and Extended LSTM architectures were

determined through a manual and iterative fine-tuning process, rather than a formal grid or random search. This pragmatic approach allowed for direct control over the model's complexity and architecture, enabling us to adapt the models based on preliminary training observations. The implementation was employed by using the Keras framework in Google Colaboratory, along with Pandas, Numpy, and Matplotlib. Both architectures consistently applied a dropout rate of 0.1 after each recurrent layer to mitigate overfitting, and were trained for 50 epochs with a batch size of 32. This manual tuning strategy was selected to achieve a balance between computational efficiency and model performance.

A flowchart of the forecasting steps is shown in Figure 4. Table 2–4 summarize the forecast performance of the traditional and hybrid models according to the three loss functions above. The bold numbers indicate the best predictive scores across the four volatility proxies, namely 5-minute Realized Volatility (RV5), 10-minute Realized Volatility (RV10), Two-Scale Realized Kernel (TSRK), and Parzen-type Realized Kernel (RKP), see [27] for the formula.

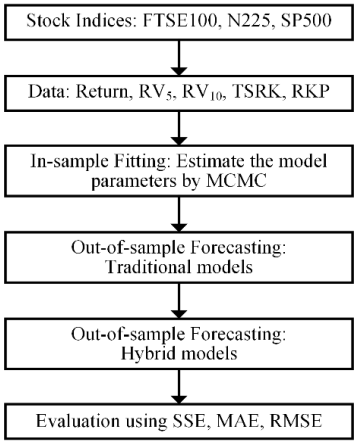


Figure 4. The Conceptual Framework of Forecasting

3.2. Performance Comparison between Traditional GARCH-type and Neural-GARCH-type Models

Table 2 presents the comparative evaluation results between traditional GARCH-type and Neural-GARCH-type models. The forecasting performance of competing models varies across stock indices and loss functions. The only exception is for the Extended version models adopting SP500, where the model consistently provide the best forecasting performance for all lost functions and all proxies.

The detailed analysis is as follows:

1. Comparing GARCH with Neural-GARCH, both Baseline and Extended versions show improvement in reducing errors across stock indices. Here, the Extended version outperforms the Baseline, suggesting better adaptability in capturing volatility dynamics.
2. For EG versus Neural-EG, EG outperforms Neural-EG on the adoption of FTSE100 data, especially in SSE and RMSE. On the other data adoptions, the Neural-EG models consistently outperform EG. In this regard, the Extended version demonstrates improvements across most indices and volatility proxies, confirming its effectiveness in refining predictions.
3. In the case of GJR versus Neural-GJR, the performance of GJR competes with Neural-GJR, especially in SSE and RMSE, in the adoption of FTSE100 data. GJR consistently performs better than the Baseline version of Neural-GJR in the SP500. However, in the adoption of N225 data, both Neural-GARCH models outperformed GJR in all cases. Overall, the Extended Neural-GJR model has the most accurate forecasts compared to the Baseline Neural-GJR and GJR models in many cases, highlighting its superior forecasting capability.

Overall, the comparison between GARCH-type traditional and their hybrid models indicates that hybrid models generally improve forecasting accuracy, particularly when using the Extended version.

Table 2. Forecasting Evaluation for the GARCH-type Models with RV5 as Exogenous Variable

