Wiley
Journal of Applied Mathematics
Volume 2024, Article ID 6305525, 19 pages
https://doi.org/10.1155/2024/6305525

Research Article

A Hybrid GARCH and Deep Learning Method for Volatility Prediction

Hailabe T. Araya,[1,2] **Jane Aduda**,[3] **and Tesfahun Berhane**[4]

[1]*Department of Mathematics, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi 62000, Kenya*
[2]*Department of Mathematics, Debre Markos University, Debre Markos 269, Ethiopia*
[3]*Department of Statistics and Actuarial Sciences, Jomo Kenyatta University of Agriculture and Technology, Nairobi 62000, Kenya*
[4]*Department of Mathematics, Bahir Dar University, Bahir Dar 26, Ethiopia*

Correspondence should be addressed to Hailabe T. Araya; hayle899@gmail.com

Received 21 February 2024; Revised 5 June 2024; Accepted 15 June 2024

Academic Editor: Man Leung Wong

Volatility prediction plays a vital role in financial data. The time series movements of stock prices are commonly characterized as highly nonlinear and volatile. This study is aimed at enhancing the accuracy of return volatility forecasts for stock prices by investigating the prediction of their price volatility through the integration of diverse models. Thus, the study integrated four powerful methods: seasonal autoregressive (AR) integrated moving average (MA), generalized AR conditional heteroskedasticity (ARCH) family models, convolutional neural network (CNN), and bidirectional long short-term memory (LSTM) network. The hybrid model was developed using the residuals generated by the seasonal AR integrated MA model as input for the generalized ARCH model. Following this, the estimated volatility obtained was utilized as an input feature for both the hybrid CNNs and bidirectional LSTM models. The model's forecasting performance was assessed using key evaluation metrics, including mean absolute error (MAE) and root mean squared error (RMSE). Compared to other hybrid models, our new proposed hybrid model demonstrates an average reduction in MAE and RMSE of 60.35% and 60.61%, respectively. The experimental results show that the model proposed in this study has good performance and accuracy in predicting the volatility of stock prices. These findings offer valuable insights for financial data analysis and risk management strategies.

Keywords: deep learning; GARCH-family models; hybrid model; volatility

1. Introduction

In financial terms, volatility denotes the degree of uncertainty or risk associated with fluctuations in the value of a security. Some securities exhibit high volatility, indicating significant fluctuations in their values over a wide range, while others demonstrate lower volatility, with their values varying within a narrower range. The fluctuations in securities' returns are not directly observable, leading traders, institutional investors, and other market participants to understand the relationship between returns and volatility [1]. The global expansion of stock markets has sparked interest among researchers and practitioners in modeling the volatility of stock returns. Modeling volatility plays a crucial role in designing investment strategies to mitigate risk and enhance stock returns and in pricing securities and options [2]. Moreover, its significance extends beyond investors and market participants to encompass the broader economy. High levels of volatility can disrupt the stability of capital markets, affect currency values, and impede international trade [3]. In light of these consequential challenges, the necessity of forecasting volatility emerges as an imperative, ensuring investors can adeptly navigate the complexities of financial markets and make informed, stable investment decisions. Therefore, in this paper, the aim was to obtain an optimal model (a hybrid of GARCH and deep learning models) to predict volatility in stock prices.

The forecasting of volatility in financial instruments has been extensively examined in recent decades, primarily due to its role as an indicator that enables the estimation of risk associated with the asset within a specified time frame. In recent years, substantial literature has emerged on modeling and predicting volatility in financial markets. Primarily, Engle [4] developed the autoregressive (AR) conditional heteroskedasticity (ARCH) model by incorporating conditional variance and modeling the serial correlation of returns as a function of past errors and changing time. This was carried out as part of Engle's attempts to explain how inflation dynamics operate in the United Kingdom.

To enhance Engle's model, the GARCH models were developed by Bollerslev [5]. This enhancement involved incorporating a long memory and creating a more flexible lag structure by adding lagged conditional variance to the original model. The standard GARCH (SGARCH) model cannot model the leverage effect because its specifications assume that the variance depends on the shock's magnitude and is independent of its sign [6].

Later, various adaptable GARCH models were introduced, incorporating additional parameters to capture the asymmetric behavior of time series data such as the exponential GARCH (EGARCH) model proposed by Nelson [7] and the threshold GARCH (TGARCH) model introduced by Zakoian [8].

B. Almansour, Alshater, and A. Almansour's [9] research focused on assessing the effectiveness of ARCH and GARCH models in predicting volatility within the cryptocurrency market. The findings indicated that both positive and negative news events have a notable impact on conditional volatility across various cryptocurrency markets. Additionally, the study concluded that the GARCH model demonstrates promising predictive capabilities for cryptocurrency price movements in the market.

Franses and Van Dijk [10] conducted a study on forecasting stock market volatility using the nonlinear GARCH method. The investigation involved an analysis of the GARCH model and two of its nonlinear modifications to forecast weekly stock market volatility. The findings of the study indicated that the quadratic generalized ARCH (QGARCH) model was the most effective for forecasting the volatility of the stock market.

Sen, Mehtab, and Dutta [11] predicted the volatility of stocks from selected sectors of the National Stock Exchange (NSE) of the Indian economy using GARCH. The researchers found that asymmetric GARCH models, notably, provide more precise forecasts regarding the future volatility levels of the selected stocks. In a modified model, specifically concerning ARIMA-GARCH modeling as demonstrated by Aduda et al. [12], there has been an exploration of using the residuals of ARIMA as a vital factor for improving forecasting in GARCH.

Implementing deep learning models in financial markets, especially in stock markets, has become a burgeoning research subject in recent times due to the increasing use of artificial intelligence (AI) in various fields. Artificial neural network (ANN) models can capture the nonlinearity of the series, do not require the series to be stationary for modeling, and perform better in volatility forecasts than SGARCH-type models. Liu [13] showed that for a considerable time interval, volatility predictions for the Standard & Poor's 500 (S&P 500) and Apple Inc. indicate that the long short-term memory (LSTM) can outperform the GARCH model.

In recent years, the field of financial time series analysis has witnessed a growing interest in the development of hybrid models that combine various deep-learning techniques with statistical models to enhance the accuracy of volatility prediction. Kim and Won [14] developed a hybrid model to predict the volatility of the Korea Composite Stock Price Index (KOSPI 200) by integrating GARCH-type models with the LSTM model.

Kakade et al. [15] investigated the advantage of hybridizing GARCH-type models with LSTM to predict the volatility of metals in the Indian commodity market. The study found that hybrid GARCH-LSTM models outperform standalone models. Vidal and Kristjanpoller [16] studied gold volatility prediction using a hybrid CNN-LSTM approach. This study found that the hybrid CNN-LSTM model outperforms the GARCH and LSTM models in forecasting the volatility of gold.

Zeng et al. [17] studied a natural gas load volatility prediction model by combining GARCH family models, the eXtreme Gradient Boosting (XGBoost) algorithm, and the LSTM network. Mademlis and Dritsakis [18] presented two hybrid models in the investigation of predictive models for the volatility of the Financial Times Stock Exchange Milano Italia Borsa (FTSE MIB) index and evaluated their efficacy alongside an asymmetric GARCH model and a neural network.

All the above studies have their limitations; for instance, hybrid SARIMA-GARCH family models for volatility forecasting often face limitations when confronted with nonlinear sequences and influential factors [19]. Moreover, a single Convolutional neural network (CNN) model has a poor interpretation of volatility. Therefore, prediction accuracy will not be high.

Combining hybrid econometric models such as SARIMA-GARCH family models with CNN models can effectively solve the shortcomings in volatility forecasting. CNN excels GARCH family models in capturing complex temporal patterns and nonlinear dependencies in volatility structures. Furthermore, CNN can automatically identify hierarchical features from the data, aiding in the extraction of meaningful representations and proving robust to noisy data.

However, integrating CNN with GARCH family models alone often fails to yield superior prediction results, as an abundance of feature inputs can degrade model performance. BiLSTM excels CNN in capturing long-term dependencies and sequential patterns in the data. By utilizing both forward and backward information patterns of the market, BiLSTM enhances accuracy in time series prediction. Its memory cells effectively handle data prone to irregular and seasonal patterns.

In response to this research gap, the present study introduces a novel hybrid SARIMA-GARCH-CNN-BiLSTM model. The forecasting performance of this new model is evaluated. The inclusion of BiLSTM in the extended model provides a significant improvement over the unidirectional approach of traditional LSTM models. The findings of this study provide valuable insights into understanding the movement and comovement of volatility prices, which is essential for investors, traders, and risk managers in navigating and mitigating the high-risk nature of these investments.

2. Methodology

The study used the SARIMA, GARCH family, CNN, and LSTM as components to create a new hybrid model. Each of these components has unique traits that may be extracted from historical data.

2.1. SARIMA Model. The seasonal AR integrated MA model, SARIMA $(p, d, q)(P, D, Q)_s$ is a member of the Box–Jenkins family of time series forecasting models for the nonstationary data series. The SARIMA$(p, d, q)(P, D, Q)_s$ can be written as

$$\Phi_P(B^s)\phi_p(B)\nabla^d\nabla_s^D(r_t - \mu) = \Theta_Q(B^s)\theta_q(B)\varepsilon_t \tag{1}$$

where the nonseasonal AR and MA components are represented by $\theta_q(B) = 1 + \sum_{i=1}^{q}\theta_i B^i$ and $\phi_p(B) = 1 - \sum_{j=1}^{p}\phi_j B^j$, respectively. The seasonal AR and MA components are represented by $\Phi_P(B^s) = 1 - \sum_{i=1}^{P}\Phi_i B^{\mathrm{is}}$ and $\Theta_Q(B^s) = 1 + \sum_{j=1}^{Q}\Theta_j B^{js}$, respectively. r_t represents the time series, and ε_t denotes the random error at time period t, where μ is the mean of the model. ∇^d and ∇_s^D represent the nonseasonal and seasonal differencing operators defined as $\nabla^d r_t = (1 - B)^d r_t$ and $\nabla_s^D r_t = (1 - B^s)^D r_t$.

2.2. SGARCH Model. The SGARCH(r, s) model is one in which the variance of the error term of the SARIMA model follows a GARCH process. The model used for the returns series is represented as follows: the error term ε_t is equal to $z_t\sigma_t$, where z_t is independent and identically distributed (i.i.d.) with $\mathbb{E}[z_t] = 0$ and $\mathrm{Var}[z_t] = 1$. The variance σ_t^2 is determined by the following equation:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{r}\alpha_i\varepsilon_{t-i}^2 + \sum_{j=1}^{s}\beta_j\sigma_{t-j}^2 \tag{2}$$

where $\alpha_0 > 0$, $\alpha_i > 0$, and $\beta_i > 0$ are constants.

2.3. EGARCH Model. The EGARCH model can accurately evaluate an asymmetric distribution and also quantify the increased impact of significant shocks on volatility [7]. The conditional mean equation of EGARCH is the same as above. The EGARCH which includes positive and negative asymmetric effects on returns is expressed as

$$\ln\sigma_t^2 = \alpha_0 + \sum_{j=1}^{p}\beta_j\ln\sigma_{t-i}^2 + \sum_{i=1}^{q}\left(\alpha_i\left|\frac{\varepsilon_{t-i}}{\sigma_{t-i}}\right| + r_i\frac{\varepsilon_{t-i}}{\sigma_{t-i}}\right) \tag{3}$$

where $\varepsilon_{t-i} > 0$ and $\varepsilon_{t-i} < 0$ imply good news and bad news and their overall effects are $(\alpha_i + \gamma_i)\cdot|\varepsilon_{t-i}|$ and $(\alpha_i - \gamma_i)\cdot|\varepsilon_{t-i}|$, respectively.

