

Article

Forecasting Bitcoin Volatility Using Hybrid GARCH Models with Machine Learning

Mamoona Zahid ¹, Farhat Iqbal ¹ and Dimitrios Koutmos ^{2,*}¹ Department of Statistics, University of Balochistan, Quetta 87300, Pakistan² Department of Accounting, Finance, and Business Law, College of Business, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA

* Correspondence: dimitrios.koutmos@tamucc.edu

Abstract: The time series movements of Bitcoin prices are commonly characterized as highly nonlinear and volatile in nature across economic periods, when compared to the characteristics of traditional asset classes, such as equities and commodities. From a risk management perspective, such behaviors pose challenges, given the difficulty in quantifying and modeling Bitcoin's price volatility. In this study, we propose hybrid analytical techniques that combine the strengths of the non-stationary properties of Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models with the nonlinear modeling capabilities of deep learning algorithms, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM (BiLSTM) algorithms with single, double, and triple layer network architectures to forecast Bitcoin's realized price volatility. Our findings, both in-sample and out-of-sample, show that such hybrid models can generate accurate forecasts of Bitcoin's price volatility.

Keywords: volatility; Bitcoin; machine learning; GARCH; recurrent neural networks

Citation: Zahid, Mamoona, Farhat Iqbal, and Dimitrios Koutmos. 2022. Forecasting Bitcoin Volatility Using Hybrid GARCH Models with Machine Learning. *Risks* 10: 237. <https://doi.org/10.3390/risks10120237>

Academic Editor: Mogens Steffensen

Received: 12 October 2022

Accepted: 2 December 2022

Published: 13 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid development of technology has spurred changes in the structure of financial markets and payment systems. In recent years, global financial markets have become increasingly digital, supporting cashless payment systems across our society. Present-day technology allows people to invest and generate their own money through virtual currency, known as cryptocurrency (Pabuçcu et al. 2020).

Bitcoin is the most popular and commonly utilized among the cryptocurrencies available on the crypto market. Bitcoin's market capitalization has exceeded USD 600 billion as of January 2021 (Aysan et al. 2021). Since Bitcoin's inception in 2009, its price has experienced volatility, unlike anything we have witnessed from traditional asset classes, such as equities and commodities (Pabuçcu et al. 2020).

These irregular Bitcoin price spikes are a common feature of the extreme volatility that is regularly witnessed in cryptocurrency markets (Gbadebo et al. 2021). This heightened volatility in cryptocurrency markets also indicates that traders might quickly make or lose a significant amount of money as investor sentiment and risk aversions change (Koutmos 2022). This is one of the reasons why much of the research has focused on Bitcoin's price volatility and the degree to which price movements can be exploited (Wellenreuther and Voelzke 2019; Al-Yahyaee et al. 2019). In addition, there is a growing interest in how cryptocurrencies' unique microstructures, such as their coin mining methods, can play a role in their price behaviors (Bowden et al. 2021; King et al. 2021).

The Generalized Autoregressive Conditional Heteroscedastic (GARCH) model has been widely used in financial economic research to model asset price volatility. Many extensions of the plain vanilla GARCH model have been proposed by researchers to better capture the stylized facts of the returns of a variety of assets and markets (Lim and Sek 2013;

Koutmos 2015; Kyriazis 2021). Bouoiyour and Selmi (2016) examined the volatility dynamics of Bitcoin using several types of asymmetric GARCH models and concluded that the Component with Multiple Threshold (CMT-GARCH) model properly reflects the dynamic aspects of Bitcoin price volatility. Katsiampa (2017) argues that the Autoregressive-Component GARCH (AR-CGARCH) best fits the GARCH family models for modeling Bitcoin volatility. Similarly, Chu et al. (2017) concluded that the IGARCH (1,1) model provides a good fit for the volatilities of cryptocurrencies. Conrad et al. (2018) found that the GARCH-MIDAS (Mixed Data Sampling) model improves Bitcoin's long-term volatility modeling. Baur and Dimpfl (2018) investigated the asymmetric volatility characteristic in different cryptocurrencies by applying the threshold GARCH (TGARCH) model. Charles and Darné (2019) studied GARCH-type models to assess the volatility of Bitcoin by considering different stylized facts. Similarly, Gyamerah (2019) discovered that asymmetric GARCH models with long-memory and heavy-tailed error distributions accomplish better volatility forecasts for Bitcoin as well as other cryptocurrencies. Zahid et al. (2022) applied Realized HA-GARCH-type models with jumps and inverse leverage effect to model and forecast the realized volatility of Bitcoin.

The successful development of machine learning (ML) techniques in time series has encouraged analysts to apply these techniques for modeling the dynamics of financial markets. This has also motivated researchers to apply ML models for predicting cryptocurrencies (Shen et al. 2021). Unlike traditional models, ML approaches do not require strict assumptions. While conventional time series and econometrics models look at the whole data, ML models split the data into training and testing datasets. These models aim to increase the accuracy by decreasing predefined loss functions (Butner et al. 2019; Makridakis et al. 2018). The ML models have generally performed better than traditional models when specially developed to deal with the particular problems of big datasets (Pabuçcu et al. 2020).

Recently, many researchers have introduced hybrid methods by combining ML methods, such as Artificial Neural Networks (ANN) and Support Vector Regression (SVR), with GARCH-type models to improve the volatility forecasts of cryptocurrencies. For instance, Kristjanpoller and Minutolo (2018) demonstrated that an ANN-GARCH model improves the volatility of Bitcoin prices. On the other hand, Peng et al. (2018) found that the SVR-GARCH model performs better and outperforms asymmetric GARCH models. Seo and Kim (2020) emphasized that the GARCH hybrid with Higher Order Neural Network (HONN) provided more accurate forecasts than the GARCH-ANN model.

In recent years, experts from various financial sectors have shifted their focus away from machine learning and onto its more sophisticated form, deep learning (DL). While ML algorithms, such as feed-forward neural networks, excel in forecasting nonlinear series, they are limited in understanding temporal dependencies within the data. Recurrent Neural Networks (RNNs) are a different class of neural networks suited for modeling time series problems because they can learn the temporal correlations between sequential and time series data. DL approaches are regarded as extremely effective in recognizing the financial market's chaotic characteristics (Kamnitsas et al. 2017; Lahmiri and Bekiros 2019). Explanatory factors are input information to improve the neural network models' volatility estimates.

Hinton and Salakhutdinov (2006) proposed the DL method and developed its various extensions to deal with classification and regression problems. These methods reflect the human brain processes information and strengthen the ANN by using a series of hidden layers, improving models' forecasting ability. The DL algorithm is also modified into different architectures, providing a novel perspective on financial time series (Feuerriegel and Fehrer 2016).

The DL models are commonly employed to forecast financial assets such as bonds and stocks. These methods are useful in predicting prices, volatility, and directional movements or trends (Singh and Srivastava 2017; Sim et al. 2019; Zhang et al. 2019; Makinen et al. 2019). However, according to a survey, just 13.8 percent of academics who researched

cryptocurrencies between 2013 and 2019 used DL techniques (Fang et al. 2022). Most researchers forecasted cryptocurrency prices rather than volatility using various DL approaches. Shen et al. (2021) compared the RNN and GARCH models for forecasting Bitcoin's volatility.

GARCH-type models generally outperform neural networks for forecasting the volatility of financial assets. This is because neural networks perform best in nonlinear environments and require large data to approximate the estimated function (Laily et al. 2018). The DL models, however, can capture short-term and long-term features by learning complex and nonlinear relationships in financial time series data (Vidal and Kristjanpoller 2020). The DL models capture market dynamics more efficiently than traditional and ML models and, in some cases, can even be used to create automated trading systems (Jeenanunta et al. 2018).

Due to the rapid advancement of DL algorithms in time series forecasting, researchers have begun to utilize them alone or in combination with one or more classical approaches to enhance forecasts. However, a limited number of studies have integrated GARCH-type models with the DL algorithm in cryptocurrency price modeling. We believe that merging volatility models with an RNN can help improve volatility forecasts by appropriately modeling volatility processes using the sequence-based learning capabilities of RNNs. Additionally, combining the strengths of the two models may result in more precise volatility projections by combining the data derived by GARCH models and using it as input data to enable RNNs to adjust to volatility processes. Future studies on GARCH-DL hybrid models may use this study's findings as a motivation.