Model	FTSE100				N255				SP500			
	Volatility Proxy:											
	RV_5	RV_{10}	TSRK	RKP	RV_5	RV_{10}	TSRK	RKP	RV_5	RV_{10}	TSRK	RKP
Loss function: SSE												
GARCH	116.7	**157.4**	121.1	139.4	68.0	83.8	58.4	92.6	105.5	108.7	74.5	102.0
Baseline Neural-GARCH	**109.0**	158.1	***48.3***	**134.6**	**49.0**	***66.4***	**35.8**	***74.5***	**82.2**	**83.9**	***59.4***	**77.5**
Extended Neural-GARCH	***100.4***	***145.4***	**65.0**	***123.6***	***47.8***	**67.0**	***34.1***	**77.5**	***74.2***	***77.0***	***47.2***	***69.1***
EG	***91.4***	***135.5***	80.0	***115.4***	52.8	69.3	42.1	78.0	95.3	98.4	64.1	90.6
Baseline Neural-EG	103.7	148.1	**73.7**	126.3	**47.6**	***65.1***	**35.0**	***72.3***	**87.2**	**85.9**	***57.7***	**77.2**
Extended Neural-EG	**99.3**	**145.1**	***64.0***	**122.9**	***47.4***	**67.6**	***33.4***	**77.5**	***70.1***	***73.8***	***43.1***	***66.6***
GJR	103.5	***128.1***	128.7	***116.5***	58.1	73.9	48.4	82.9	**84.6**	**87.8**	**60.2**	**83.5**
Baseline Neural-GJR	**103.4**	143.5	**56.1**	123.5	**50.0**	**66.6**	**37.5**	**74.7**	88.8	90.0	70.9	86.0
Extended Neural-GJR	***98.6***	**142.8**	***56.0***	**121.5**	***48.5***	***68.5***	***34.8***	***79.4***	***71.3***	***74.7***	***44.5***	***67.3***
Loss function: MAE												
GARCH	0.3206	0.3368	0.3861	0.3227	0.2548	0.2686	0.2501	0.2921	0.2902	0.3021	0.2784	0.3101
Baseline Neural-GARCH	**0.2647**	**0.2963**	***0.2182***	**0.2768**	**0.1846**	**0.2080**	**0.1591**	**0.2371**	**0.2725**	**0.2821**	**0.2511**	**0.2853**
Extended Neural-GARCH	***0.2632***	***0.2928***	**0.2679**	***0.2720***	***0.1742***	***0.1993***	***0.1469***	***0.2293***	***0.2271***	***0.2325***	***0.1980***	***0.2366***
EG	**0.2659**	**0.2912**	0.3181	**0.2721**	0.2180	0.2363	0.2076	0.2636	0.2673	0.2788	0.2570	0.2857
Baseline Neural-EG	0.2933	0.3203	**0.3121**	0.3010	**0.1913**	**0.2147**	**0.1678**	**0.2407**	**0.2553**	**0.2623**	**0.2265**	**0.2644**
Extended Neural-EG	***0.2592***	***0.2893***	***0.2613***	***0.2687***	***0.1749***	***0.1992***	***0.1460***	***0.2283***	***0.2218***	***0.2303***	***0.1947***	***0.2371***
GJR	0.2899	0.3067	0.3700	0.2925	0.2397	0.2554	0.2339	0.2824	0.2694	0.2806	0.2608	0.2887
Baseline Neural-GJR	**0.2615**	**0.2877**	**0.2273**	**0.2711**	**0.1861**	***0.2062***	**0.1574**	**0.2284**	**0.2983**	**0.3090**	**0.2876**	**0.3117**
Extended Neural-GJR	***0.2551***	***0.2834***	***0.2424***	***0.2659***	***0.1810***	**0.2063**	***0.1547***	***0.2374***	***0.2227***	***0.2314***	***0.1968***	***0.2372***
Loss function: RMSE												
GARCH	0.4831	**0.5610**	0.4920	0.5280	***0.2689***	0.4093	0.3417	0.4304	0.4593	0.4663	0.3861	0.4517
Baseline Neural-GARCH	**0.4669**	0.5623	**0.3107**	**0.5188**	0.3130	**0.3645**	**0.2677**	**0.3860**	**0.4055**	**0.4097**	**0.3446**	**0.3938**
Extended Neural-GARCH	***0.4482***	***0.5393***	***0.3606***	***0.4972***	**0.3094**	***0.3661***	***0.2612***	***0.3937***	***0.3854***	***0.3924***	***0.3071***	***0.3716***
EG	***0.4275***	***0.5206***	0.3999	***0.4803***	0.3249	0.3723	0.2901	0.3949	0.4367	0.4436	0.3580	0.4258
Baseline Neural-EG	0.4554	0.5442	**0.3838**	0.5026	**0.3087**	**0.3609**	**0.2646**	**0.2646**	**0.4176**	**0.4144**	**0.3397**	**0.3930**
Extended Neural-EG	**0.4456**	**0.5388**	***0.3579***	**0.4959**	***0.3078***	***0.3677***	***0.2583***	***0.3936***	***0.3746***	***0.3841***	***0.2936***	***0.3651***
GJR	0.4551	***0.5061***	0.5072	***0.4827***	0.3409	0.3844	0.3112	0.4071	**0.4114**	**0.4190**	**0.3469**	**0.4087**
Baseline Neural-GJR	**0.4550**	0.5357	**0.3357**	0.4971	**0.3161**	**0.3649**	**0.2737**	**0.3866**	0.4215	0.4243	0.3767	0.4148
Extended Neural-GJR	***0.4441***	**0.5345**	***0.3347***	**0.4929**	***0.3114***	***0.3701***	***0.2638***	***0.2638***	***0.3776***	***0.3866***	***0.2985***	***0.3668***