The EGARCH model achieves covariance stationarity when $\sum_{j=1}^{p}\beta_j < 1$.

2.4. TGARCH Model. The TGARCH model is an asymmetric model developed to treat news such as mergers and launches of discovery by introducing multiplicative dummy variables to the variance equation. This model was developed by the works of [20, 21]. The generalized specification of the TGARCH (r, s) model for the conditional variance equation is expressed as

$$\sigma_t^2 = \alpha_0 + \sum_{j=1}^{p}\beta_j\sigma_{t-j}^2 + \sum_{i=1}^{q}(\alpha_i + \gamma_i I_{t-i})\varepsilon_{t-i}^2 \tag{4}$$

where α_0, α_i, β_j, and γ_i are parameters that will be estimated for evaluation. I_{t-1} is an indicator dummy variable such that $I_t - 1 = 0$ for $\varepsilon_{t-i} \geq 0$ and 1 for $\varepsilon_{t-i} < 0$. When $\varepsilon_{t-i} \geq 0$ and $\varepsilon_{t-i} < 0$, the total contribution to the volatility is $\alpha_i\varepsilon_{t-i}^2$ and $(\alpha_i + \gamma_i)\varepsilon_{t-i}^2$, respectively. Furthermore, when $I_{t-i} = 0$, the model becomes SGARCH. Thus, these two pieces of news, of equal length, have different effects on conditional volatility. When $\gamma_i > 0$, bad news causes volatility to increase, which leads to a leveraging impact in the i-th order. Positive shocks with equal size raised conditional volatility more than negative shocks when $\gamma_i < 0$. Positive conditional volatility occurs when $\alpha_0 > 0$, $\beta_j \geq 0$, $\alpha_i \geq 0$, and $\alpha_i + \gamma_i \geq 0$ are all nonnegative. According to Poon [22], the TGARCH model is stationary if $\sum_{i=1}^{p}(\alpha_i + (\gamma_i/2)) + \sum_{j=1}^{q}\beta_j < 1$.

2.4.1. Model Selection Criteria. The study employed the most widely used model selection method, the Akaike information criterion (AIC) [23], to determine which GARCH model fits the data. The best possible model was selected based on the AIC scores of the models. AIC balances the goodness of fit against the complexity of the model. Lower AIC values indicate a better trade-off between fit and complexity.

2.5. CNN Model. Convolution is a mathematical process that takes two functions and produces a third, which is typically understood to be a filtered or modified version of one of the original functions [24]. One convolution operand, $f[n]$, corresponds to the filter, $h[n]$, with which we process the signal. The convolution procedure involves carrying out K multiplications and $K - 1$ sums for every value of the signal when the filter is finite and only specified in the domain $\{0, 1, \cdots, K - 1\}$ [25]. This can be mathematically represented as

$$(f * h)(n) = \sum_{k=0}^{K-1} h[k] f[n-k] \tag{5}$$

This operation is called convolution.

CNNs are specialized neural networks designed for processing inputs with inherent spatial structures. Renowned for efficacy across diverse data formats such as one-dimensional time series data, two-dimensional image data, and three-dimensional video data [26], the study used the 1D-CNN model for sequential data. The network uses algorithms for data mining to automatically recognize and pick the most critical features from the raw data. The sequential 1D-CNN model was constructed using convolutional layers, pooling layers, and dense layers. The input was a 3D tensor with shape (batch size, time steps, and input dim).

The role of the convolutional layer in the CNN model is to identify temporal patterns and relationships within sequential data, such as time series [27]. The role of the pooling layer, on the other hand, is to perform downsampling to address computational complexity and achieve translation invariance, enabling the model to identify features regardless of location in the input. Furthermore, downsampling helps to minimize the likelihood of overfitting, increase computational efficiency, and decrease the number of parameters. CNNs combine concepts such as weight sharing and local connectivity to improve the model's ability to do complex tasks. Relu is a popular activation function in CNNs due to its nonlinearity, computational efficiency, and ability to solve vanishing gradient problems of the training model in deep learning. While CNNs are effective models for time series forecasting applications, overfitting is a problem that can affect them. The overfitting problem can arise from factors such as the complexity of the model and highly correlated training data. Dropout is a regularization technique used in neural networks to prevent overfitting.

Feature maps in 1D-CNNs serve as representations of extracted features from time series data. These maps are generated through convolutional filters, capturing specific patterns and structures inherent in the data. Convolutional filters, also known as kernels, are small matrices used in CNNs to extract features from input data. These filters slide or convolve across the input data, performing mathematical operations to capture patterns and features at different locations. Each feature map corresponds to learned features within the time series, such as trends or periodicities. As the data propagates through the layers, deeper layers learn more abstract features, building upon earlier representations. In a CNN's convolutional layer, features from the previous layer are combined with learnable kernels and activation functions like a hyperbolic tangent, sigmoid, and Relu to produce feature maps [28]. As such, each feature map output is combined with more than one input feature map. In general, the convolved features at the output of the l-th layer can be written as shown in Equation (6) [28].

$$C_i^l = \mathrm{Rel}\left(b_i^l + \sum_j C_j^{l-1} \cdot K_{ij}^l\right) \tag{6}$$

In this context l-th layer, C_i^l denotes the i-th feature, C_j^{l-1} refers to the j-th feature in the preceding $(l-1)$-th layer, K_{ij}^l signifies the kernel connecting the i-th to the j-th features, b_i^l is the bias associated with these features, and Relu stands for the activation function.

Figure 1 illustrates a diagram representing how feature maps are formed in a CNN model. In the diagram, u represents a sample filter with adjustable weights of Size 3. Each C_i denotes the i-th element of the feature map. N stands for the number of 1D data tensor units, and f represents the filter size with a stride of 1.

According to Rala Cordeiro et al. [29], the stride value defines how the kernel moves in the input data. The most common value is one, meaning that the kernel moves over one column of the input data at each iteration. After convolution, pooling reduces dimensionality and improves feature robustness. Pooling size corresponds to input feature units. It applies a function to multiple inputs (convolutional features). Max pooling defines the pooling layer. This study used two CNN layers to strike a balance between complexity and performance. This approach reduces the risk of overfitting by avoiding excessive complexity and mitigating challenges like vanishing or exploding gradients during training. Previous research, like [30], may support the efficacy of two-layer CNN architectures for the task at hand. The decision reflects a thoughtful consideration of computational efficiency and training stability. Figure 2 depicts the structure of a 1D-CNN model with two convolutional and two max pooling layers. The output of the last pooling layer is flattened and connected to a dense layer with N units. Following the final pooling layer, the output is flattened and connected to a dense layer containing N units. Subsequently, this dense layer is connected to the final output layer, which consists of a single neuron.

2.6. BiLSTM Model. Hochreiter and Schmidhuber [31] first proposed the LSTM architecture as a specific type of recurrent neural network (RNN) to overcome the limitations of traditional RNNs in capturing and learning long-term dependencies in sequential data. While LSTM captures information from extended periods, the acquired data pertains to the time before the output moment, which lacks reverse information. However, for time series prediction, it is crucial to consider both backward and forward information patterns to enhance predictive performance. The two LSTMs that make up BiLSTM are forward and reverse. In contrast to the regular LSTM's one-way state transfer, the BiLSTM takes into account the data's changing laws both before and after data transmission, enabling it to make more thorough and precise decisions by utilizing both past and future knowledge. It has performed better than expected.

Given the input sequence $x = (x_1, x_2, \cdots, x_T)$, the hidden layer sequence $h = (h_1, h_2, \cdots, h_T)$ and the network output vector $y = (y_1, y_2, y_T)$ of the standard BiLSTM model are

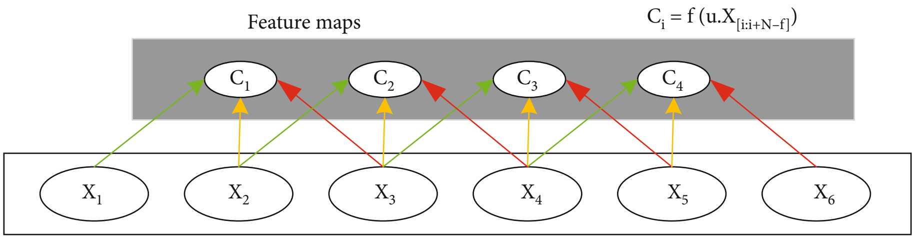


Figure 1: Generation of feature maps of convolution layers in 1D-CNN.

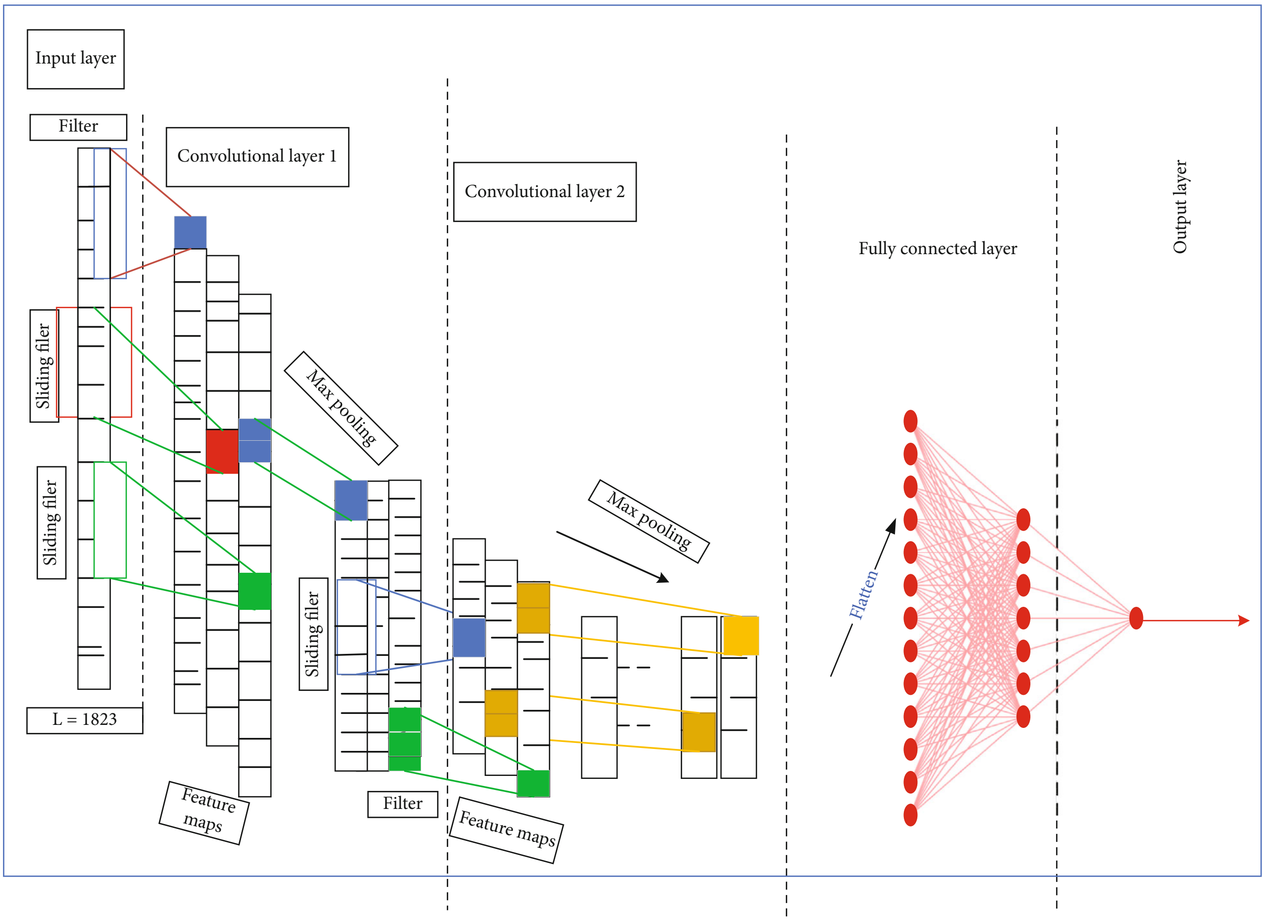


Figure 2: Simplified view of the 1D-CNN model for Apple Inc. data.

iteratively calculated from $t = 1$ to $t = T$. The updated memory cell can calculate the current hidden state h_t through the following formulas:

$$i_t = \sigma(W_i[\cdot h_{t-1}, x_t] + b_i) \tag{7}$$

$$\tilde{C}_t = \tanh\ (W_c \cdot [h_{t-1}, x_t] + b_c) \tag{8}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{9}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{10}$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

$$h_t = O_t \cdot \tanh\ (C_t) \tag{12}$$

In Equation (9), denoted as f_t, the forget gate mechanism. The sigmoid activation function σ was used to judge whether the last memory needs to be retained for the current memory state. Equation (7) describes the computation of i_t, serving as the input gate to assess the significance of retaining current input data. Equation (8) describes the calculation of $\tilde{C}_t$, which is used to calculate the data that needs to be updated. Equation (10) shows whether the state at the current moment needs to be updated. After a new state was obtained, Equation (10) was used to calculate the output gate value O_t.