This research contributes to the existing literature on the hybrid DL model in the following ways. Firstly, it combines various GARCH models with DL algorithms such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM (BiLSTM) algorithms with single, double, and triple layer network architectures to forecast the realized volatility of Bitcoin. Secondly, it utilizes the parameters of GARCH models, daily log-returns, squared log-returns, and volatility indicators, such as the relative volatility index (RVI) and relative strength index (RSI), as inputs into the DL algorithms to improve the volatility forecasts of Bitcoin. These combinations of hybrid models and indicators have yet to be considered extensively for forecasting the realized volatility of Bitcoin's price movements.

In our study, a combination of hybrid models was used to forecast realized Bitcoin price volatility at multiple time horizons (7-, 14-, and 21-day-ahead) using a rolling window technique, and whereby the performance was assessed using loss functions such as the Heteroscedasticity-Adjusted Mean Absolute Error (HMAE), and Heteroscedasticity-Adjusted Mean Squared Error (HMSE) loss functions, as well as the Model Confidence Set (MCS) procedure. These contributions have the potential to have a significant impact on the field of Bitcoin volatility forecasting. The hybrid models considered in this study may encourage practitioners and investors to get accurate and reliable volatility estimates, thereby improving risk management initiatives.

The remainder of this study is structured as follows: Section 2 provides the specification of DL and hybrid models and Section 3 discusses the data, methods, and evaluation measures, respectively. Section 4 provides a discussion of the findings, while Section 5 concludes our study.

2. Model Specifications

2.1. Volatility Models

The standard GARCH model and two asymmetric GARCH models were used to model the volatility of Bitcoin. The GARCH model of Bollerslev (1986) has been considered one of the most popular volatility models. In this model, conditional variance is defined as a function of the historical values of both the squared residuals and the conditional variance. Leptokurtosis and volatility clustering are both captured by the

GARCH model, but time-dependent asymmetry, which is regarded as a key stylized aspect of volatility, is frequently left out (Kim and Won 2018). Hence, the GARCH model considers the shock's magnitude but not the positive or negative direction (Muhammed and Faruk 2018). Various extensions of the standard GARCH model have been proposed to accommodate this asymmetric volatility characteristic.

Nelson's (1991) Exponential GARCH (EGARCH) model accurately depicts the asymmetric effect of both positive and negative shocks on volatility. The parameters can be unconstrained in this model's logarithmic form while still maintaining a positive conditional variance. Additionally, the conditional variance is a function of prior standardized innovations rather than using past innovations (Naimy et al. 2021). The GJR model of Glosten et al. (1993), which is comparable to the EGARCH model in that it additionally takes into account the asymmetric impact of positive and negative shocks, is another widely used asymmetric GARCH model.

2.2. Recurrent Neural Networks

The RNN is generally suited for processing sequential problems (Siegelmann and Sontag 1991). It is a specific type of ANN that consists of multiple layers, known as recurrent layers. These layers operate sequentially and map the sequences to other sequences, resulting in a network with a higher overall performance than ANN. The RNN retains information in its internal state, referred to as a memory cell. The output of RNN networks at a specific time interval depends on the input at that time interval and the network's state in the preceding time interval. The network, however, either stops learning or keeps learning at a very high learning rate; as a result, it is unable to grasp the concept of the smallest error. This vanishing gradient problem makes it more difficult to train the network. (Mehtab et al. 2020).

On the other hand, RNN is incapable of storing long-term memory (Moghar and Hamiche 2020). Therefore, its performance is not considered adequate when the learning requires long-term sequential dependencies and is hence considered incapable of forecasting samples with long-time data (Hochreiter and Schmidhuber 1997). The RNN architecture is presented in Figure 1.

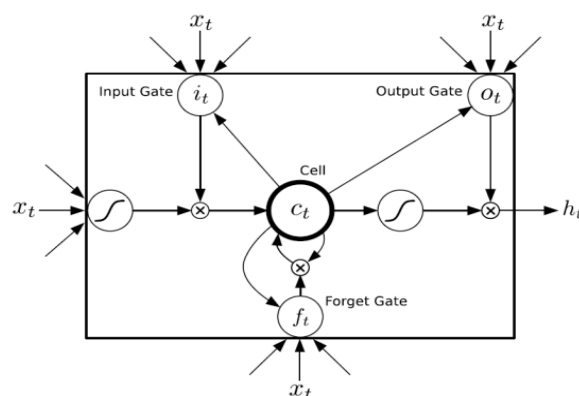


Figure 1. Long Short-Term Memory cell.

2.3. Long Short-Term Memory

The LSTM is a particular type of RNN that deals with sequential data (Hochreiter and Schmidhuber 1997). It consists of a memory cell and different gates; the former store the information while later updates it (Kraus and Feuerriegel 2017). The LSTM often provides better predictions than the RNN for long sequential data. The structure or architectures of LSTM networks make it possible to forget past irrelevant information, thus resolving the vanishing gradient problem. Hence, such networks are very suitable for modeling complex time series. Each memory cell has three sigmoid layers and one

tanh layer (Mehtab et al. 2020). Figure 2 displays the structure of the LSTM cell, including three special gates: forget gate, input gate, and output gate. The output of the LSTM unit is represented by h_t , while c_t represents the value of the memory cell.

The forget gate determines which cell state information is deleted from the LSTM unit. It is employed to weed out extraneous memories from the past and retain only knowledge pertinent to the present situation. The memory cell accepts the output h_{t-1} of the previous moment and the external information represented by X_t of the current moment as inputs and combines them in a long vector $[h_{t-1}, X_t]$ by σ transformation to become a forget gate:

$$f_t = \sigma (W_c[h_{t-1}, X_t] + b_c) \quad (1)$$

In Equation (1), W_c, b_c represent the weight matrix and bias of the forget gate, respectively, and σ is the sigmoid function. The forget gate's main function is to record how much the cell state C_{t-1} of the previous time is reserved for the cell state C_t of the current time.

The input gates control the new information that acts as the input to the current state of the network. It reserved the current input X_t and passes it into the cell state C_t which prevents insignificant content from entering the memory cells. It has two functions; one is to find the state of the cell which must be updated and the updated value is selected by the sigmoid layer, as in Equation (2), and the other function is to update the information in the cell state. The input gate determines how updated the cell state is:

$$i_t = \sigma (W_i[h_{t-1}, X_t] + b_i) \quad (2)$$

Meanwhile, a new candidate vector \hat{C}_t is created through the tanh layer to control how much new information is added, as presented in Equation (3):

$$\hat{C}_t = \tanh (W_c[h_{t-1}, X_t] + b_c) \quad (3)$$

Finally, Equation (4) is used to update the cell state of the memory cells:

$$C_t = (f_t * C_{t-1}) + (i_t * \hat{C}_t) \quad (4)$$

The old cell state, the forget gate, the input gate, and the intermediate cell state are added. The new state is then calculated using the operation's result. Thus, LSTM is ideal for sequence prediction thanks to this enhanced cell with four interacting layers instead of just one sigma cell or tanh in RNN (Struga and Qirici 2018).

The output gate determines the LSTM cell's next hidden state. A sigmoid layer determines the output information first, and the cell state is processed by tanh and multiplied by the sigmoid layer's output to generate the final output component. Finally, the output gates serve to output the network's output at the specified time (Qiu et al. 2020):

$$\text{Output gate: } o_t = \sigma (W_o h_{t-1}, X_t + b_o) \quad (5)$$

The final output value of the cell is defined as follows:

$$h_t = o_t * \tanh(C_t) \quad (6)$$

The output can be considered as the forecasted value computed by the model for the current state (Struga and Qirici 2018).

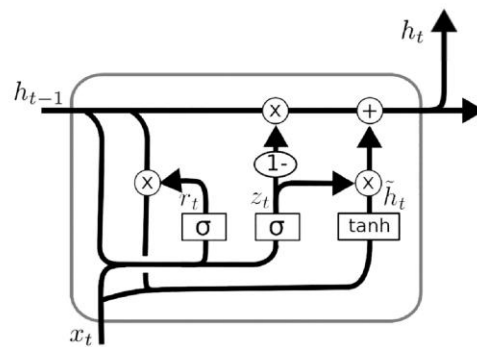


Figure 2. The architecture of the Gated Recurrent Unit.