Notice: **Bold-italic numbers** indicate the smallest error, while **bold numbers** indicate the second smallest error.

a. Performance Comparison between Traditional GX-type and Neural-GX-type Models

Next, Table 3 presents the evaluation results of the comparison between traditional GX-type and Neural-GX-type models. Based on the results in Table 3, the comparison of forecasting performance among the GX-type models with exogenous variables highlights the advantages of Neural-based approaches.

The detailed analysis is as follows:

1. GX and Neural-GX, both Baseline and Extended Neural-GX improve forecasting accuracy across stock indices. The Extended version consistently reduces errors in all error metrics, making it the best-performing variant. The only exception is for the FTSE100 data series with the MAE metric, where the Baseline version outperforms the Extended version.
2. Comparing EGX with Neural-EGX, the Neural-EGX is superior with the exception of FTSE100 data with SSE and RMSE metrics. In the SP500 case, the Baseline version consistently reduces error better than the Extended version. In the other cases, the Extended version demonstrates significant error reductions across most loss functions and proxies, confirming its robustness in volatility forecasting.
3. In the case of GJRX versus Neural-GJRX, the Neural-GJRX shows improved accuracy in almost all cases.

Overall, the Extended version outperforms Baseline version, reinforcing the effectiveness of Extended version in enhancing volatility forecasting.

Table 3. Forecasting Evaluation for the GX-type Models with RV5 as Exogenous Variable

Model	FTSE100				N255				SP500			
	Volatility Proxy:											
	RV_5	RV_{10}	TSRK	RKP	RV_5	RV_{10}	TSRK	RKP	RV_5	RV_{10}	TSRK	RKP
Loss function: SSE												
GX	128.7	181.7	102.2	156.8	69.9	90.2	57.3	101.4	109.5	112.6	72.6	102.3
Baseline Neural-GX	106.9	153.9	*46.0*	131.2	51.8	68.5	39.7	*75.7*	85.5	83.4	60.6	76.8
Extended Neural-GX	*102.5*	*147.8*	63.0	*126.4*	*50.7*	*70.0*	*36.7*	79.8	*73.2*	*75.4*	*45.1*	*66.8*
EGARCHX	104.9	147.2	86.5	126.7	54.5	72.8	42.8	84.4	94.3	96.5	71.6	91.4
Baseline Neural-EGX	108.2	155.0	*45.1*	132.5	51.8	*69.3*	39.1	*75.7*	*75.3*	*75.5*	*52.3*	*68.0*
Extended Neural-EGX	*91.6*	*132.6*	60.3	*112.5*	*51.1*	71.4	*36.0*	80.9	86.9	85.3	62.8	78.2
GJRX	124.9	161.8	104.1	143.5	60.3	80.4	46.1	87.6	97.7	98.9	60.6	86.3
Baseline Neural-GJRX	102.9	*142.6*	59.8	122.4	46.2	63.3	33.5	72.1	84.1	84.4	53.7	73.6
Extended Neural-GJRX	*99.0*	143.5	*56.6*	*122.4*	*44.1*	*61.9*	*31.1*	*71.7*	*72.8*	*75.9*	*47.7*	*68.6*
Loss function: MAE												
GX	0.3178	0.3405	0.3618	0.3244	0.2600	0.2752	0.2465	0.3040	0.2915	0.3034	0.2751	0.3090
Baseline Neural-GX	*0.2615*	*0.2907*	*0.2126*	*0.2729*	0.1902	0.2131	0.1649	0.2396	0.2613	0.2704	0.2384	0.2718
Extended Neural-GX	0.2696	0.3025	0.2540	0.2818	*0.1746*	*0.1993*	*0.1468*	*0.2247*	*0.2295*	*0.2372*	*0.2019*	*0.2399*
EGX	0.2685	0.2940	0.3099	0.2769	0.2101	0.2343	0.1980	0.2656	0.2756	0.2868	0.2651	0.2945
Baseline Neural-EGX	0.2602	0.2889	*0.1968*	0.2720	0.1990	0.2215	0.1739	0.2459	*0.2505*	*0.2613*	*0.2254*	*0.2594*
Extended Neural-EGX	*0.2562*	*0.2869*	0.2622	*0.2663*	*0.1759*	*0.1997*	*0.1440*	*0.2242*	0.2743	0.2846	0.2581	0.2866
GJRX	0.2935	0.3139	0.2928	0.3035	0.1926	0.2124	0.1747	0.2305	0.2600	0.2682	0.2276	0.2689
Baseline Neural-GJRX	0.2640	0.2897	*0.2345*	0.2714	0.1907	0.2128	0.1627	0.2388	0.2580	0.2653	0.2253	0.2630
Extended Neural-GJRX	*0.2582*	*0.2881*	0.2464	*0.2702*	*0.1704*	*0.1967*	*0.1413*	*0.2231*	*0.2227*	*0.2300*	*0.1964*	*0.2346*
Loss function: RMSE												
GX	0.5074	0.6028	0.4522	0.5600	0.3739	0.4248	0.3386	0.4504	0.4680	0.4746	0.3811	0.4524
Baseline Neural-GX	0.4624	0.5547	*0.3033*	0.5123	0.3220	*0.3702*	0.2828	*0.3891*	0.4136	0.4083	0.3481	0.3919
Extended Neural-GX	*0.4527*	*0.5437*	0.3550	*0.5029*	*0.3185*	0.3742	*0.2710*	0.3995	*0.3827*	*0.3884*	*0.3004*	*0.3655*
EGX	0.4581	0.5426	0.4158	0.5034	0.3300	0.3816	0.2927	0.4107	0.4343	0.4393	0.3785	0.4275
Baseline Neural-EGX	0.4652	0.5568	*0.3003*	0.5148	0.3219	*0.3723*	0.2797	*0.3892*	*0.3882*	*0.3886*	*0.3235*	*0.3688*
Extended Neural-EGX	*0.4281*	*0.5150*	0.3474	*0.4743*	*0.3196*	0.3779	*0.2685*	0.4022	0.4169	0.4131	0.3543	0.3954
GJRX	0.4997	0.5689	0.4564	0.5357	0.3474	0.4011	0.3035	0.4186	0.4420	0.4447	0.3481	0.4155