The predicted value of the system can be given by the linear activation function as

$$y_t = W_{hy} \cdot h_t + b_y \tag{13}$$

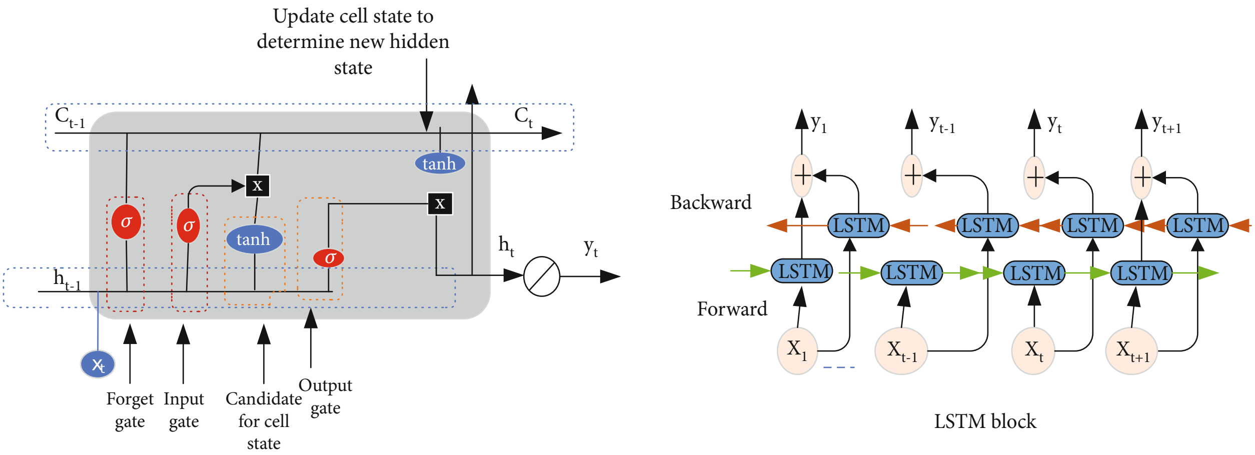


Figure 3: Internal structure of the BiLSTM model.

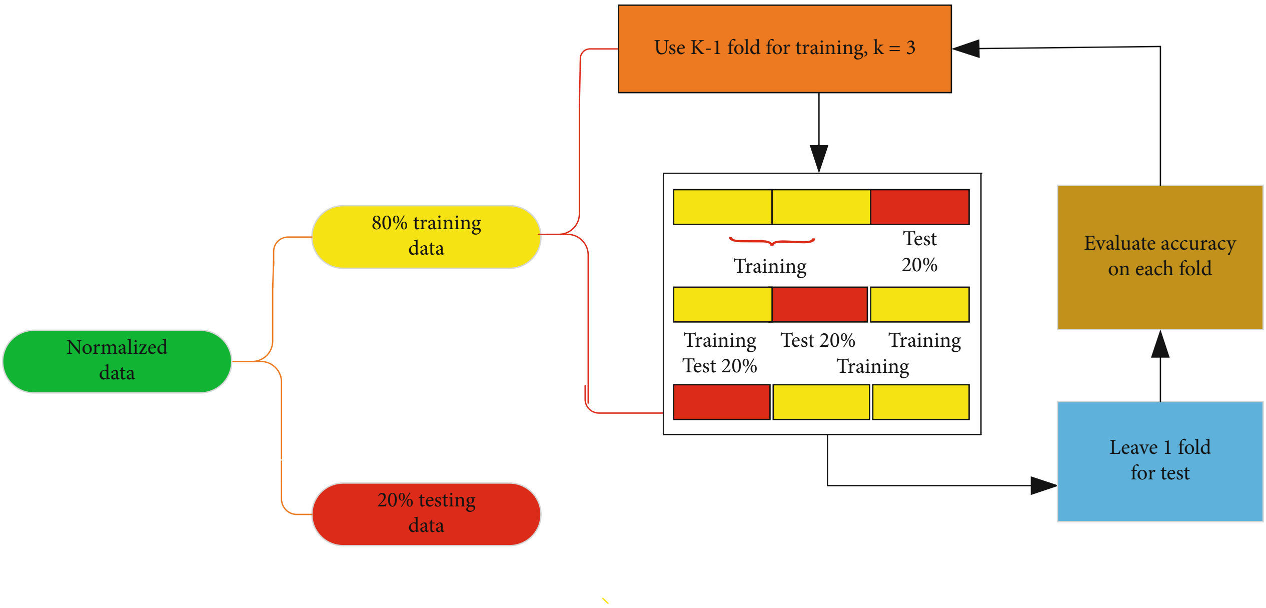


Figure 4: Data division criteria.

Figure 3 illustrates the internal organization of the BiLSTM model, constructed with LSTM blocks. BiLSTM, comprising both forward and backward LSTM components, necessitates a reversal of the computation.

2.7. SARIMA-GARCH-CNN-BiLSTM. The study introduced a novel approach to volatility forecasting called the hybrid SARIMA-GARCH-CNN-BiLSTM Model. First, the mean model was built using SARIMA, which was well known for its ability to detect seasonal and temporal trends in financial time series data. Second, the volatility was estimated using three models from the GARCH family: GARCH, TGARCH, and EGARCH. This makes it easier to choose the most accurate model among the GARCH variations. As a result, asymmetry and time-varying patterns can be captured. SARIMA residuals were used as input to estimate volatility in the GARCH family models. Lastly, CNN and BiLSTM architectures receive the predicted output of the GARCH family models. CNN effectively captures spatial dependencies, while BiLSTM excels at capturing long-term dependencies, leveraging both temporal and spatial features for improved forecasting.

2.7.1. Model Development Procedure. The overall process of modeling followed the following algorithms.

2.7.2. Data Preprocessing

1. Filling missing data. Due to the closure of financial markets on weekends (Saturdays and Sundays) and public holidays, missing values may occur in stock datasets. Additionally, issues with processing and registration during the data retrieval process could contribute to missing data. Incomplete data introduces biases due to discrepancies between observed and unobserved data. In time series prediction, it is

Table 1: Internal hybrid structure of deep learning models.

HDL model	Structure of layer
CNN	conv1D layer (filters: 256, filter size: 2, Relu activation) + maxpooling1D (pooling size: 2, padding: same) + conv1D layer (filters: 128, filter size: 2, Relu activation) + maxpooling1D (polling size: 2, padding: same) + flatten layer + dense layer (neuron: 1, Relu activation) + dropout = 0.2 with fully connected layer (neurons = 1: linear activation)
CNN-BiLSTM	conv1D layer (filters: 256, filter size: 2, Relu activation) + maxpooling1D (pooling size: 2, padding: same) + conv1D layer (filters: 128, filter size: 2, Relu activation) + maxpooling1D (polling size: 2, padding: same) + flatten layer + BiLSTM layer (neurons: 128, Relu activation) + BiLSTM layer (neurons: 50, Relu activation) + dense layer (neuron: 1, linear activation)

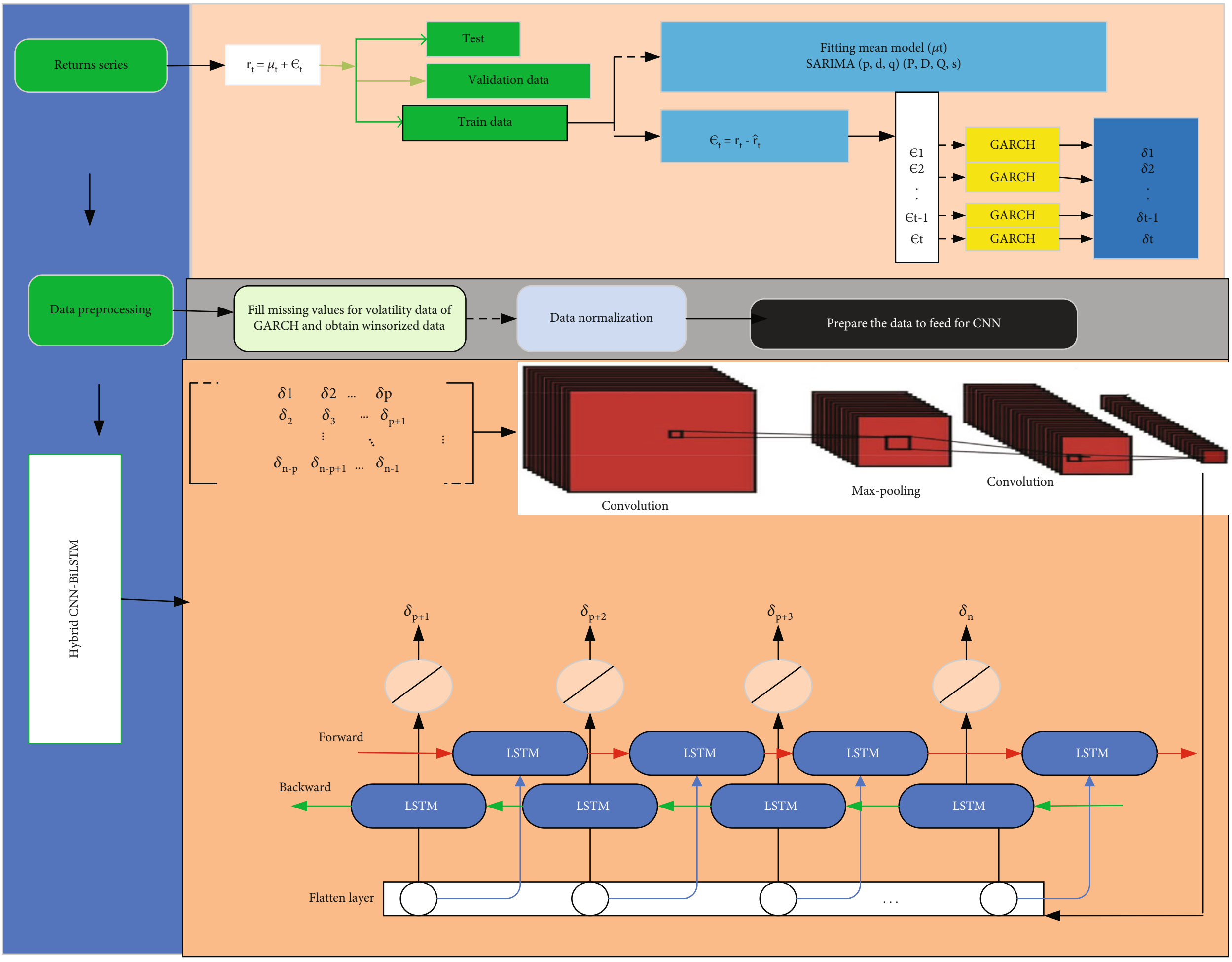


Figure 5: The structure of the hybrid SARIMA-GARCH-CNN-BiLSTM model.

essential to address this issue without discarding values from the series [32]. In this study, cubic spline interpolation was used to capture complex and nonlinear trend structures that linear methods cannot adequately handle. In cubic spline interpolation, the cubic spline polynomial is fit to time series data to estimate the missing values.