2.4. Gated Recurrent Unit

This is a more recent alteration of LSTM introduced by Cho et al. (2014) to analyze sequential data. It combines the forget gate and input gate into a single update gate, z_t , and merges the cell state C_t and hidden state h_t . The GRU structure is simple compared to the LSTM. The hidden state h_t generated at each time step t is defined as follows:

$$z_t = \sigma(w_z[h_{t-1}, x_t]) \quad (7)$$

$$r_t = \sigma(w_r[h_{t-1}, x_t]) \quad (8)$$

$$\hat{h}_t = \tanh(w_h[r_t \Delta h_{t-1}, x_t]) \quad (9)$$

$$h_t = (1 - z_t) \Delta h_{t-1} + z_t \Delta \hat{h}_t \quad (10)$$

GRU might be able to learn the data at the combined gate by streamlining the LSTM's architectural design. This single update gate, though, might not be able to fully uncover some hidden information. Consequently, the effectiveness of GRU networks may be diminished when attempting to forecast long-term time series (Vo et al. 2019).

The GRU has fewer tensor operations; therefore, they are faster to train than LSTM. Researchers usually try both to determine which one works better for analyzing different financial markets and their unique data sets.

2.5. Bidirectional LSTM

Graves and Schmidhuber (2005) introduced the BiLSTM model, a bidirectional RNN variant that combines a forward and backward unidirectional LSTM as expressed in Equation (11).

$$\text{Concatenate}(h_t) = [\vec{h}_t \ \overleftarrow{h}_t] \quad (11)$$

In contrast to BiLSTM, which uses the combined two hidden layers, unidirectional LSTM can only store long-term information from prior observations. As shown in Figure 3, it divides the RNN's neurons in two directions. While the other makes use of forward states or positive time directions, the first is for backward states or negative time directions. As a result, this technique uses two-time directions and input data from the past and future of the current time frame (Schuster and Paliwal 1997).

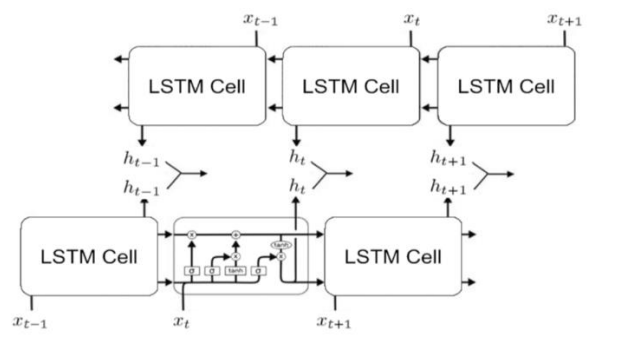


Figure 3. The architecture of BiLSTM Cell.

3. Data and Methodology

3.1. Data Description

The data used in this study consists of daily Bitcoin closing prices from 1 January 2015, to 31 March 2021, a total of 2283 observations. The data were extracted from www.Bitcoincharts.com (accessed on 5 April 2022). The log-returns are calculated as:

$$r_t = \log(p_t) - \log(p_{t-1}), \quad (12)$$

where p_t is the closing price of Bitcoin at time t , and p_{t-1} represents the closing price of Bitcoin at a previous time. The realized variance was calculated by aggregating the squares of Bitcoin log-returns:

$$\text{Realized variance} = \sum_{i=1}^n r_t^2 \quad (13)$$

where r_t^2 represents the square of log-returns, and n represents the number of observations within a day. The square root of the realized variance is known as realized volatility:

$$RV_t = \sqrt{\sum_{i=1}^n r_t^2} \quad (14)$$

The initial N observations were considered for the in-sample period to estimate the parameters, while the remaining observations, from 15 June 2020, through 31 March 2021, which resulted in approximately 287 observations, were left for the forecast evaluation.

3.2. Evaluation Measures

Two nonlinear loss functions, HMSE and HMAE used in this study (Kristjanpoller and Minutolo 2016; Kim and Won 2018; Fuertes et al. 2009) to assess the forecasting performance of models. The mathematical formulas of loss functions are given as follows:

$$HMSE = \frac{1}{n} \sum_{t=1}^n (1 - \sigma_t / RV_t)^2 \quad (15)$$

$$HMAE = \frac{1}{n} \sum_{t=1}^n |1 - \sigma_t / RV_t| \quad (16)$$

In these equations, RV_t and σ_t represent the observed and forecasted realized volatility, respectively, while n represents the out-of-sample size. The relative importance of the models was evaluated and MCS selected the best model. The MCS process consists of a series of tests that, by accepting the equal predictive ability (EPA) null hypothesis at a specified confidence level, enable the building of a set of superior models. This procedure evaluates the EPA statistic for loss functions, HMSE, and HMAE (Shang and Haberman 2018).

3.3. Experiments

Three GARCH-type models (GARCH, EGARCH, and GJR) and three DL models (LSTM, BiLSTM, and GRU) were used to model the realized volatility of Bitcoin. First, volatility estimates from GARCH-type models were obtained. These volatility estimates

were fed into DL models as input variables. The single, double, and triple GARCH-type models were then combined with LSTM, BiLSTM, and GRU models with different layers to improve the volatility forecasts of Bitcoin. More specifically, a single GARCH model with single, double, and triple layer DL models (GARCH-LSTM₁, GARCH-BiLSTM₁, GARCH-GRU₁, GJR-LSTM₁, ..., GARCH-LSTM₂, ..., EGARCH-LSTM₃, etc.), double GARCH models with single, double and triple layer DL models (GARCH-GJR-LSTM₁, GARCH-GJR-BiLSTM₁, GARCH-GJR-GRU₁, GJR-EGARCH-LSTM₁, ..., GARCH-GJR-LSTM₂, ..., GJR-EGARCH-LSTM₃, etc.) and triple GARCH models with single, double and triple layer DL models (GARCH-GJR-EGARCH-LSTM₁, GARCH-GJR-EGARCH-BiLSTM₁, GARCH-GJR-EGARCH-GRU₁, ..., GARCH-GJR-EGARCH-LSTM₂, ..., GARCH-GJR-EGARCH-LSTM₃, etc.). In this way, a total of 75 models were fitted to the in-sample data, and forecasts of 7-, 14-, and 21-day-ahead realized volatility of Bitcoin were obtained. In the GARCH-LSTM₁ model, the estimated volatility of the GARCH model was used as an input to the single-layer LSTM models along with other inputs, such as daily log-returns, squared log-returns volatility indicators RSI and RVI, in order to get better forecasts of the Bitcoin volatility. This strategy can more effectively capture volatility clustering, leptokurtosis, and the leverage effect of the returns.

For training the DL models, the network architecture was specified to have at most three LSTM/BiLSTM/GRU layers with 128, 64, and 32 neurons and a dense layer. The Adam optimizer was used for training the network and a dropout of 0.3 was added between the first and second layers. All the networks were trained for 150 epochs. The architecture was kept the same for all DL and hybrid models to allow for a more even assessment of the forecasts under the same network architecture. For initial training of the DL models, the rolling window was set at $(t - 12)$ to forecast $(t + n)$ -day-ahead realized volatility, with $n = 7, 14$, and 21 . All the models were fitted using Python.

4. Results and Discussion

Table 1 summarizes the summary statistics of Bitcoin closing prices, Bitcoin log-returns, Bitcoin squared log returns, and volatility indicators RSI and RVI. The Bitcoin closing prices showed a large mean value, high standard deviation, positive skewness, and high kurtosis, which are commonly observed in financial time series. Other researchers have also observed these anomalies in cryptocurrency prices (see Phillip et al. 2020; Zahid et al. 2022, among others). To fit an ARCH method, stationary and normal data are required. The Jarque–Bera (JB) and Augmented Dickey-Fuller (ADF) tests were used to analyze the two conditions. A highly significant value (20,227) of the JB test, at a 5% significance level, confirmed the non-normality in the Bitcoin prices. The ADF statistic of -4.89 at a 5% significance level showed that the series was not stationary for ARCH modeling.