Baseline Neural-GJRX	**0.4535**	***0.5341***	**0.3459**	**0.4948**	**0.3040**	**0.3557**	**0.2590**	**0.3798**	**0.4102**	**0.4110**	**0.3276**	**0.3836**
Extended Neural-GJRX	***0.4450***	**0.5358**	***0.3366***	***0.4948***	***0.2969***	***0.3520***	***0.2496***	***0.3788***	***0.3817***	***0.3897***	***0.3090***	***0.3705***

Notice: **Bold-italic numbers** indicate the smallest error, while **bold numbers** indicate the second smallest error.

b. Performance Comparison between Traditional RG-type and Neural-RG-type Models

Table 4 presents the evaluation results of forecasting comparison between traditional RG-type and Neural-RG-type models. The results highlights the advantages of Neural-based models.

The detailed analysis is as follows:

1. Between LRG and Neural-LRG, both Baseline and Extended versions improve forecasting accuracy, with the Extended version provide better predictive capability (except for SP500 data measured by SSE).
2. Comparing REG and Neural-REG, the Baseline version performs similarly to REG in the SP500 case and outperforms REG in the other data cases, but the Extended version significantly improves accuracy across all three indices, especially when adopting FTSE100 and N225 data. This reinforces the robustness of the Extended version in capturing volatility dynamics.
3. In the case of G@C versus Neural-G@C, G@C has the highest forecasting errors, while the Neural-G@C models substantially reduce errors, making them perform better across all loss functions.

Overall, Extended version demonstrate its superior ability to forecast volatility effectively.

Based on the analysis results from Tables 2–4, the Extended Neural models generally demonstrate improved performance compared to traditional models. Empirically, the performance of the Extended versions proves to be superior to both traditional models and the Baseline model. These findings differ from those in [12] due to variations in experimental settings.