The mathematical formulation of the cubic spline method was used under the methodology outlined by Erdogan [33].

$$S(x) = \begin{cases} s_1(x) & \text{if } x_1^2 \le x < x_2 \\ s_2(x) & \text{if } x_2^2 \le x < x_3 \\ \vdots & \\ s_{n-1}(x) & \text{if } x_{n-1}^2 \le x < x_n \end{cases} \tag{14}$$

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

Equation (14) combines the piecewise function and the polynomial definition with a single equation number.

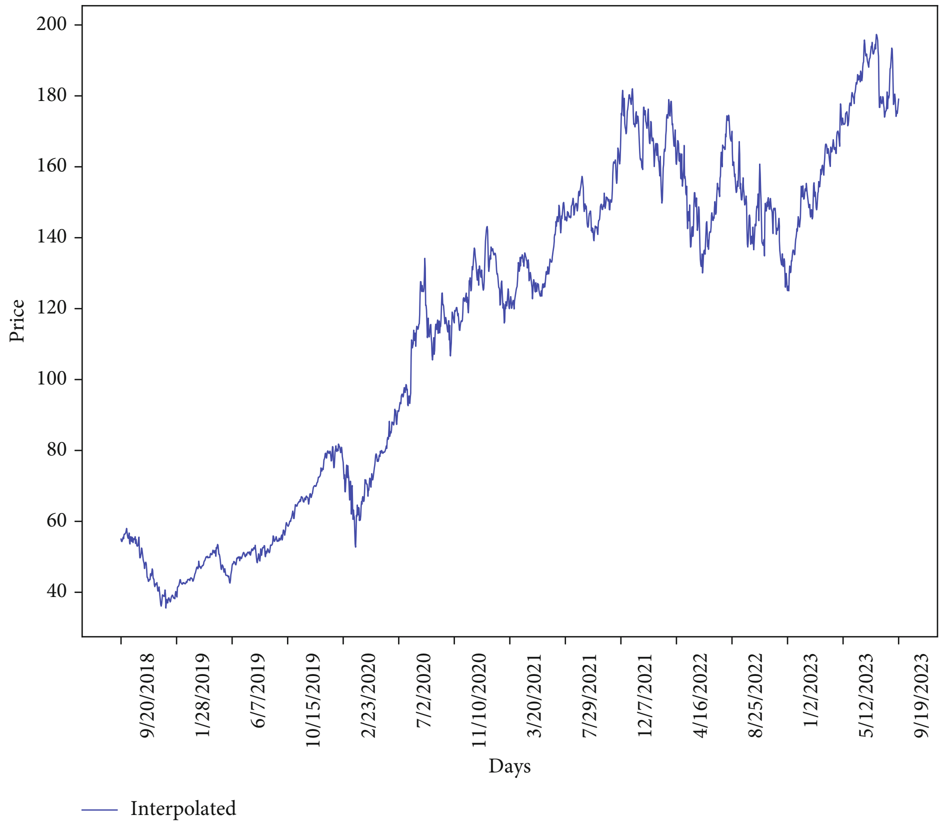


Figure 6: Time series plot of Apple Inc. price across time.

2. Outlier detection. Data points that are located at the outermost limits of a dataset are referred to be outliers.

3. Data normalization. The winsorized data, devoid of outliers, originates from various forecasting points, each exhibiting different scales of values. To address this issue and enhance the training efficiency of the model, min–max normalization, as represented in Equation (15), is employed to scale the dataset within the range of $(0, 1]$.

$$x_i = \frac{x_w - \min(x_w)}{\max(x_w) - \min(x_w)} \tag{15}$$

where x_i is the normalized value, x_w is the original winsorized value, $\max(x_w)$ is the maximum value of winsorized data x_w, and $\min(x_w)$ is the minimum value of winsorized data value x_w.

4. Data splitting. To create and assess a forecasting model, the normalized data needs to be divided. In practice, there has not been a universally perfect percentage for data splitting in the past. Nonetheless, there are various methods to partition the dataset. One instance involves an 80% training and 20% testing split [24, 34], while another approach entails allocating 70% to the training dataset, 15% to the validation dataset, and 15% to the test dataset [35]. Figure 4 presents the data splitting procedure utilized in this research, illustrating the allocation of datasets. In this study, the data is divided into three segments (60:20:20 ratio), facilitating a thorough model assessment. The training set educates the model about underlying patterns, while the validation set serves as an impartial benchmark for comparing algorithms trained on this data. Subsequently, the test set evaluates the model's performance on fresh data, ensuring a robust evaluation process and guarding against overfitting. Furthermore, to bolster model training and minimize overfitting, a $k = 3$-fold cross-validation technique is utilized within the training dataset.

Table 2: Summary statistics of Apple Inc. winsorized returns.

Statistic	Original returns	Winsorized returns
Mean	0.000648	0.000715
Median	0.000617	0.000617
Variance	0.000299	0.00017
Skewness	−0.128	−0.084
Kurtosis	5.52139	−0.3254
Jarque–Bera	2316.86	10.2145
p value	0.00	0.006
Sample size	1820	1820

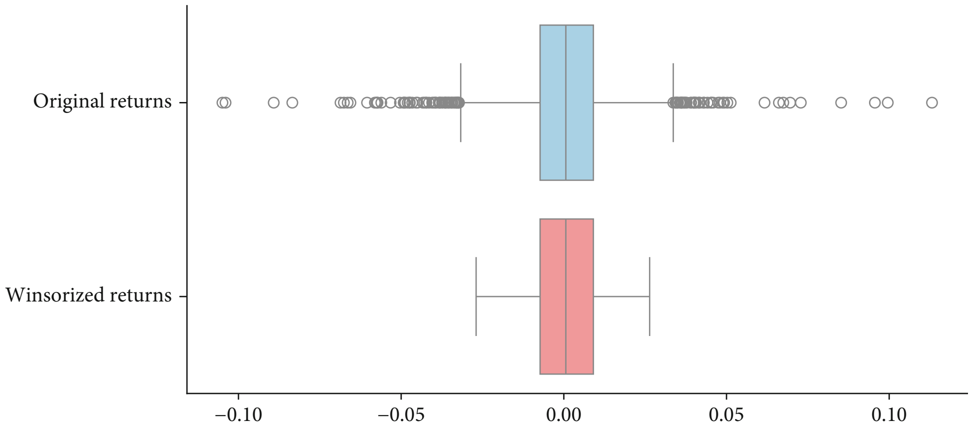


Figure 7: The box plot of returns and winsorized returns.

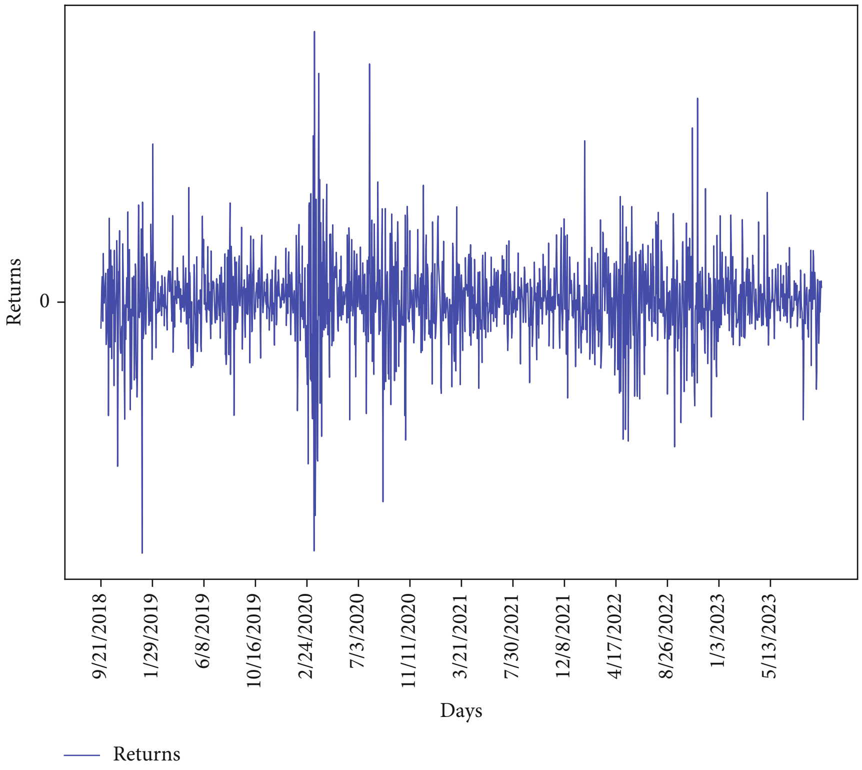


Figure 8: Apple Inc. price winsorized returns.

Table 3: ADF unit root test results.

	Daily Apple Inc. winsorized returns (r_t)		
Test	Statistic	Critical values	p value
ADF	−10.2166	% critical value: −3.4339 % critical value: −2.8631 % critical value: −2.5676	0.025

After splitting the data, it has to be reshaped into 3D to be an input for the hybrid CNN-BiLSTM model. Thus, input dimensions are samples, time steps, and features. The number of time steps (window size) was a hyperparameter that represents the number of previous lags used as input to predict the next time steps. The study used empirical testing to fix an optimal value of the window hyperparameter. The sequence of observations $\{x_1, x_2, \cdots, x_n\}$ have to be changed to multiple examples (samples) by developing a matrix X which served as the independent variable of the model and y as the dependent variable of the model of which the model can learn. Then, divide the time series to $n - p + 1$ examples where each sample has a size equal to the number of time steps (lagged variables) that is p and the size of learning samples is $n + 1$. The obtained size of the independent and

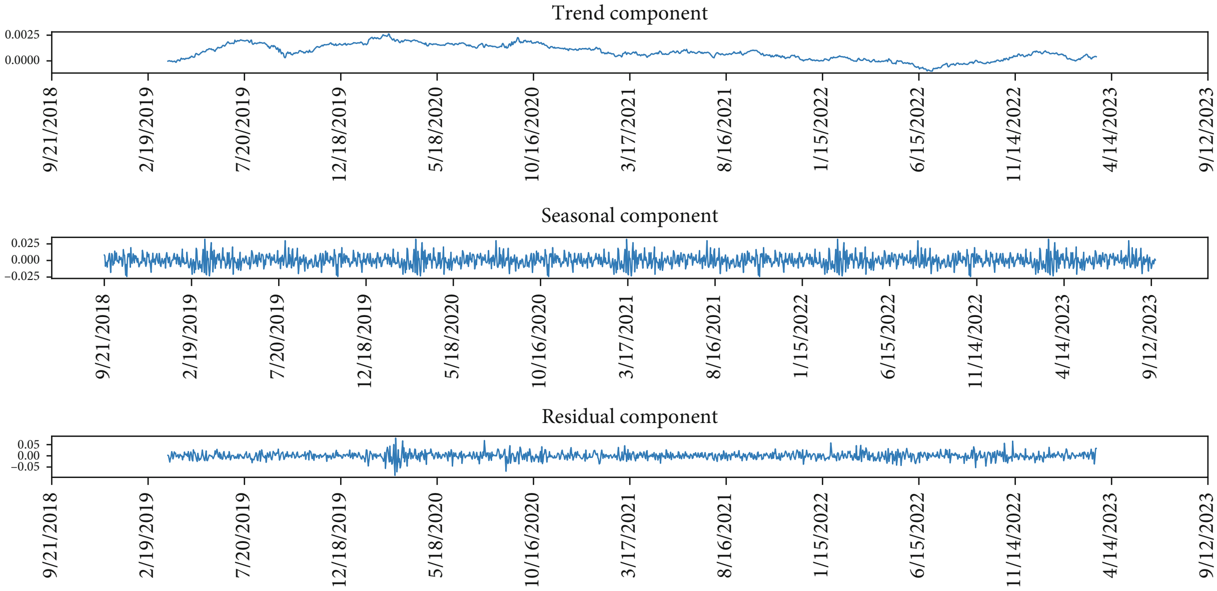


Figure 9: Additive decomposition of Apple Inc. winsorized return data.