Furthermore, the ARCH-LM test with a highly significant value (243.5) at a 5% significance level indicated strong autoregressive conditional heteroscedastic effects in the residual variances. The ADF test on log-returns rejected the null hypothesis of a unit-root process. The JB test rejected the hypothesis that the log returns distribution was normal. Similarly, the ARCH-LM test at 12 lags was also rejected, indicating heteroscedasticity in Bitcoin log-returns. The Ljung-Box test up to the lag 20 was also rejected. It revealed that Bitcoin log-returns were appropriate for fitting the GARCH model.

Figure 4 shows the charts of daily prices, log-returns, squared log-returns, and realized volatility of Bitcoin from 1 January 2015, to 31 March 2021. It can be observed from the figure that the price of Bitcoin started increasing in the mid of 2017 with slight variations till 2020 until it reached its top value at the start of 2020. The daily log-returns of Bitcoin show volatility clustering.

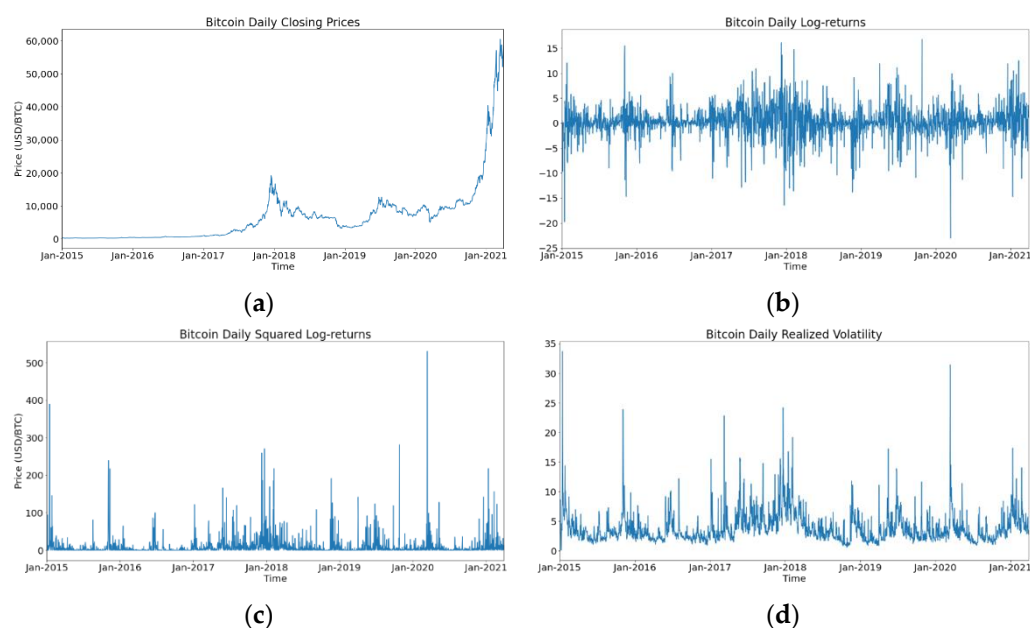


Figure 4. Bitcoin (a) daily closing prices, (b) log-returns, (c) squared log-returns, and (d) realized volatility from 1 January 2015 to 31 March 2021.

Figure 5 compares Bitcoin's log-returns with the estimated volatility of GARCH-type models.

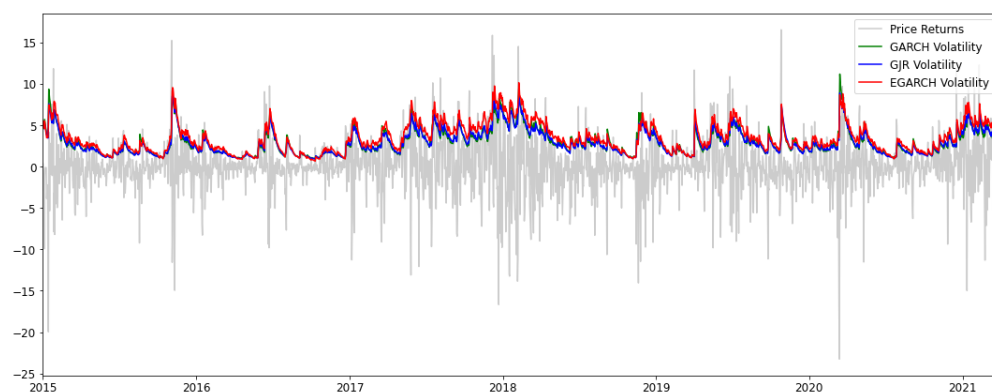


Figure 5. Actual Bitcoin returns and GARCH-type models' estimated volatility from 1 January to 31 March 2021.

Table 1. Summary statistics of Bitcoin closing prices, volatility indicators, Bitcoin log-returns, and square log-returns.

	Mean	Std. Dev.	Skewness	Kurtosis	JB	ADF	ARCH(12)	$Q^2(20)$
Bitcoin closing prices	6698.595	9279.506	3.290	13.017	20,227.478	−4.897	243.488 ***	38,493.172 ***
RSI	55.674	16.669	0.072	−0.4301	19.576	−6.936	2107.303 ***	17,499.154 ***
RVI	55.908	15.739	0.053	−0.073	1.590	−8.594	1763.135 ***	8421.058 ***
Bitcoin log-returns	0.004	3.229	15.820	923.830	2968.189	−31.786	243.488 ***	159.362 ***
Bitcoin squared log-returns	10.472	28.344	7.441	85.787	720,809.075	−6.713	86.503 ***	512.371 ***

ARCH (·) and JB denote the ARCH-LM test statistic and Jarque-Bera normality test statistics, respectively, while $Q^2(20)$ denotes the Ljung-Box (Q -statistics) for the squared error terms up to lag 20, *** denotes the significance of the test at the 1% level.

Figure 6 utilizes proxies of Bitcoin squared return, also called RV, and compares it with GARCH-type model volatility. RV values seem close to volatility estimated by the GARCH type model and visualized.

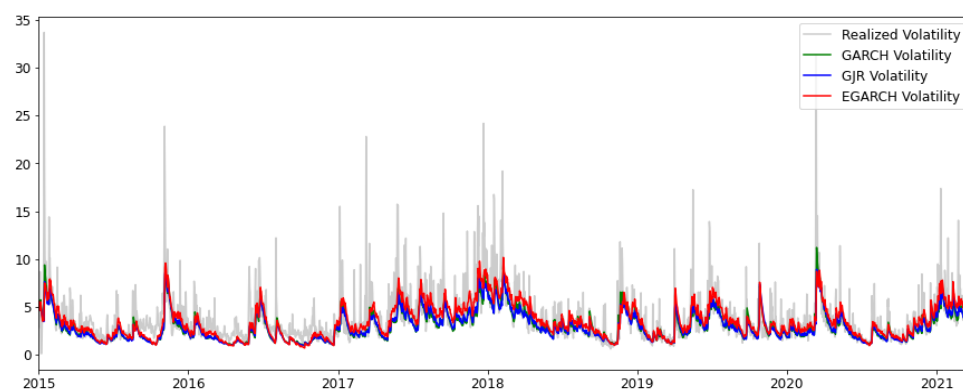


Figure 6. Bitcoin's realized volatility and GARCH-type models' estimated volatility from 1 January 2015 to 31 March 2021.

In Figure 7, the probability plot of the Bitcoin log-returns confirms the results of Table 1. Similarly, Q and ADF statistics indicated statistical significance, as in Table 1. The autocorrelation plot showed that the first five lags, such as 1, 10, 20, 28, and 45 seem significant, while the consecutive lags approached the significance line. This confirms the stationarity of the time series. On the other hand, partial autocorrelations at lags 1, 10, and 28 were found statistically significant. The subsequent lags are near the significance line. As a result, the PACF suggested fitting a first-order autoregressive model.