Table 4. Forecasting Evaluation for the RG-type Models with RV5 as Exogenous Variable

Model	FTSE100				N225				SP500			
	Volatility Proxy:											
	RV_5	RV_{10}	TSRK	RKP	RV_5	RV_{10}	TSRK	RKP	RV_5	RV_{10}	TSRK	RKP
Loss function: SSE												
LRG	**105.7**	149.4	84.4	128.4	55.6	75.1	42.9	87.1	**80.7**	84.7	58.7	78.5
Baseline Neural-LRG	106.0	**145.0**	**78.9**	**125.1**	**49.7**	***66.9***	**37.1**	***74.8***	84.8	**84.2**	***55.8***	**76.5**
Extended Neural-LRG	***99.4***	***143.6***	***50.9***	***121.7***	***48.6***	**68.6**	***34.5***	**79.0**	***72.1***	***75.8***	***47.0***	***69.6***
REG	105.9	149.3	85.5	128.3	55.5	75.0	42.8	87.0	**80.6**	**84.6**	58.6	78.4
Baseline Neural-REG	**102.1**	**142.3**	**72.7**	**121.8**	**49.4**	***65.6***	**37.1**	***32.8***	85.1	84.8	**57.0**	**76.4**
Extended Neural-REG	***98.2***	***140.6***	***57.4***	***119.6***	***48.9***	**68.7**	***35.4***	**79.9**	***68.0***	***71.0***	***44.6***	***64.9***
G@C	168.5	244.3	90.4	209.7	92.7	115.3	76.2	128.0	160.1	163.8	111.5	145.3
Baseline Neural-G@C	**101.1**	***137.3***	**75.0**	***118.8***	**48.9**	***66.6***	**35.4**	***75.5***	**87.4**	**84.7**	**59.0**	**75.9**
Extended Neural-G@C	***99.7***	**143.6**	***66.8***	**122.0**	***47.7***	**67.6**	***33.7***	**78.0**	***72.9***	***74.7***	***43.8***	***66.6***
Loss function: MAE												
LRG	**0.2693**	**0.2937**	0.3103	**0.2767**	0.2185	0.2429	0.2032	0.2732	0.2573	0.2707	0.2454	0.2789
Baseline Neural-LRG	0.2831	0.3061	**0.2977**	0.2890	**0.1962**	**0.2199**	**0.1716**	**0.2468**	**0.2548**	**0.2621**	**0.2262**	**0.2641**
Extended Neural-LRG	***0.2509***	***0.2795***	***0.2150***	***0.2603***	***0.1744***	***0.2001***	***0.1440***	***0.2291***	***0.2344***	***0.2434***	***0.2146***	***0.2514***
REG	**0.2701**	**0.2945**	0.3129	**0.2775**	0.2179	0.2425	0.2024	0.2727	**0.2573**	**0.2708**	0.2454	0.2789
Baseline Neural-REG	0.2771	0.3032	**0.2896**	0.2841	***0.1838***	***0.2034***	***0.1581***	***0.1989***	0.2633	0.2730	**0.2403**	**0.2758**
Extended Neural-REG	***0.2546***	***0.2832***	***0.2310***	***0.2646***	**0.1875**	**0.2125**	**0.1619**	**0.2437**	***0.2222***	***0.2306***	***0.1945***	***0.2337***
G@C	0.3316	0.3607	0.3352	0.3418	0.3182	0.3364	0.3074	0.3602	0.3399	0.3512	0.3231	0.3517

Baseline Neural-G@C	**0.2750**	***0.3002***	**0.2910**	**0.2821**	**0.1917**	**0.2161**	**0.1658**	**0.2447**	**0.2527**	**0.2585**	**0.2232**	**0.2584**
Extended Neural-G@C	***0.2705***	**0.3005**	***0.2807***	***0.2788***	***0.1786***	***0.2032***	***0.1504***	***0.2334***	***0.2204***	***0.2267***	***0.1887***	***0.2297***
Loss function: RMSE												
LRG	**0.4599**	0.5467	0.4109	0.5067	0.3334	0.3875	0.2931	0.4173	**0.4017**	0.4115	0.3427	0.3962
Baseline Neural-LRG	0.4604	**0.5386**	**0.3971**	**0.5003**	**0.3153**	**0.3659**	**0.2724**	***0.3869***	0.4118	**0.4103**	**0.3339**	**0.3911**
Extended Neural-LRG	***0.4459***	***0.5359***	***0.3192***	***0.4933***	***0.3117***	***0.3703***	***0.2627***	**0.3976**	***0.3797***	***0.3894***	***0.3067***	***0.3731***
REG	0.4602	0.5464	0.4136	0.5066	0.3330	0.3872	0.2924	0.4170	**0.4016**	**0.4114**	0.3432	0.3960
Baseline Neural-REG	**0.4518**	**0.5335**	**0.3813**	**0.4935**	**0.3143**	***0.3621***	**0.2723**	***0.2562***	0.4125	0.4117	**0.3377**	**0.3910**
Extended Neural-REG	***0.4433***	***0.5302***	***0.3390***	***0.4890***	***0.3128***	**0.3708**	***0.2659***	**0.3998**	***0.3687***	***0.3768***	***0.2987***	***0.3603***
G@C	0.5805	0.6990	0.4251	0.6476	0.4306	0.4802	0.3904	0.5059	0.5659	0.5723	0.4722	0.5391
Baseline Neural-G@C	**0.4496**	***0.5241***	**0.3872**	***0.4874***	**0.3128**	***0.3649***	**0.2662**	***0.3886***	**0.4180**	**0.4116**	**0.3435**	**0.3896**
Extended Neural-G@C	***0.4464***	**0.5359**	***0.3656***	**0.4941**	***0.3088***	**0.3677**	***0.2597***	**0.3949**	***0.3819***	***0.3866***	***0.2959***	***0.3648***