predicted matrix will have a size $(n-p+1)\times p$ and $(n+1)\times 1$. The deep learning models described in this paper are outlined in Table 1. The goal is to pick out features from the input dataset using CNN layers. After that, the outputs of these CNN layers are passed into BiLSTM layers and a dense layer at the output to help predict sequences. The comprehensive architecture of the hybrid SARIMA-GARCH-CNN-BiLSTM model is detailed in Figure 5. This structure begins by leveraging the return series of Apple Inc.'s price data to establish the mean model. Subsequently, it analyzes the residuals of the mean model to estimate volatility using GARCH models. Finally, it preprocesses the GARCH model outputs to serve as inputs for the hybrid CNN-BiLSTM model.

2.8. Forecasting Performance Evaluation. The study evaluated forecast efficiency using two key metrics: mean absolute error (MAE) and root mean squared error (RMSE). MAE, less sensitive to outliers, measures average error magnitude, while RMSE, more sensitive to outliers, provides deeper insights into prediction performance. By utilizing both metrics, the study comprehensively assesses the model's predictive capabilities, considering data properties.

$$\mathrm{MAE} = \frac{1}{N}\sum_{k=1}^{N}\left|Y_k - \widehat{Y}_k\right| \tag{16}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(Y_k - \widehat{Y}_k\right)^2} \tag{17}$$

where N is the number of observations, Y_t are the actual values at time t, and $\widehat{Y}_k$ are the predicted values of the model at time t.

Table 4: Model comparison using the Akaike information criterion (AIC).

Model	AIC
ARIMA(1,0,0)(1,0,0)2	−5993.333
ARIMA(0,0,1)(0,0,1)2	−5993.063
ARIMA(1,0,0)(1,0,1)2	−5992.025
ARIMA(1,0,0)(0,0,1)2	−5992.182
ARIMA(1,0,1)(1,0,0)2	−5992.183
ARIMA(0,0,1)(1,0,0)2	−5993.920
ARIMA(0,0,1)(1,0,1)2	−5992.0

Table 5: Parameter estimation of the ARIMA(1,1,2)(0,0,0)2 model using the training set.

	Coef.	**Std. err.**	z	$P>\lvert z\rvert$	**0.025**	**0.975**
Intercept	0.0011	0.001	1.660	0.097	0.001	0.002
ma.L1	0.1998	0.017	12.014	0.001	0.167	0.232
ar.S.L2	−0.2096	0.018	−11.902	0.001	−0.244	−0.1375
Sigma2	0.0003	0.00659	46.658	0.001	0.000	0.000

Table 6: ARCH effect test for residuals of the SARIMA model.

	Lag	**Statistics**	***p* value**
For residual series			
ARCH-LM	Lag 1	99.0427	0.001
	Lag 21	153.3352	0.001
Ljung–Box	Lag 1	0.008505	0.926522
	Lag 21	8.949068	0.111113
For squared residual series			
Ljung–Box	Lag 1	100.864047	265.57485
	Lag 21	0.0096	0.00249

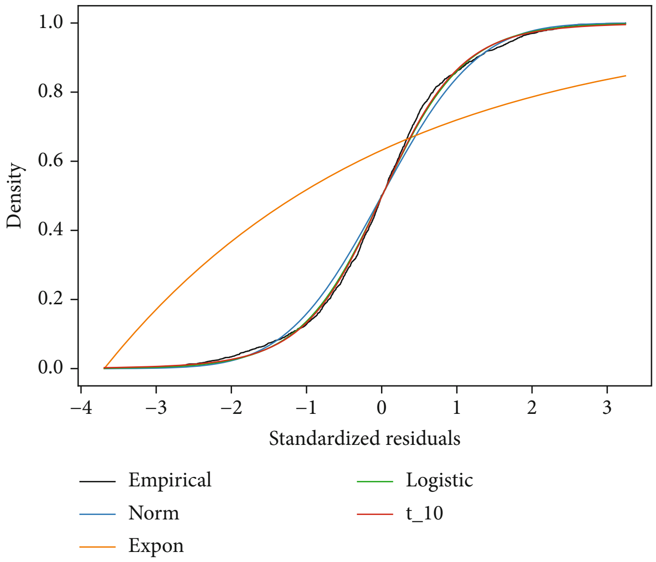


Figure 10: Comparison of standard residuals between distributions.

3. Data

This study used secondary data, namely, daily Apple Inc. close price data from 23 September 2018 to 19 September 2023, extracted from the Yahoo Finance website. There were a total of 1256 daily observations. The database had 565 missing data points. Thus, the author filled in values using cubic spline interpolation, as stated in Equation (14). The dataset was split into three portions: the training set, which accounted for 60% of the data; the validation set, which constituted 20% of the data to assess the performance of the trained model; and the remaining 20%, which was used for testing to evaluate the final performance of the model. Investors tend to focus more on price returns, which reflect price variation, rather than the price itself [36, 37], because return data is typically stationary, making it suitable for use in time series models. Thus, we chose to conduct our experiments using this variable. Additionally, the study scales the price returns to percentages to depict daily returns, as shown in Equation (18).

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right) \times 100 \tag{18}$$

where P_t is the daily closing price of Apple Inc. on day t, P_{t-1} is the daily closing price of Apple Inc. on the previous day, and r_t represents the daily returns of the Apple Inc. price index at time t. Since the returns exhibited a leptokurtic property, meaning excessive kurtosis compared to a Gaussian distribution (kurtosis = 3), the sensitivity to extreme fluctuations around the mean is pronounced. To identify outliers within the return series, the z-score test technique was employed. The z-score test is a statistical method used to evaluate the positioning of a data point in relation to the mean of a dataset, determining whether it falls within a specified range of values [38]. Upon identifying outliers within the dataset, winsorization was utilized as a remedial strategy, applying percentile thresholds for adjustment. Specifically, the study selected the 1st and 99th percentiles, substituting exceedingly low values with those corresponding to the 1st percentile and exceedingly high values with those associated with the 99th percentile.

4. Result and Discussion

The Apple Inc. price series in Figure 6 illustrates long-term increasing and decreasing trends. It suggests a nonstationarity series. Since the price series exhibits nonstationarity, the study applies a logarithmic transformation and takes the first difference at lag 1 to obtain the return series. Thus, when examining higher kurtosis in original return data, as shown in Table 2, it is desirable to winsorize data to make it less susceptible to outliers. Figure 7 displays the distribution of returns and winsorized returns through box plots, showcasing their respective central tendencies and dispersion. Due to the detection of outliers in the return data, the study opted to winsorize the dataset as shown in Figure 7. Table 2 shows the summary statistics of the returns and winsorized returns. The Jarque–Bera test rejects the normality assumption for both returns. With negative kurtosis, a lighter tail, and fewer outliers in the winsorized returns, the study utilized the depicted winsorized returns data in Table 2 for further analysis.

Figure 8 displays the winsorized graph that fluctuated above and below the 0 lines, indicating that the price series achieved stationarity in the mean. However, the series is not stationary in variance.

The study also tested for stationarity using the augmented Dickey–Fuller test. The ADF test result displayed in Table 3 rejects the null hypothesis of a unit root's existence, supported by the ADF test value of −10.2166 and a p value of 0.02576. It is safe to say that the winsorized daily returns are stationary.

The visual representation of the decomposition of the return time series is shown in Figure 9. This decomposition separates the series into three distinct components: trend,

Table 7: Anderson–Darling test results.

	W_n^2	p value	Best fit
Expo	∞	1.34	0
Norm	20.167945	0.784	0
Logistic	4.936475	0.66	1
t-student	90.8091	1.24	0

Table 8: AIC and BIC of SARIMA-SGARCH models.

Model	AIC	Log-likelihood
SARIMA(0,0,1)(1,1,0)2-SGARCH(1,1)	1008.52	−501.259
SARIMA(0,0,1)(1,1,0)2-SGARCH(1,2)	979.6836	−485.842
SARIMA(0,0,1)(0,1,1)2-SGARCH(2,1)	1010.5175	−501.255
SARIMA(0,0,1)(1,1,1)2-SGARCH(2,2)	981.6836	−485.842

seasonal, and residuals. It is evident from the figure that the observed series exhibits seasonality with a period of two.

The study employed the AIC for choosing the order, as it effectively minimizes prediction error and is asymptotically comparable to cross-validation [39]. Table 4 summarizes the AIC values of different models. After assessing various SARIMA models, the most suitable model for the observed series was ARIMA(0, 0, 1)(1, 0, 0)2, which exhibits the lowest AIC value of −5993.920.

The chosen model parameters are represented in Table 5, and the residual variance was significantly different from zero, suggesting that the model fits the data well. Using the estimates, the mean model SARIMA(0,0,1)(1,0,0,2) is written as

$$r_t = 0.0011 - 0.2096 r_{t-2} + 0.1998 \varepsilon_{t-1} + \varepsilon_t \tag{19}$$

At a significance level of 5%, Engle's Lagrange Multiplier ARCH test, employing up to 21 lags representing a 1-month trading period of Apple Inc., as presented in Table 6, rejects the null hypothesis of no ARCH effects in the winsorized returns. This means that the variances of returns were heteroscedastic and suggested the use of the ARCH/GARCH model for capturing the time-varying volatility in the returns. Furthermore, the result was supported by the Ljung–Box test, which indicated that the residuals were uncorrelated and that the squared residuals did not exhibit serial correlation. In simpler terms, the error variance exhibited autocorrelation. Next, a distribution for the residuals of the return series was estimated to approximate the empirical distribution. Based on the work conducted by Budiarti et al. [40], the study examined the distribution of residuals. A comparison was made among the standard residuals of three distributions, namely, normal distribution, t-student distribution, and logistic distribution, against the empirical distributions, as depicted in Figure 10.

The Anderson–Darling test showed that the most suitable distribution for Apple Inc. residual data was the logistic distribution, as shown in Table 7.

Table 9: AIC and BIC of SARIMA-TGARCH models.

Model	AIC	Log-likelihood
SARIMA(0,0,1)(1,1,0)2-TGARCH(1,1,1)	1010.28	−501.139
SARIMA(0,0,1)(1,1,0)2-TGARCH(1,1,2)	1012.28	−501.139
SARIMA(0,0,1)(0,1,1)2-TGARCH(1,2,1)	981.476	−485.476
SARIMA(0,0,1)(1,1,1)2-TGARCH(2,1,1)	1012.28	−501.139
SARIMA(0,0,1)(1,1,1)2-TGARCH(1,2,2)	983.476	−485.738
SARIMA(0,0,1)(1,1,1)2-TGARCH(2,1,2)	1014.28	−501.139
SARIMA(0,0,1)(1,1,1)2-TGARCH(2,2,1)	983.476	−485.738
SARIMA(0,0,1)(1,1,1)2-TGARCH(2,2,2)	985.476	−485.738

Table 10: AIC and BIC of SARIMA-EGARCH models.