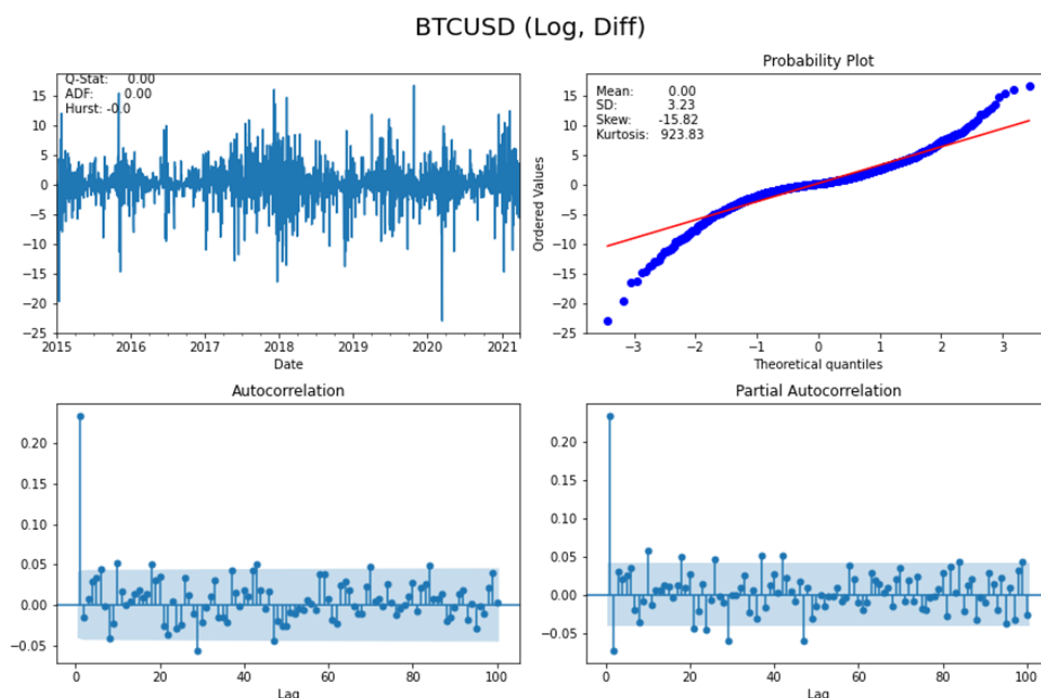


Figure 7. Bitcoin log-returns, Q-Q plot, autocorrelation, and partial autocorrelation plots.

4.1. Estimation Results

Three GARCH-type models (GARCH, GJR, and EGARCH) were fitted to the in-sample data and the results of the parameters estimates, along with their standard errors, are presented in Table 2. All three GARCH models' estimated parameters were highly

significant, with the GJR and EGARCH models confirming the leverage effect in log-returns of Bitcoin.

Table 2. Parameters estimates of the GARCH-type models.

Model	ω	α	γ	β
GARCH(1,1)	0.1347 *** (0.008)	0.1758 *** (0.0003)	—	0.8242 *** (0.0004)
GJR(1,1)	0.1150 *** (0.0008)	0.1907 *** (0.0004)	−0.0511 *** (0.0003)	0.8340 *** (0.0005)
EGARCH(1,1)	0.1083 *** (0.0004)	0.3744 *** (0.0008)	0.0245 *** (0.0002)	0.9824 *** (0.0001)

Standard error in parenthesis. *** means significance at a 1% level.

4.2. Out-of-Sample Forecast Results

This section presents the results of out-of-sample forecasts of models considered in this study. For the sake of brevity, the results of models with better performance are presented. Table 3 shows the result of out-of-sample forecasts of GARCH models and single GARCH models combined with two layers of DL models.

It was observed from this table that the EGARCH model has a lower HMAE and HMSE than GARCH and GJR models. The EGARCH model exhibited 1.57%, 2.24%, and 2.37% improvement in HMAE, whereas it attained 2.86%, 3.08%, and 3.14% improvement in HMSE against the GARCH model at 7-, 14- and 21-day-ahead forecasts, respectively. Similarly, the EGARCH achieved 1.66%, 1.76%, and 1.87% improvement in HMAE, while it attained 2.61%, 2.55%, and 3.13% in HMSE from the GJR model at selected days ahead forecasts, respectively.

Next, we combined single GARCH models with two-layer DL models to investigate whether including single GARCH volatility as inputs to DL models improves the volatility forecasts based on the HMAE and HMSE loss functions at the selected time horizons. The results are tabulated in Table 3 and show that the EGARCH-LSTM₂ model had the minimum errors in a class of single hybrid models. This model exhibited 4.45%, 4.85%, and 5.58% relative improvement in HMAE, while 5.5%, 5.78%, and 6.06% in HMSE against the E-GARCH model at 7-, 14-, and 21-day-ahead ahead forecasts, respectively.

Even the single hybrid models, which showed the worst performances in a class of the LSTM₂, had improved performances over the E-GARCH model. The GARCH-LSTM₂ improved the performances by 1.93%, 2.35%, and 3.09% in HMAE and 2.45%, 3.23%, and 3.10% in HMSE against the E-GARCH model. Similarly, the GJR-LSTM₂ was shown to have an improvement of 1.94%, 2.36%, and 3.10% in HMAE and 3.22%, 3.36%, and 3.64% in HMSE at 7-, 14-, and 21-day-ahead forecasts, respectively.

These results revealed that a single GARCH-type model hybrid with DL improved the volatility forecasts and that adding other features, such as RSI and RVI, further strengthened the predictability of Bitcoin price volatility. The performance of E-GARCH-LSTM₂ was found better in a class of single GARCH-type models hybrid with the DL algorithm.

Table 3. Out-of-sample forecast results for GARCH-type and hybrid models.

Models	7-Day-Ahead Forecast		14-Day-Ahead Forecast		21-Day-Ahead Forecast	
	HMAE	HMSE	HMAE	HMSE	HMAE	HMSE
GARCH	0.40406	0.40106	0.40857	0.39108	0.41220	0.40008
GJR	0.40445	0.39999	0.40657	0.38999	0.41009	0.40001
EGARCH	0.39770	0.38956	0.39940	0.37999	0.40240	0.38750
GARCH-LSTM ₂	0.39001	0.38001	0.38995	0.36700	0.38995	0.37550

GARCH-GRU ₂	0.39450	0.38303	0.39545	0.39303	0.41692	0.39663
GARCH-BiLSTM ₂	0.39455	0.38362	0.39551	0.39463	0.42082	0.39620
GJR-LSTM ₂	0.38794	0.37699	0.38798	0.36720	0.38991	0.37340
GJR-GRU ₂	0.39650	0.38079	0.39740	0.39977	0.39920	0.37998
GJR-BiLSTM ₂	0.39555	0.38270	0.39605	0.39621	0.41805	0.38721
EGARCH-LSTM ₂	0.37998	0.36799	0.37999	0.35800	0.37991	0.36400
EGARCH-GRU ₂	0.39705	0.38354	0.39905	0.37655	0.40002	0.38632
EGARCH-BiLSTM ₂	0.39736	0.38680	0.39740	0.38759	0.39748	0.38498

HMAE: Heteroscedasticity-adjusted mean absolute errors; HMSE: Heteroscedasticity-adjusted mean squared error; bold values represent the least value for each column.

Next, we combined double and triple GARCH models with two-layer DL models to investigate whether including double and triple GARCH volatility as inputs to DL models improves the volatility forecasts. Volatility indicators such as RSI and RVI were also added as input variables to further improve the volatility forecast of hybrid models. Table 4 presents the out-of-sample forecasts for double and triple GARCH-type models combined with two layers of DL models along with the volatility indicators.

Table 4. Out-of-sample forecasts results for GARCH-type double and triple models, hybrid with DL models.

Models	7 Days Ahead Forecast		14 Days Ahead Forecast		21 Days Ahead Forecast	
	HMAE	HMSE	HMAE	HMSE	HMAE	HMSE
GARCH-GJR-LSTM ₂	0.35505	0.33935	0.35400	0.32940	0.35301	0.33440
GARCH-GJR-GRU ₂	0.37876	0.33511	0.37881	0.33447	0.37901	0.33373
GARCH-GJR-BiLSTM ₁	0.37801	0.34511	0.37825	0.34541	0.37900	0.34722
GARCH-EGARCH-LSTM ₂	0.34601	0.33250	0.34440	0.32150	0.34240	0.32650
GARCH-EGARCH-GRU ₂	0.37581	0.34303	0.37602	0.34325	0.37102	0.34563
GARCH-EGARCH-BiLSTM ₁	0.37660	0.35505	0.37851	0.35461	0.37952	0.35561
GJR-EGARCH-LSTM ₂	0.35105	0.33835	0.35100	0.32740	0.35001	0.33240
GJR-EGARCH-GRU ₂	0.37400	0.33851	0.37756	0.33457	0.41373	0.33650
GJR-EGARCH-BiLSTM ₁	0.37410	0.33789	0.33889	0.33889	0.38405	0.33884
GARCH-GJR-EGARCH-LSTM ₂	0.33280	0.30281	0.33282	0.30110	0.32002	0.30020
GARCH-GJR-EGARCH-GRU ₂	0.34325	0.30380	0.35914	0.30604	0.35942	0.30098
GARCH-GJR-EGARCH-BiLSTM ₂	0.33809	0.30381	0.34250	0.30241	0.35010	0.30491

HMAE: Heteroscedasticity-adjusted mean absolute errors; HMSE: Heteroscedasticity-adjusted mean squared error; bold values represent the least value for each column.