Notice: **Bold-italic numbers** indicate the smallest error, while **bold numbers** indicate the second smallest error.

c. Visualization of Forecasting Dynamics

Figure 5 illustrates daily volatility forecasts for the S&P 500 index using RG-type and hybrid models. Unlike other GARCH and GX models, the G@C model fails to capture extreme volatility spikes and is outperformed by LRG and REG (contrary to [20] due to data differences). Both hybrid models offer smoother predictions while retaining main volatility trends. The **Extended hybrid model closely tracks actual volatility and remains reactive**, whereas the **Baseline model's over-smoothing** stems from its simpler architecture (single dropout, no ReLU or GRU).

d. Practical Implementation Insights

Our comparative analysis of GARCH-type and Neural-GARCH models reveals several key practical implications:

- Superior volatility forecasting: Neural-GARCH models, particularly the Extended versions, consistently provide more accurate volatility forecasts than traditional GARCH models across all stock indices examined. This demonstrates deep learning's (LSTM) effectiveness in capturing complex financial volatility patterns.
- Refined risk management: Enhanced forecast accuracy directly improves risk assessment for financial institutions and investors, leading to:
  - Precise derivatives pricing: Minimizing mispricing errors.
  - Optimized portfolio management: Enabling better asset allocation and hedging strategies.
  - Proactive capital protection: Serving as an early warning for volatility spikes.
- Robust decision-making: The sophisticated architectures of Extended hybrid models (e.g., ReLU, GRU, dropout layers) offer more responsive and accurate predictions, supporting confident and timely strategic decisions in dynamic markets.

In essence, these findings strongly recommend adopting Neural-GARCH models, especially their Extended configurations, as a more reliable tool for financial forecasting, offering significant advantages for risk management, investment optimization, and market stability.

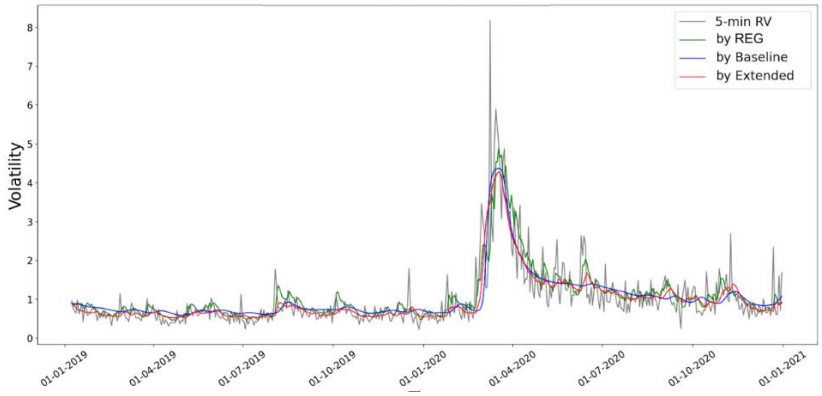


Figure 5. Time Series Plot of REG-type Models Forecasts

4. CONCLUSION

This study compared GARCH, GARCH-X, and Realized GARCH models with hybrid GARCH-LSTM models (Baseline and Extended LSTM) for volatility forecasting. Analyzing FTSE 100, Nikkei 225, and S&P 500 predictions, we found hybrid models generally offered improved performance. The Extended hybrid model was the most accurate, addressing the Baseline's over-smoothing with ReLU, GRU, and additional dropout layers.