Model	AIC	Log-likelihood
SARIMA(0,0,1)(1,1,0)2-EGARCH(1,1,1)	−617.779	−584.740
SARIMA(0,0,1)(1,1,0)2-EGARCH(1,1,2)	−730.704	372.352
SARIMA(0,0,1)(0,1,1)2-EGARCH(1,2,1)	−671.070	342.535
SARIMA(0,0,1)(1,1,1)2-EGARCH(2,1,1)	−730.284	691.738
SARIMA(0,0,1)(1,1,1)2-EGARCH(1,2,2)	−730.594	686.541
SARIMA(0,0,1)(1,1,1)2-EGARCH(2,1,2)	−728.706	−684.653
SARIMA(0,0,1)(1,1,1)2-EGARCH(2,2,1)	−687.262	351.631
SARIMA(0,0,1)(1,1,1)2-EGARCH(2,2,2)	−728.601	371.301

Table 11: Parameter estimation of GARCH family models using the training set.

Models	Model parameters	Coefficients	p values
SGARCH(1,2)	Intercept (ω)	0.0177	0.07805
	α_1	0.2337	0.02582
	β_1	0.3201	0.07668
	β_2	0.4431	0.01286
TGARCH(1,2,1)	Intercept (ω)	0.0177	0.07.602
	α_1	0.2338	0.02499
	β_1	0.2147	0.047391
	β_2	0.4223	0.01255
	γ_1	−0.2338	0.939
EGARCH(1,1,2)	Intercept (ω)	1.9742	0.01346
	α_1	2.5914	0.06552
	β_1	0.9946	0.001
	γ_1	−0.7159	0.619
	γ_2	−1.5419	0.02563

The study applied three variants of GARCH family models: SGARCH, EGARCH, and TGARCH, to the residuals of SARIMA for model identification. The AIC and

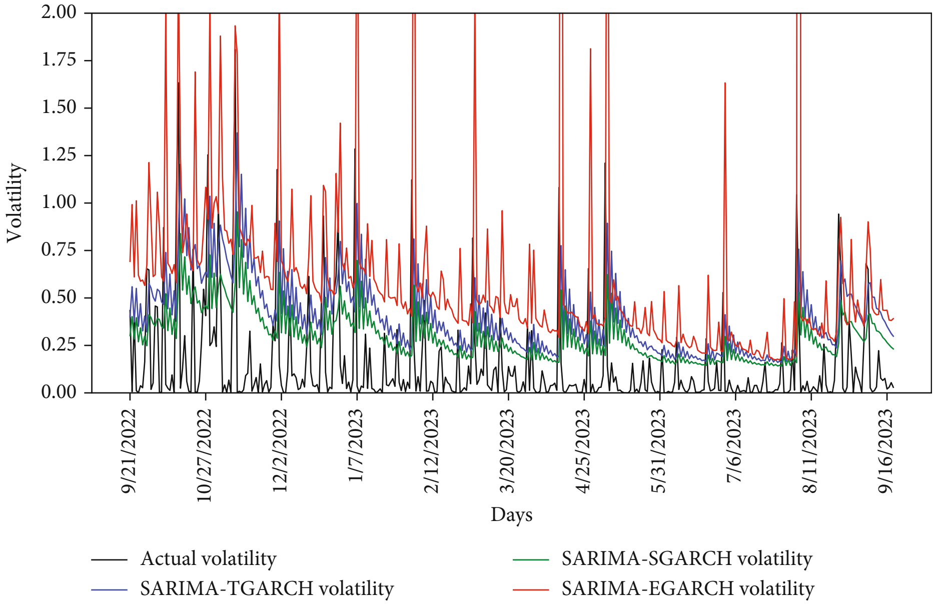


Figure 11: Volatility forecast of hybrid econometric models.

log-likelihood of the estimated models are noted for each of the models.

4.1. SARIMA-SGARCH Model Identification. The SARIMA(0,0,1)(0,0,1)2-SGARCH(1,2) model under the logistic distribution was favored over the other models due to its smallest AIC value (979.6836) and larger log-likelihood (−485.842), as indicated in Table 8.

4.2. SARIMA-TGARCH Model Identification. The SARIMA(0,0,1)(0,0,1)2-TGARCH(1,2,1) model under the logistic distribution was favored over the other models due to its smallest AIC value (981.476) and larger log-likelihood (−485.476), as illustrated in Table 9.

4.3. SARIMA-EGARCH Model Identification. The SARIMA(0,0,1)(0,0,1)2-EGARCH(1,1,2) model under the logistic distribution was favored over the other models due to its smallest AIC value (−730.704) and larger log-likelihood (372.352), as demonstrated in Table 10.

The mean and volatility models were developed as follows by utilizing Equation (19) and Table 11.

$$\begin{aligned}\text{Mean model}: \quad & r_t = 0.0011 - 0.2096 r_{t-2} + 0.1998 \varepsilon_{t-1} + \varepsilon_t \\ \text{Volatility model}: \quad & \sigma_t^2 = 0.0177 + 0.2337 \varepsilon_{t-1}^2 + 0.3201 \sigma_{t-1}^2 + 0.4431 \sigma_{t-1}^2\end{aligned} \tag{20}$$

Equation (20) shows the mean and volatility model of hybrid SARIMA(0,0,1)(1,0,0,2)-SGARCH(1,2). In the hybrid SARIMA(0,0,1)(1,0,0,2)-SGARCH(1,2) model, the degree of volatility persistence was found to be 0.9969. Given that the persistence parameter was very close to unity, volatility was predictable for future periods.

For the hybrid SARIMA(0,0,1)(1,0,0,2)-EGARCH(1,1,2) model, the mean equation was as described in Equation (20), while the volatility model was described as follows:

Table 12: Evaluation results of hybrid SARIMA-GARCH family models.

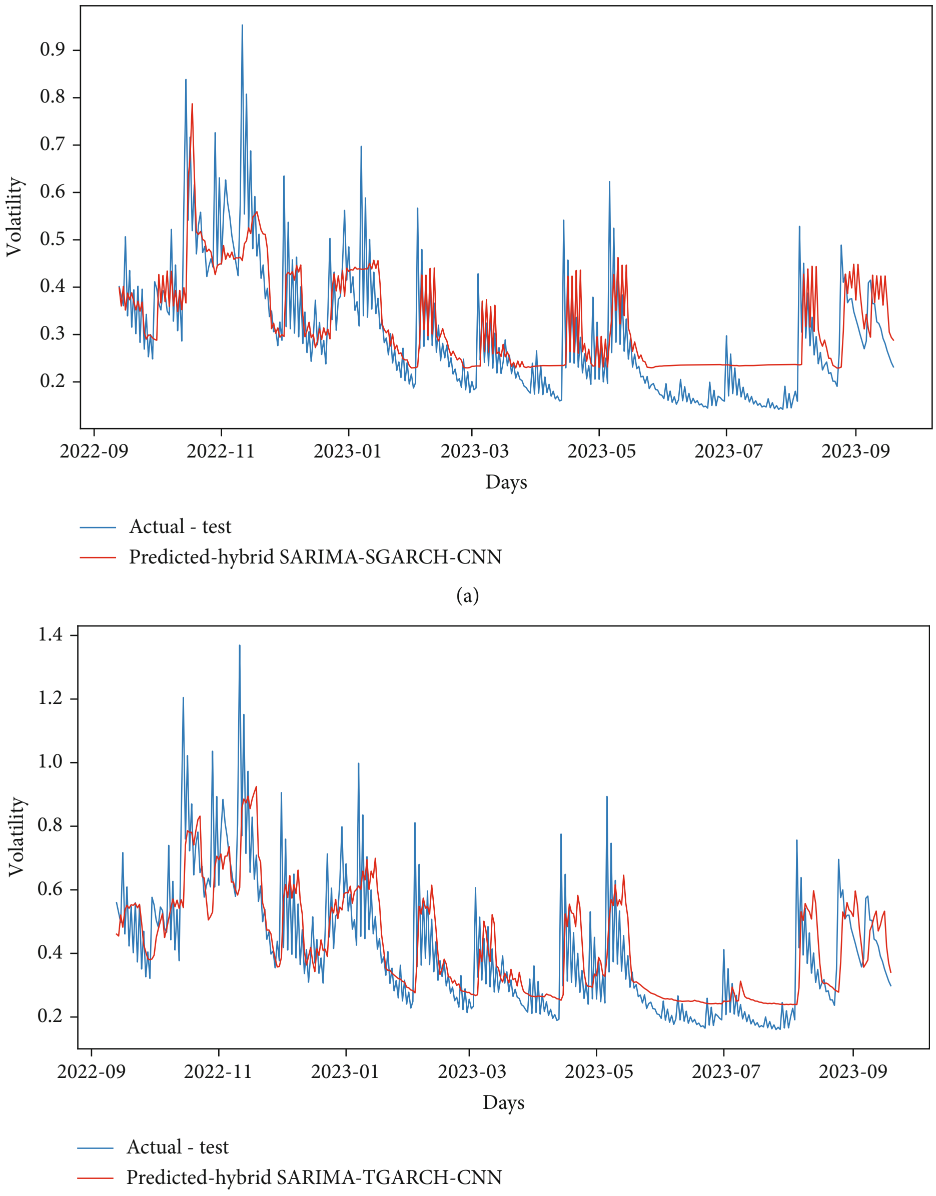
Model	MAE	RMSE
Hybrid SARIMA-SGARCH	0.2053	0.2731
Hybrid SARIMA-EGARCH	0.444	1.1706
Hybrid SARIMA-TGARCH	0.2601	0.3281

$$\ln\left(\sigma_t^2\right) = -1.9742 + 2.5914 \cdot \left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right| - 0.7159 \cdot \frac{\varepsilon_{t-1}}{\sigma_{t-1}} - 1.5419 \cdot \frac{\varepsilon_{t-2}}{\sigma_{t-2}} + 0.9946 \cdot \ln\left(\sigma_{t-1}^2\right) \tag{21}$$

Alexander [41] stated that volatility takes longer to diminish when β_j is relatively large. Therefore, based on Equation (21), the value of $\beta_1 = 0.9946$ indicated longer volatility persistence. Moreover, since both $\gamma_1 = -0.7159$ and $\gamma_2 = -1.5419$ were negative, it suggested that adverse news had a greater impact on the volatility of returns for Apple Inc. Finally, the hybrid SARIMA(0,0,1)(1,0,0,2)-TGARCH(1,2,1) model exhibited a similar mean model as described in Equation (20), but the volatility equation was formulated as follows:

$$\sigma_t^2 = 0.0177 + 0.2147 \varepsilon_{t-1}^2 - 0.2338 \varepsilon_{t-1}^2 \cdot d_{t-1} + 0.4223 \sigma_{t-1}^2 \tag{22}$$

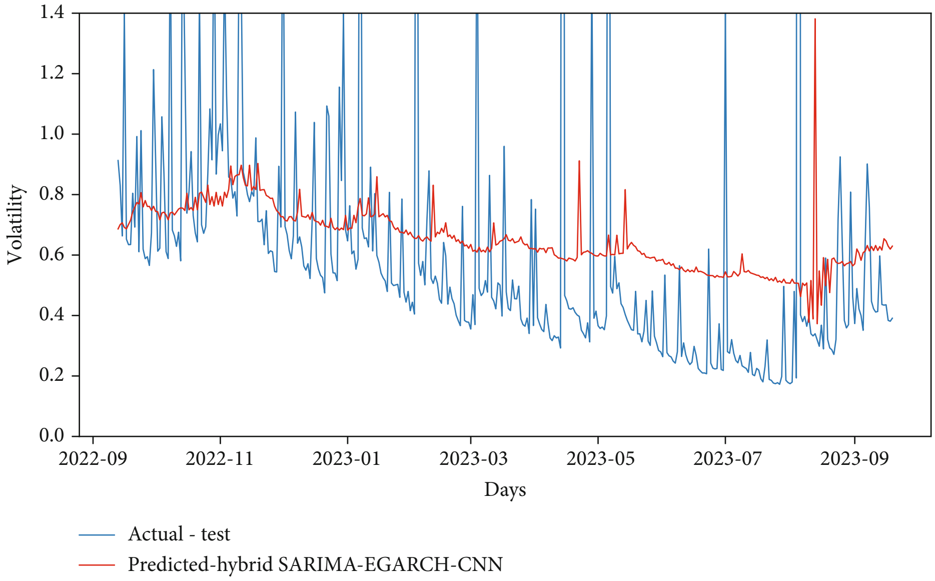
From Equation (22), $\gamma_1 = -0.2338 \neq 0$ indicates that there exists a leverage effect. In addition to this, $\gamma_1 < 0$ indicates that volatility increased more by positive shocks than negative shocks at equal length.