The results indicate that double GARCH-type models hybrid with DL models were statistically better than single GARCH-type models hybrid with DL models. More specifically, the GARCH-EGARCH-LSTM₂, combining the GARCH and EGARCH models with a two-layer LSTM model, had the minimum errors in a class of double GARCH-type models. This model achieved 8.94%, 9.36%, and 9.87% improvement in HMAE for a 7-, 14-, and 21-day-ahead forecasts, respectively, when benchmarked against the best-performing single GARCH hybrid model (which was the EGARCH-LSTM₂ model). Similarly, this model attained 9.64%, 10.19%, and 10.30% improvement in HMSE against the best single GARCH hybrid model for 7-, 14-, and 21-day-ahead forecasts, respectively.

Other double GARCH-type hybrid models, such as GARCH-GJR-LSTM₂ and GJR-EGARCH-LSTM₂, also performed better than the single EGARCH-LSTM₂. The GARCH-GJR-LSTM₂ was shown to have an improvement of 6.56%, 6.83%, and 7.08% in HMAE and 7.36%, 7.53%, and 8.41% in HMSE. Similarly, compared to the EGARCH-LSTM₂ model, the forecasting performance of GJR-EGARCH-LSTM₂ increased by 7.61%, 7.63%, and 7.87% in HMAE and 8.05%, 8.55%, and 8.68% in terms of HMSE for 7-, 14-, and 21-day-ahead forecasts, respectively.

The results of triple GARCH models combined with DL models are also presented in Table 4. These hybrid models were found to be better than the single and double GARCH-type models hybrid with DL models. More specifically, GARCH-GJR-EGARCH-LSTM₂ (which combines the GARCH, GJR and EGARCH models with a two-layer LSTM) had the minimum errors in a class of triple GARCH-type models. This model attained 18.61%, 19.88%, and 20.51% improvement in HMAE and 20.04%, 19.88%, and 20.51% improvement in HMSE for 7-, 14-, and 21-day-ahead forecasts when benchmarked against the best performing single GARCH-hybrid model (which was the EGARCH-LSTM₂). On the other hand, the GARCH-GJR-EGARCH-LSTM₂ achieved 10.62%, 11.59%, and 11.80% improvement in HMAE and 11.50%, 11.89%, and 11.80% improvement in HMSE for 7-, 14-, and 21-day-ahead forecast, respectively, when benchmarked against the best performing double GARCH-type models (which was the GARCH-EGARCH-LSTM₂). These results showed that combinations of double and triple GARCH-type models with DL models, along with other features such as RSI and RVI, further improved the forecasts of Bitcoin price volatility.

4.3. Rolling Window Forecast Results

In addition, this study adopted a rolling window approach for improving the performances of hybrid models. This approach used a fixed window length and generated a one-step-ahead forecast while dropping the oldest observation and including a new observation. The aforementioned models were estimated again and a one-step-ahead forecast was generated. In this way, 7-, 14-, and 21-day-ahead forecasts were made. The last 288 days were chosen as a test size for out-of-sample forecasting. The results of the rolling window forecasts are shown in Tables 5 and 6.

Table 5 presents the out-of-sample forecast results for single GARCH-type models hybrid with DL models. It was observed from the table that the rolling window affected the model's performances and reduced the HMAE and HMSE further, as compared to the results of Table 3. The EGARCH-LSTM₂ was found to have lower HMSE and HMAE values. The EGARCH-LSTM₂ was shown to have a relative improvement of 5.18%, 10.56%, and 12.28% in HMAE and 8.71%, 11.41%, and 13.29% in HMSE for 7-, 14-, and 21-day-ahead forecasts, respectively.

Table 6 presents out-of-sample forecast results of double and triple GARCH-type models hybrid with DL models. The results indicated that the double GARCH-EGARCH-LSTM₂ improved by 9.83%, 11.70%, and 11.47% in HMAE and 16.00%, 16.41%, and 18.08% improvement as compared to its single benchmark (which is the EGARCH-LSTM₂).

The triple GARCH-type hybrid with DL models further improved the model's forecasting performance by using a fixed rolling window scheme. More specifically, the triple GARCH-type models with two layers of LSTM (i.e., GARCH-GJR-EGARCH-LSTM₂) improved the performance by 14.45%, 14.64%, and 14.94% against the GARCH-EGARCH-LSTM₂ in terms of HMAE. Likewise, this model attained 16.27%, 17.51%, and 19.53% enhancement over the GARCH-EGARCH-LSTM₂ by employing a rolling window scheme for a one-day-ahead forecast at different time horizons.

To focus more on predictive accuracy dynamics, the performance of GARCH-GJR-EGARCH-LSTM₂ was compared with EGARCH-LSTM₂, which was shown to be the best-performing model among a single class of models. This triple GARCH-LSTM model was shown to have an improvement of 22.86%, 24.63%, and 24.87% in HMAE. Similarly, this model attained 29.70%, 31.05%, and 33.92% in HMSE for 7-, 14-, and 21-day-ahead forecasts, respectively.

Table 5. Out-of-sample forecast results for single GARCH-type models hybrid with DL models, with fixed rolling-window size at different forecasts horizons.

Models	One Day Rolling Window at 7 Days Ahead Forecast		One Day Rolling Window at 14 Days Ahead Forecast		One Day Rolling Window at 21 Days Ahead Forecast	
	HMAE	HMSE	HMAE	HMSE	HMAE	HMSE
GARCH-LSTM ₂	0.38890	0.38761	0.38892	0.38265	0.38895	0.37650
GARCH-GRU ₂	0.39120	0.38000	0.39122	0.38102	0.39105	0.38202
GARCH-BiLSTM ₁	0.39155	0.39263	0.39255	0.39264	0.39355	0.38421
GJR-LSTM ₂	0.38120	0.36604	0.38122	0.36702	0.38225	0.36851
GJR-GRU ₃	0.38704	0.37079	0.38854	0.37178	0.38902	0.37256
GJR-BiLSTM ₂	0.38777	0.37217	0.38901	0.37122	0.38906	0.37522
EGARCH-LSTM ₂	0.37707	0.35561	0.35720	0.33650	0.35299	0.33602
EGARCH-GRU ₂	0.38706	0.36455	0.38746	0.36456	0.38801	0.38298
EGARCH-BiLSTM ₂	0.38737	0.38381	0.38740	0.38385	0.38902	0.38386

HMAE: Heteroscedasticity-adjusted mean absolute errors; HMSE: Heteroscedasticity-adjusted mean squared error; bold values represent the least value for each column.

Interestingly, it is also noteworthy that the models that accomplished the minimum errors by selected loss functions attained greater values of significance associated with their p -values than other models. To find a significantly better model, the analysis was further proceeded by applying the MCS procedure. Table 7 shows the results of the MCS. It compares the multiple forecasting models by generating a set of superior models and picking those models which are statistically significant. In this analysis, 24 models were assessed by the MCS. It was observable that 15 out of the 24 models were statistically significant. This represents 62% of the overall models that were statistically significant.

The EGARCH-LSTM₂ was found statistically significant among single GARCH family models hybrid with LSTM models at different forecast horizons. Similarly, the GARCH-EGARCH-LSTM₂ model was shown to be statistically significant among double GARCH family hybrid with LSTM models. It was interesting to observe that triple GARCH hybrid with DL models attained the topmost count of statistically significant results compared to single and double GARCH hybrid with DL models. The MCS shows that the GARCH-GJR-EGARCH-LSTM₂ is the best model for forecasting Bitcoin volatility at a selected time horizon by considering a p -value less than 0.01.