These findings offer valuable practical applications for financial practitioners. First, portfolio managers can leverage more precise volatility predictions to optimize risk-adjusted asset allocation, particularly during periods of market turbulence. Second, derivatives traders can utilize improved volatility estimates to enhance option pricing models where volatility serves as a critical input parameter. Third, risk management teams can benefit from the model's enhanced responsiveness to volatility spikes when calculating dynamic Value-at-Risk metrics and determining margin requirements. From a regulatory perspective, our framework could strengthen stress testing scenarios by better capturing the nonlinear volatility dynamics that conventional models often miss.

Future research in financial time series forecasting should focus on several key areas. First, broader feature engineering is essential; incorporating sentiment indicators derived from news analytics (e.g., earnings call transcripts) and social media (Twitter/X financial sentiment indices) could capture important behavioral drivers of volatility that price data alone may miss. Second, we recommend investigating different asset classes like foreign exchange and cryptocurrencies, as neural networks may not generalize across diverse market dynamics. Third, evaluating hybrid models' forecasting capability for Value-at-Risk (VaR) estimation is a critical next step for financial risk management. Lastly, refining model architectures through hyperparameter tuning (e.g., epochs, batch sizes, memory lengths, forget gates, LSTM layers, dropout layers, activation functions) would further bridge the gap between academic research and industry practice.

5. REFERENCE

[1] T. Bollerslev, "The story of GARCH: A personal odyssey," *J Econom*, vol. 234, pp. 96–100, 2023, doi: 10.1016/J.JECONOM.2023.01.015.

[2] D. B. Nugroho, J. Wijaya, and A. Setiawan, "Modeling of returns volatility through EGARCH model using high-frequency data," *J Appl Probab Stat*, vol. 18, no. 2, pp. 55–73, 2023.

[3] D. B. Nugroho, L. P. Panjaitan, D. Kurniawati, Z. Kholil, B. Susanto, and L. R. Sasongko, "GRG non-linear and ARWM methods for estimating the GARCH-M, GJR, and log-GARCH models," *Jurnal Teori Apl Mat*, vol. 6, no. 2, pp. 448–460, 2022, doi: 10.31764/jtam.v6i2.7694.

[4] D. B. Nugroho, H. Wibowo, and A. Saragih, "Modeling daily return volatility through GJR(1,1) model and realized volatility measure," *Thail Stat*, vol. 22, no. 1, pp. 50–62, 2024, [Online]. Available: https://ph02.tci-thaijo.org/index.php/thaistat/article/view/252221

[5] D. B. Nugroho, B. A. A. Wicaksono, and L. Larwuy, "GARCH-X(1,1) model allowing a non-linear function of the variance to follow an AR(1) process," *Commun Stat Appl Methods*, vol. 30, no. 2, pp. 163–178, 2023, doi: 10.29220/CSAM.2023.30.2.163.

[6] D. B. Nugroho, O. C. Dimitrio, and F. Tita, "The GARCH-X(1,1) model with exponentially transformed exogenous variables," *J Sains Teknol*, vol. 12, no. 1, pp. 65–72, 2023, doi: 10.23887/jstundiksha.v12i1.50714.

[7] D. B. Nugroho, M. T. Wijaya, and H. A. Parhusip, "Modeling and estimating GARCH-X and Realized GARCH using ARWM and GRG methods," *Int J Comput Sci Appl Math*, vol. 11, no. 1, pp. 14–20, 2025, doi: 10.12962/J24775401.V11I1.20333.

[8] A. Moghar and M. Hamiche, "Stock market prediction using LSTM Recurrent Neural Network," *Procedia Comput Sci*, vol. 170, pp. 1168–1173, 2020, doi: 10.1016/J.PROCS.2020.03.049.

[9] R. Qiao, W. Chen, and Y. Qiao, "Prediction of stock return by LSTM neural network," *Appl Artif Intell*, vol. 36, no. 1, p. 2151159, 2022, doi: 10.1080/08839514.2022.2151159.

[10] K. Kakade, I. Jain, and A. K. Mishra, "Value-at-Risk forecasting: A hybrid ensemble learning GARCH-LSTM based approach," *Resour Policy*, vol. 78, p. 102903, 2022, doi: 10.1016/J.RESOURPOL.2022.102903.