(a)

(b)

Figure 12: Continued.



(c)

Figure 12: Volatility forecasts for hybrid econometrics with CNN models. (a) Volatility forecast of hybrid SARIMA-SGARCH-CNN, (b) volatility forecast of hybrid SARIMA-TGARCH-CNN, and (c) volatility forecast of hybrid SARIMA-EGARCH-CNN.

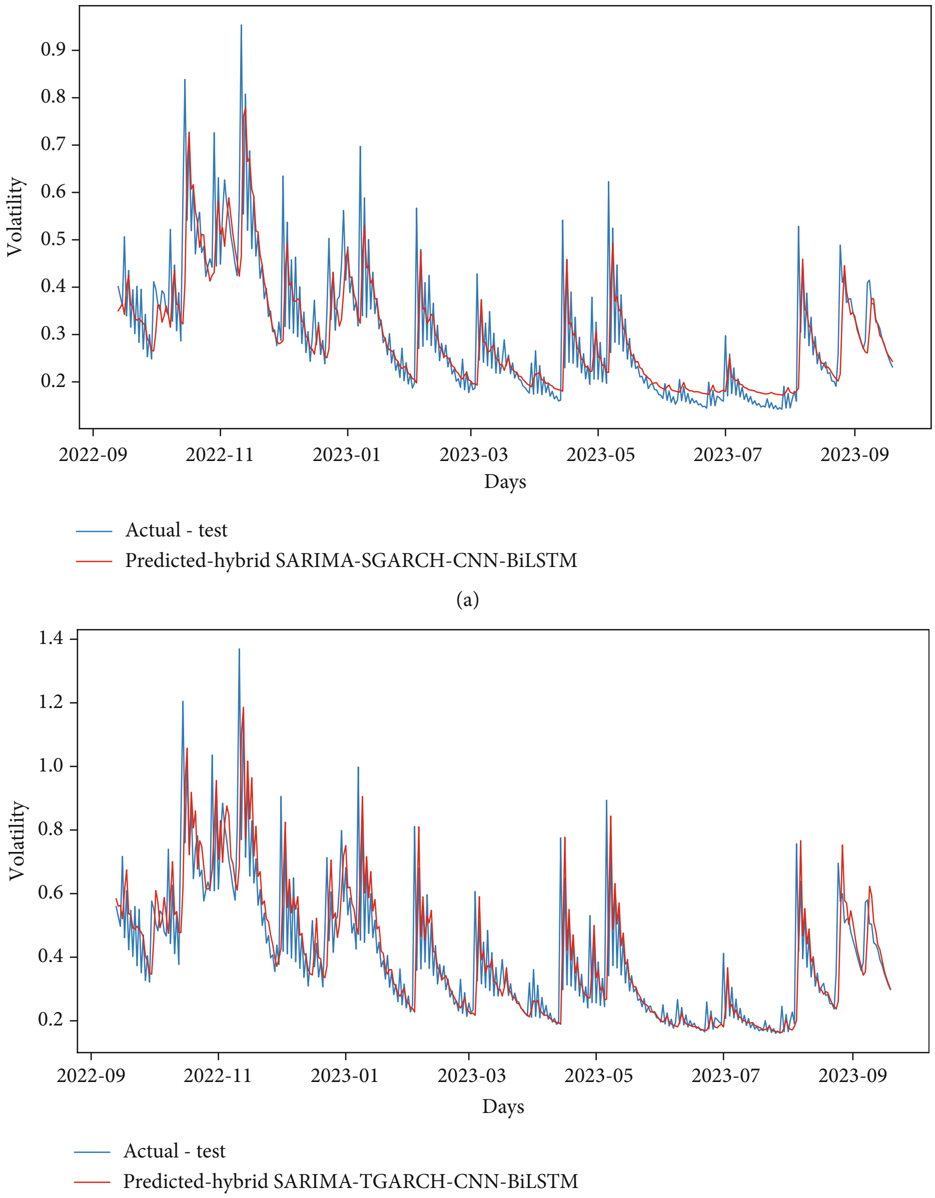
4.4. The Prediction Results of Hybrid Econometric Models. As shown in Figure 11, it is the fitting curve of the predicted value and the true value of every hybrid econometric model. It can be seen that the fitting effect of the hybrid SARIMA-SGARCH model was better than the other two models. As shown in Table 12, it is the value of the evaluation index of hybrid econometric models. Thereinto, the value of MAE (0.2053) and RMSE (0.2731) of hybrid SARIMA-SGARCH is the smallest. Therefore, the overall performance of the hybrid SARIMA-SGARCH model is better. However, these hybrid econometric models did not show good prediction performance because they did not capture complex nonlinear relationships and sequential dependencies of the data.

4.5. Prediction Results of Hybrid Econometric Models With CNN Model. Through the above prediction, it is found that the hybrid SARIMA-SGARCH model has a better fitting effect and higher prediction accuracy than the prediction results of the other two econometric models. Furthermore, to improve the prediction accuracy of volatility, rely on (1) hybrid econometric models to describe the series pattern and volatility clustering and (2) the CNN model to capture spatial patterns and local dependencies in the data, enhancing model robustness and accuracy. Through this experiment, the study used the estimated volatilities of hybrid econometric models as input features for the CNN model. The prediction results of hybrid econometrics with CNN models are shown in Figures 12(a), 12(b), and 12(c), while the corresponding evaluation outcomes are summarized in Table 13. Through the prediction results and evaluation results, it can be found that, compared with hybrid SARIMA-SGARCH, the MAE and RMSE of hybrid SARIMA-SGARCH-CNN decreased by 65% and 66.7%, respectively. The MAE and RMSE of the hybrid SARIMA-EGARCH-CNN increased by 36.3% and 75.4%, respectively. The MAE and RMSE of the hybrid SARIMA-TGARCH-CNN decreased by 51.5% and 54.5%, respectively. The MAE and RMSE values for hybrid SARIMA-SGARCH-CNN are the lowest among the models considered, indicating its superior overall performance. However, despite these favorable metrics, the hybrid SARIMA-SGARCH-CNN model exhibited suboptimal prediction performance. This limitation could stem from its inability to effectively capture sequential patterns and long-term dependencies within the data.

Table 13: Evaluation results of hybrid econometrics with CNN models.

Model	MAE	RMSE
Hybrid SARIMA-SGARCH-CNN	0.0717	0.0909
Hybrid SARIMA-EGARCH-CNN	0.3223	1.1120
Hybrid SARIMA-TGARCH-CNN	0.099510	0.12419

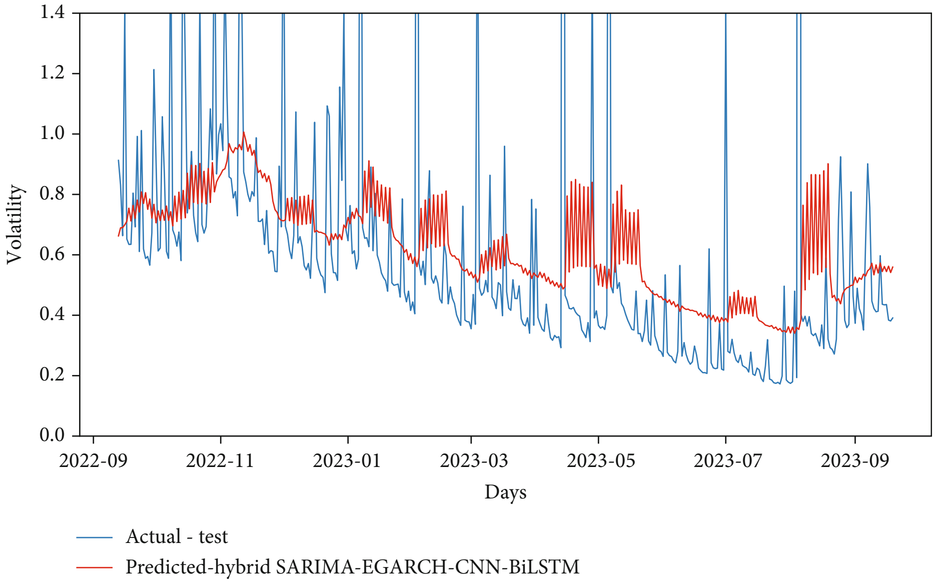
4.6. Prediction Results of the Proposed Hybrid Econometrics With Hybrid CNN-BiLSTM Model. The study introduced three novel hybrid models, denoted as hybrid SARIMA-SGARCH-CNN-BiLSTM, hybrid SARIMA-EGARCH-CNN-BiLSTM, and hybrid SARIMA-TGARCH-CNN-BiLSTM, and conducted comparative analyses with existing hybrid econometric models and hybrid econometric models incorporating CNN. Additionally, the study performed intercomparisons among the newly proposed hybrid models to identify the superior-performing model. Now, the new models are constructed by taking the forecasted volatility of hybrid econometric models as input features to the hybrid



(a)

(b)

Figure 13: Continued.



(c)

Figure 13: Volatility forecasts for a hybrid of econometrics with CNN-BiLSTM models. (a) Volatility forecast of hybrid SARIMA-SGARCH-CNN-BiLSTM, (b) volatility forecast of hybrid SARIMA-TGARCH-CNN-BiLSTM, and (c) volatility forecast of hybrid SARIMA-EGARCH-CNN-BiLSTM.

Table 14: Evaluation results of hybrid econometrics with hybrid CNN-BiLSTM models.