Table 6. Out-of-sample forecast results for double and triple GARCH-type models hybrid with DL models, with fixed rolling window sizes at different forecasts horizons.

Models	One day Rolling Window at 7 Days Ahead Forecast		One Day Rolling Window at 14 Days Ahead Forecast		One Day Rolling Window at 21 Days Ahead Forecast	
	HMAE	HMSE	HMAE	HMSE	HMAE	HMSE
GARCH-GJR-LSTM ₂	0.37430	0.29930	0.37200	0.28454	0.37250	0.28484
GARCH-GJR-GRU ₂	0.37680	0.30210	0.37680	0.30247	0.37701	0.30373
GARCH-GJR-BiLSTM ₁	0.37600	0.29790	0.37620	0.30021	0.37700	0.30122
GARCH-EGARCH-LSTM	0.34000	0.29857	0.31540	0.28125	0.31250	0.27527
GARCH-EGARCH-GRU ₂	0.37380	0.29420	0.37400	0.30000	0.37102	0.30025
GARCH-EGARCH-BiLSTM ₂	0.37460	0.29300	0.37750	0.29261	0.37752	0.29361
GJR-EGARCH-LSTM ₂	0.37460	0.29450	0.37650	0.28454	0.37752	0.27650
GJR-EGARCH-GRU ₂	0.37200	0.29650	0.377360	0.28257	0.41173	0.28180
GJR-EGARCH-BiLSTM ₁	0.37210	0.29790	0.37220	0.30005	0.38205	0.30084
GARCH-GJR-EGARCH-LSTM ₂	0.29085	0.25000	0.26921	0.23201	0.26580	0.22205
GARCH-GJR-EGARCH-GRU ₂	0.33180	0.27000	0.29124	0.23301	0.24004	0.23101
GARCH-GJR-EGARCH-BiLSTM ₂	0.30610	0.27080	0.30650	0.27141	0.30810	0.26991

HMAE: Heteroscedasticity-adjusted mean absolute errors; HMSE: Heteroscedasticity-adjusted mean squared error; bold values represent the least value for each column.

With respect to existing studies, Hu et al. (2020) incorporated the GARCH forecasts with ANN, LSTM, and BiLSTM methods, along with a group of explanatory variables, to create hybrid models for assessing the volatility forecast of copper price futures. Unlike their study, this study incorporated the GARCH-type models' forecasts with LSTM, BiLSTM, and GRU algorithms with different layers, along with explanatory variables, to generate hybrid models for the measurement of Bitcoin's volatility forecast. This study also highlights that GARCH forecasts could serve as informative features to substantially boost the volatility prediction of asset prices. Our findings complement Kristjanpoller and Hernández (2017) as well as Verma (2021). Empirically, this study further elaborates the significance of multiple hybrid models by using another DL algorithm (i.e., GRU), which has been shown to be an advanced form of RNN in previous studies.

Kim and Won (2018) combined the LSTM model with GARCH, E-GARCH, and Exponential Weighted Moving Average (EWMA) model to develop the GEW-LSTM hybrid model for assessing stock price volatility forecast. They compared hybrid model performance by analyzing the benchmark model as well as a deep-feed forward neural network (DFN) and E-DFN. In contrast, this study compared the performance of their benchmark model while considering measurement errors and also calculated the relative importance of models. They found that GEW-LSTM has the lowest prediction errors in terms of different measurement errors compared to other prescribed models. Somewhat consistent with their study, this study also found that a triple GARCH hybrid with LSTM attained minimum prediction errors in terms of measurement errors and was shown to be statistically significant. Our findings, along with Kim and Won (2018), show the importance of combining multiple traditional models with the DL algorithm rather than only a single traditional model to enhance prediction performance.

Table 7. Model confidence test (MCS) for significance across forecasting models.

Hybrid Models.	Days Ahead Forecast	HMAE	HMSE	Significance
EGARCH-LSTM ₂	7	0.37707	0.35561	**
	14	0.35920	0.33650	**
	21	0.35299	0.33602	*
GARCH-EGARCH-LSTM ₂	7	0.34000	0.29857	*
	14	0.31540	0.28125	**
	21	0.31250	0.27527	*
GARCH-GJR-EGARCH LSTM ₂	7	0.29805	0.25000	**
	14	0.26921	0.23201	***
	21	0.26580	0.22205	*
GARCH-GJR-EGARCH-GRU ₂	7	0.30180	0.27000	**
	14	0.29124	0.23301	**
	21	0.29008	0.23301	**
GARCH-GJR-EGARCH-BiLSTM ₂	7	0.30610	0.27080	**
	14	0.30650	0.27141	*
	21	0.30810	0.27290	**

The significance level associated with the p-value of the MCS test is taken at 1%, 5%, and 10% levels, represented by ***, **, and *, respectively. HMAE denotes heteroscedasticity-adjusted mean absolute error and HMSE denotes heteroscedasticity-adjusted mean squared error. The bold value shows the best-performing models.

5. Concluding Remarks

Bitcoin volatility forecasting has emerged as a dominant research area within cryptocurrency research for academics, market participants, and regulators alike. This study builds and compares three forms of hybrid models by combining GARCH, EGARCH, and GJR-GARCH individually with DL algorithms and then combining single, double, and triple GARCH models with the DL algorithm in order to construct and assess volatility forecasts.

The presence of GARCH, EGARCH, and GJR-GARCH, in a hybrid model captured the well-known stylized facts, such as volatility clustering, leptokurtosis, and the leverage effect, more effectively, and then their forecasts were used as inputs into the LSTM, BiLSTM, and GRU models. These models were considered to assist in learning the high-level temporal pattern in Bitcoin price data, therefore increasing the amount of information for RV to learn more effectively by considering the necessary features for prediction. The other features, such as RSI and RVI, which shared the correlation patterns with Bitcoin price volatility, were also used in DL algorithms to further improve the forecast of Bitcoin price volatility. The predictive performance of the best triple hybrid model (GARCH-GJR-EGARCH-LSTM2) compared to the best single hybrid model (EGARCH-LSTM2) improved by 18.61%, 19.88%, and 20.51% in HMAE, and 20.04%, 19.88%, and 20.51% in HMSE, for the selected days-ahead forecasts.

We further proceeded with our rolling window scheme to generate a one-day-ahead forecast and to investigate whether this approach minimizes the errors and improves model performance. This study achieved optimistic results and demonstrated that the prediction performance of the best triple hybrid model (GARCH-GJR-EGARCH-LSTM2) compared to the best single hybrid model (EGARCH-LSTM2) improved by 22.86%, 24.63%, and 24.87% in terms of HMAE by using the fixed window size. Similarly, the best triple hybrid model attained 29.70%, 31.05%, and 33.92% in HMSE compared to the best single hybrid model, for the selected days-ahead forecasts, by applying the rolling window approach.

The empirical results herein provide evidence that the GARCH-type model forecasts can serve as informative features, along with volatility indicators, to significantly improve the models' predictive power. Moreover, integrating the GARCH-type model with the DL algorithm was found to be an effective approach to construct useful hybrid models to boost the forecast performance of Bitcoin price volatility. The findings of this study can assist regulators and other market participants in better modeling the volatility of cryptocurrencies using hybrid models rather than individual models.