[11] H. T. Araya, J. Aduda, and T. Berhane, "A hybrid GARCH and deep learning method for volatility prediction," *J Appl Math*, vol. 2024, no. 1, p. 6305525, 2024, doi: 10.1155/2024/6305525.

[12] J. C. Sullivan, "Stock price volatility prediction with long short-term memory neural networks," in *Comput Sci*, 2019.

[13] D. S. Kambouroudis, D. G. McMillan, and K. Tsakou, "Forecasting realized volatility: The role of implied volatility, leverage effect, overnight returns, and volatility of realized volatility," *J Futures Mark*, vol. 41, no. 10, pp. 1618–1639, 2021, doi: 10.1002/FUT.22241.

[14] E. S. Gunnarsson, H. R. Isern, A. Kaloudis, M. Risstad, B. Vigdel, and S. Westgaard, "Prediction of realized volatility and implied volatility indices using AI and machine learning: A review," *Int Rev Financ Anal*, vol. 93, p. 103221, 2024, doi: 10.1016/J.IRFA.2024.103221.

[15] X. Wu, J. Pu, and Y. Wang, "Forecasting VIX using realized EGARCH model with dynamic jumps," *Appl Econ Lett*, 2024, doi: 10.1080/13504851.2024.2308565.

[16] R. Gerlach, A. Naimoli, and G. Storti, "Time-varying parameters realized GARCH models for tracking attenuation bias in volatility dynamics," *Quant Financ*, vol. 20, no. 11, pp. 1849–1878, 2020, doi: 10.1080/14697688.2020.1751257.

[17] R. G. S. Queiroz and S. A. David, "Performance of the Realized-GARCH Model against other GARCH Types in predicting cryptocurrency volatility," *Risks*, vol. 11, no. 211, 2023, doi: 10.3390/risks11120211.

[18] P. R. Hansen and Z. Huang, "Exponential GARCH Modeling with Realized Measures of Volatility," *J Bus Econ Stat*, 2016, doi: 10.1080/07350015.2015.1038543

[19] A. Naimoli, R. Gerlach, and G. Storti, "Improving the accuracy of tail risk forecasting models by combining several realized volatility estimators," *Econ Model*, vol. 107, p. 105701, 2022, doi: 10.1016/J.ECONMOD.2021.105701.

[20] H. Xie and C. Yu, "Realized GARCH models: Simpler is better," *Financ Res Lett*, vol. 33, p. 101221, 2020, doi: 10.1016/J.FRL.2019.06.019.

[21] D. A. Spade, "Markov chain Monte Carlo methods: Theory and practice," *Handb Stat*, vol. 43, pp. 1–66, 2020, doi: 10.1016/BS.HOST.2019.06.001.

[22] L. Ni *et al.*, "Streamflow and rainfall forecasting by two long short-term memory-based models," *J Hydrol (Amst)*, vol. 583, p. 124296, 2020, doi: 10.1016/J.JHYDROL.2019.124296.

[23] H. Alizadegan, B. R. Malki, A. Radmehr, H. Karimi, and M. A. Ilani, "Comparative study of long short-term memory (LSTM), bidirectional LSTM, and traditional machine learning approaches for energy consumption prediction," *Energ Explor Exploit*, vol. 43, no. 1, pp. 281–301, 2025, doi: 10.1177/01445987241269496.

[24] N. P. Dharani, "Analysis and prediction of COVID-19 by using recurrent LSTM neural network model in machine learning," *Int J Adv Comput Sci Appl*, vol. 13, no. 5, pp. 171–178, 2022.

[25] C. H. Kim and Y. C. Kim, "Application of artificial neural network over nickel-based catalyst for Combined Steam-Carbon Dioxide of Methane Reforming (CSDRM)," *J Nanosci Nanotechnol*, vol. 20, no. 9, pp. 5716–5719, 2020, doi: 10.1166/JNN.2020.17627.

[26] Y. Lyu, Z. Yang, Y. Luo, Z. Qin, H. Yi, and R. Ke, "Forecasting the volatility of crude oil futures market: Does the simple 5-minute RV hold up?," *Energy Econ*, vol. 146, p. 108509, 2025, doi: 10.1016/J.ENECO.2025.108509.

[27] C. Floros, K. Gkillas, C. Konstantatos, and A. Tsagkanos, "Realized measures to explain volatility changes over time," *J Risk Financ Manag*, vol. 13, no. 6, p. 125, 2020, doi: 10.3390/jrfm13060125.