Model	MAE	RMSE
Hybrid SARIMA-SGARCH-CNN-BiLSTM	0.0514	0.0706
Hybrid SARIMA-EGARCH-CNN-BiLSTM	0.2840	1.1110
Hybrid SARIMA-TGARCH-CNN-BiLSTM	0.07301	0.11864

CNN-BiLSTM model to capture dependencies from both past and future contexts. The prediction results of the newly proposed models are shown in Figures 13(a), 13(b), and 13(c). The evaluation results of the model are presented in Table 14. In comparison to hybrid SARIMA-SGARCH-CNN, the hybrid model hybrid SARIMA-SGARCH-CNN-BiLSTM exhibited notable improvements, with a decrease of 28.3% in MAE and 22.3% in RMSE. Conversely, hybrid SARIMA-EGARCH-CNN-BiLSTM experienced an increase of 74.7% in MAE and 91.8% in RMSE compared to hybrid SARIMA-SGARCH-CNN. Similarly, hybrid SARIMA-TGARCH-CNN-BiLSTM showed a slight increase of 1.7% in MAE and 23.3% in RMSE relative to hybrid SARIMA-SGARCH-CNN. Moreover, hybrid SARIMA-SGARCH-CNN-BiLSTM demonstrated superior performance compared to hybrid SARIMA-EGARCH-CNN-BiLSTM, with a significant decrease of 81.9% in MAE and 93.6% in RMSE. Additionally, hybrid SARIMA-SGARCH-CNN-BiLSTM outperformed hybrid SARIMA-TGARCH-CNN-BiLSTM, with reductions of 29.5% in MAE and 40.5% in RMSE. Similarly, hybrid SARIMA-TGARCH-CNN-BiLSTM exhibited improvements over hybrid SARIMA-EGARCH-CNN-BiLSTM, with reductions of 58.2% in MAE and 89.3% in RMSE. Notably, among the newly proposed hybrid models, hybrid SARIMA-SGARCH-CNN-BiLSTM demonstrated superior predictive accuracy compared to all hybrid econometric models with CNN-BiLSTM, indicating its efficacy in forecasting. Consequently, the hybrid SARIMA-SGARCH-CNN-BiLSTM model stands out as the most effective in achieving optimal prediction performance and the highest accuracy among the proposed models.

Based on the obtained results, the hybrid SARIMA-SGARCH-CNN-BiLSTM model outperformed other hybrid econometric models, showing an average reduction in MAE and RMSE of 81.9% and 82.25%, respectively. Additionally, the hybrid SARIMA-SGARCH-CNN-BiLSTM model exhibited superior performance compared to hybrid econometrics with CNN models, with an average reduction in MAE and RMSE of 53.5% and 52.8%, respectively. Moreover, compared to hybrid SARIMA-GARCH family models, the outperformed hybrid of econometrics with CNN-BiLSTM demonstrates an average reduction in MAE and RMSE of 55.5% and 67%, respectively. The outcomes indicate that the hybrid SARIMA-SGARCH-CNN-BiLSTM model outperformed its counterparts, the hybrid econometrics with CNN-BiLSTM models, exhibiting notable improvements. This model showcased an average reduction in MAE and RMSE of 55.5% and 67%, respectively, underscoring its enhanced predictive accuracy and efficiency. Compared to three distinct hybrid models, the proposed hybrid SARIMA-STANDARD-GARCH-CNN-BiLSTM model demonstrates an average reduction in MAE and RMSE of 60.35% and 60.6%, respectively. Overall, the hybrid SARIMA-SGARCH-CNN-BiLSTM model emerged as the top performer among the various hybrid econometric models evaluated in this study. These findings underscore the effectiveness of the SARIMA-SGARCH-CNN-BiLSTM approach in volatility forecasting.

5. Conclusion

Based on the findings presented in the study, the following conclusions were drawn. The study used the input as residuals of the SARIMA model for three hybrid econometric models and selected the best model. The results showed that the hybrid SARIMA-SGARCH model outperformed the other two hybrid models. Additionally, the study incorporated the estimated volatilities of the econometric models as input features to the CNN to enhance prediction accuracy. The results indicated that the hybrid SARIMA-SGARCH-CNN model outperformed the hybrid SARIMA-EGARCH-CNN and hybrid SARIMA-TGARCH-CNN models. Finally, the study constructed three new models and used the estimated volatility of the hybrid econometrics as input to the CNN-BiLSTM model, concluding that the hybrid SARIMA-SGARCH-CNN-BiLSTM model performs well. This proposed model demonstrated its effectiveness, particularly with Apple Inc. data, providing valuable insights for financial data analysis and risk management strategies, thus aiding investors in making informed decisions.

6. Recommendation

The study was limited to investigating the volatility prediction of Apple Inc. using the three new hybrids of econometrics with CNN-BiLSTM models. As a result, the following aspects for further investigation are suggested. Future studies should explore incorporating attention mechanisms or transformer architectures to capture complex temporal dependencies and improve forecasting accuracy. Additionally, evaluating the impact of incorporating additional data sources beyond historical stock prices, such as sentiment analysis and financial news, could augment the forecast model. Furthermore, extending the research to include volatility forecasting for a portfolio of assets and exploring correlations between the assets is recommended.

Data Availability Statement

The data of this study can be obtained by contacting the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This work is supported by the Pan African University Institute for Basic Sciences, Technology and Innovation.

Acknowledgments

This work is supported by the Pan African University Institute for Basic Sciences, Technology and Innovation.

References

[1] E. F. Fama, “The behavior of stock-market prices,” *The Journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.

[2] T. H. Roh, “Forecasting the volatility of stock price index,” *Expert Systems with Applications*, vol. 33, no. 4, pp. 916–922, 2007.

[3] D. Bhowmik, “Stock market volatility: an evaluation,” *International Journal of Scientific and Research Publications*, vol. 3, no. 10, pp. 1–17, 2013.

[4] R. F. Engle, “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation,” *Econometrica: Journal of the Econometric Society*, vol. 50, no. 4, pp. 987–1007, 1982.

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

[6] R. Khaldi, A. El Afia, and R. Chiheb, “Forecasting of BTC volatility: comparative study between parametric and nonparametric models,” *Progress in Artificial Intelligence*, vol. 8, no. 4, pp. 511–523, 2019.

[7] D. B. Nelson, “Conditional heteroskedasticity in asset returns: a new approach,” *Econometrica: Journal of the Econometric Society*, vol. 59, no. 2, pp. 347–370, 1991.

[8] J.-M. Zakoian, “Threshold heteroskedastic models,” *Journal of Economic Dynamics and Control*, vol. 18, no. 5, pp. 931–955, 1994.

[9] B. Y. Almansour, M. M. Alshater, and A. Y. Almansour, “Performance of ARCH and GARCH models in forecasting cryptocurrency market volatility,” *Industrial Engineering & Management Systems*, vol. 20, no. 2, pp. 130–139, 2021.

[10] P. H. Franses and D. Van Dijk, “Forecasting stock market volatility using (non-linear) GARCH models,” *Journal of Forecasting*, vol. 15, no. 3, pp. 229–235, 1996.

[11] J. Sen, S. Mehtab, and A. Dutta, “Volatility modeling of stocks from selected sectors of the Indian economy using GARCH,” in *2021 Asian Conference on Innovation in Technology (ASIANCON)*, PUNE, India, 2021.

[12] J. Aduda, P. Weke, P. Ngare, and J. Mwaniki, “Financial time series modelling of trends and patterns in the energy markets,” *Journal of Mathematical Finance*, vol. 6, no. 2, pp. 324–337, 2016.

[13] Y. Liu, “Novel volatility forecasting using deep learning–long short term memory recurrent neural networks,” *Expert Systems with Applications*, vol. 132, pp. 99–109, 2019.

[14] H. Y. Kim and C. H. Won, “Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models,” *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[15] K. Kakade, A. K. Mishra, K. Ghate, and S. Gupta, “Forecasting commodity market returns volatility: a hybrid ensemble learning GARCH-LSTM based approach,” *Intelligent Systems in Accounting, Finance and Management*, vol. 29, no. 2, pp. 103–117, 2022.

[16] A. Vidal and W. Kristjanpoller, “Gold volatility prediction using a CNN-LSTM approach,” *Expert Systems with Applications*, vol. 157, article 113481, 2020.

[17] H. Zeng, B. Shao, G. Bian, H. Dai, and F. Zhou, “A hybrid deep learning approach by integrating extreme gradient boosting-long short-term memory with generalized autoregressive conditional heteroscedasticity family models for natural gas load

volatility prediction," *Energy Science & Engineering*, vol. 10, no. 7, pp. 1998–2021, 2022.

[18] D. K. Mademlis and N. Dritsakis, "Volatility forecasting using hybrid GARCH neural network models: the case of the Italian stock market," *International Journal of Economics and Financial Issues*, vol. 11, no. 1, pp. 49–60, 2021.

[19] F. H. Mustapa and M. T. Ismail, "Modelling and forecasting S&P 500 stock prices using hybrid Arima-Garch model," *Journal of Physics: Conference Series*, vol. 1366, no. 1, article 012130, 2019.

[20] L. R. Glosten, R. Jagannathan, and D. 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, pp. 1779–1801, 1993.

[21] R. Rabemananjara and J.-M. Zakoian, "Threshold ARCH models and asymmetries in volatility," *Journal of Applied Econometrics*, vol. 8, no. 1, pp. 31–49, 1993.

[22] S.-H. Poon, *A Practical Guide to Forecasting Financial Market Volatility*, John Wiley & Sons, 2005.

[23] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[24] F. Berzal, *Redes Neuronales & Deep Learning: Volumen II*, Independently published, 2019.

[25] M. Vakalopoulou, S. Christodoulidis, N. Burgos, O. Colliot, and V. Lepetit, "Deep learning: basics and convolutional neural networks (CNNs)," in *Machine Learning for Brain Disorders*, O. Colliot, Ed., vol. 197 of Neuromethods, Humana, New York, NY, 2023.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.

[27] J. Cao and J. Wang, "Stock price forecasting model based on modified convolution neural network and financial time series analysis," *International Journal of Communication Systems*, vol. 32, no. 12, article e3987, 2019.

[28] L. Muhammad, A. A. Haruna, U. S. Sharif, and M. B. Mohammed, "CNN-LSTM deep learning based forecasting model for Covid-19 infection cases in Nigeria, South Africa and Botswana," *Health and Technology*, vol. 12, no. 6, pp. 1259–1276, 2022.

[29] J. Rala Cordeiro, A. Raimundo, O. Postolache, and P. Sebastião, "Neural architecture search for 1D CNNs—different approaches tests and measurements," *Sensors*, vol. 21, no. 23, p. 7990, 2021.

[30] S. Y. Yerima, M. K. Alzaylaee, and A. P. V. Shajan, "Deep learning techniques for android botnet detection," *Electronics*, vol. 10, no. 4, p. 519, 2021.

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

[32] M. Noor, A. Yahaya, N. A. Ramli, and A. M. Al Bakri, "Filling missing data using interpolation methods: study on the effect of fitting distribution," *Key Engineering Materials*, vol. 594, pp. 889–895, 2014.

[33] K. Erdogan, "Spline interpolation techniques," *Journal of Technical Science and Technologies*, vol. 2, no. 1, pp. 47–52, 2013.

[34] G. Memarzadeh and F. Keynia, "A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets," *Energy Conversion and Management*, vol. 213, article 112824, 2020.

[35] F. Aksan, Y. Li, V. Suresh, and P. Janik, "CNN-LSTM vs. LSTM-CNN to predict power flow direction: a case study of the high-voltage subnet of Northeast Germany," *Sensors*, vol. 23, no. 2, p. 901, 2023.

[36] E. Hajizadeh, A. Seifi, M. F. Zarandi, and I. Turksen, "A hybrid modeling approach for forecasting the volatility of S&P 500 index return," *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012.

[37] M. Seo and G. Kim, "Hybrid forecasting models based on the neural networks for the volatility of bitcoin," *Applied Sciences*, vol. 10, no. 14, p. 4768, 2020.

[38] D. S. Moore and G. P. McCabe, *Introduction to the Practice of Statistics*, WH Freeman/Times Books/Henry Holt & Co., 1989.

[39] M. Stone, "An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 44–47, 1977.

[40] R. Budiarti, K. Intansari, I. G. P. Purnaba, and F. Septyanto, "Modelling dependencies of stock indices during Covid-19 pandemic by extreme-value copula," *Jurnal Teori dan Aplikasi Matematika*, vol. 7, no. 3, pp. 805–819, 2023.

[41] C. Alexander, *Market Risk Analysis*, John Wiley & Sons, Boxset, 2009.