Author Contributions: conceptualization, M.Z. and F.I.; methodology, M.Z.; software, M.Z., F.I. and D.K.; validation, F.I. and D.K.; writing—original draft preparation, M.Z.; writing—review and editing, F.I. and D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study can be downloaded from www.Bitcoincharts.com.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Al-Yahyaee, Khamis Hamed, Walid Mensi, Idries Mohammad Wanas Al-Jarrah, Atef Hamdi, and Kang Sang Hoon. 2019. Volatility forecasting, downside risk, and diversification benefits of Bitcoin and oil and international commodity markets: A comparative analysis with yellow metal. *The North American Journal of Economics and Finance* 49: 104–20.
- Aysan, Ahmet Faruk, Asad Ul Islam Khan, and Humeysra Topuz. 2021. Bitcoin and Altcoins Price Dependency: Resilience and Portfolio Allocation in COVID-19 Outbreak. *Risks* 9: 74. <https://doi.org/10.3390/risks9040074>.
- Baur, Dirk G., and Thomas Dimpfl. 2018. Asymmetric volatility in cryptocurrencies. *Economics Letters* 173: 148–51.
- Bollerslev, Tim. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31: 307–27.
- Bouoiyour, Jamal, and Refk Selmi. 2016. Bitcoin: A beginning of a new phase. *Economics Bulletin* 36: 1430–40.
- Bowden, James, King Timothy, Dimitrios Koutmos, Tiago Loncan, and Francesco Saverio Stentella Lopes Stentella. 2021. A Taxonomy of FinTech Innovation. In *Disruptive Technology in Banking and Finance*. London: Palgrave Macmillan, pp. 47–91.
- Butner, Johnatan E., Ascher K. Munion, Brian R. W. Baucom, and Alexander Wong. 2019. Ghost hunting in the nonlinear dynamic machine. *PLoS ONE* 14: e0226572.
- Charles, Amélie, and Olivier Darné. 2019. Volatility estimation for Bitcoin: Replication and robustness. *International Economics* 157: 23–32.
- Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Paper presented at 2014

- Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 26–28; Association for Computational Linguistics, Stroudsburg, PA, USA; pp. 1724–34.
- Chu, Jeffrey, Stephen Chan, Saralees Nadarajah, and Joerg Osterrieder. 2017. GARCH Modelling of Cryptocurrencies. *Journal of Risk and Financial Management* 10: 17. <https://doi.org/10.3390/jrfm10040017>.
- Conrad, Christian, Anessa Custovic, and Eric Ghysels. 2018. Long- and Short-Term Cryptocurrency Volatility Components: A GARCH-MIDAS Analysis. *Journal of Risk and Financial Management* 11: 23. <https://doi.org/10.3390/jrfm11020023>.
- Fang, Fan, Carmine Ventre, Michail Basios, Leslie Kanthan, David Martinez-Rego, Fan Wu, and Lingbo Li. 2022. Cryptocurrency trading: A comprehensive survey. *Financial Innovation* 8: 1–59. <https://doi.org/10.1186/s40854-021-00321-6>.
- Feuerriegel, Stefan, and Ralph Fehrer. 2016. Improving decision analytics with deep learning: The case of financial disclosures. Paper presented at 24th European Conference on Information Systems, Istanbul, Turkey, June 12–15, vol. I, pp. 1–10.
- Fuertes, Aana-Maria, Marwan Izzeldin, and Elena Kalotychou. 2009. On forecasting daily stock volatility: The role of intraday information and market conditions. *International Journal of Forecasting* 25: 259–81.
- Gbadebo, Adedeji D., Ahmed O. Adekunle, Adedokun Wole, Adebayo-Oke A. Lukman, and Joseph Akande. 2021. BTC price volatility: Fundamentals versus information. *Cogent Business & Management* 8: 1. <https://doi.org/10.1080/23311975.2021.1984624>.
- Glosten, Lawrence R., Ravi Jagannathan, and David E. Runkle. 1993. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance* 48: 1779–801. <https://doi.org/10.1111/j.1540-6261.1993.tb05128.x>.
- Graves, Alex, and Jurgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18: 602–10. <https://doi.org/10.1016/j.neunet.2005.06.042>.
- Gyamerah, Samuel A. 2019. Modelling the volatility of Bitcoin returns using GARCH models. *Quantitative Finance and Economics* 3: 739–53. <https://doi.org/10.3934/QFE.2019.4.739>.
- Hinton, Geoffrey Everest, and Ruslan Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313: 504–07. <https://doi.org/10.1126/science.1127647>.
- Hochreiter, Sepp, and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9: 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Hu, Yan, Jian Ni, and Liu Wen. 2020. A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction. *Physica A: Statistical Mechanics and its Applications* 557: 124907.
- Jeenanunta, Chawalit, Rujira Chaysiri, and Laksmei Thong. 2018. Stock price prediction with long short-term memory recurrent neural network. Paper presented at 2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES), Khon Kaen, Thailand, May 7–9, pp. 1–7.
- Kamnitsas, Konstantinos, Christian Ledig, Virginia F. J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, and Ben Glocker. 2017. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis* 36: 61–78. <https://doi.org/10.1016/j.media.2016.10.004>.
- Katsiampa, Paraskevi. 2017. Volatility estimation for Bitcoin: A comparison of GARCH models. *Economics Letters* 158: 3–6. <https://doi.org/10.1016/j.econlet.2017.06.023>.
- Kim, Ha Y., and Chang H. Won. 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications* 103: 25–37.
- King, Timothy, Dimitrios Koutmos, and F. S. Stentella Lopes. 2021. Cryptocurrency Mining Protocols: A Regulatory and Technological Overview. In *Disruptive Technology in Banking and Finance*. London: Palgrave Macmillan, pp. 93–134. https://doi.org/10.1007/978-3-030-81835-7_4.
- Koutmos, Dimitrios. 2015. Is there a positive risk-return tradeoff? A forward-looking approach to measuring the equity premium. *European Financial Management* 21: 974–1013. <https://doi.org/10.1111/eufm.12043>.
- Koutmos, Dimitrios. 2022. Investor sentiment and bitcoin prices. *Review of Quantitative Finance and Accounting* 1–29. <https://doi.org/10.1007/s11156-022-01086-4>.
- Kraus, Mathias, and Stefan Feuerriegel. 2017. Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems* 104: 38–48. <http://doi.org/10.1016/j.dss.2017.10.001>.
- Kristjanpoller, Warner D., and Esteban Hernández. 2017. Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Systems with Applications* 84: 290–300. <https://doi.org/10.1016/j.eswa.2017.05.024>.
- Kristjanpoller, Warner D., and Marcel C. Minutolo. 2016. Forecasting volatility of oil price using an artificial neural network-GARCH model. *Expert Systems with Applications* 65: 233–41. <https://doi.org/10.1016/j.eswa.2016.08.045>.
- Kristjanpoller, Warner D., and Marcel C. Minutolo. 2018. A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. *Expert Systems with Applications* 109: 1–11. <http://doi.org/10.1016/j.eswa.2018.05.011>.
- Kyriazis, Nikolaos A. 2021. A Survey on Volatility Fluctuations in the Decentralized Cryptocurrency Financial Assets. *Journal of Risk and Financial Management* 14: 293. <https://doi.org/10.3390/jrfm14070293>.
- Lahmiri, Salim, and Stelios Bekiros. 2019. Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals* 118: 35–40.
- Laily, Vania O. N., Budi Warsito, and I. M. Di Asih. 2018. Comparison of ARCH/GARCH model and Elman Recurrent Neural Network on data return of closing price stock. *Journal of Physics: Conference Series* 1025: 1–12.

- Lim, Ching M., and Siok K. Sek. 2013. Comparing the performances of GARCH-type models in capturing the stock market volatility in Malaysia. *Procedia Economics and Finance* 5: 478–87. [https://doi.org/10.1016/S2212-5671\(13\)00056-7](https://doi.org/10.1016/S2212-5671(13)00056-7).
- Makinen, Ymir, Juho Kannianen, Moncef Gabbouj, and Alexandros Iosifidis. 2019. Forecasting jump arrivals in stock prices: New attention-based network architecture using limit order book data. *Quantitative Finance* 19: 2033–50. <https://doi.org/10.48550/arXiv.1810.10845>.
- Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE* 13: e0194889. <https://doi.org/10.1371/journal.pone.0194889>.
- Mehtab, Sidra, Jaydip Sen, and Abhishek Dutta. 2020. Stock price prediction using machine learning and LSTM-based deep learning models. In *Symposium on Machine Learning and Metaheuristics Algorithms, and Applications*. Singapore: Springer, pp. 88–106. https://doi.org/10.1007/978-981-16-0419-5_8.
- Moghar, Adil, and Mhamed Hamiche. 2020. Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science* 170: 1168–73. <https://doi.org/10.1016/j.procs.2020.03.049>.
- Muhammed, Garzali, and Bashir U. Faruk. 2018. The relevance of GARCH-family models in forecasting Nigerian oil price volatility. *Central Bank of Nigeria Bullion* 42: 14–30.
- Naimy, Viviane, Omar Haddad, Gema Fernández-Avilés, and Rim El Khoury. 2021. The predictive capacity of GARCH-type models in measuring the volatility of crypto and world currencies. *PLoS ONE* 16: e0245904. <https://doi.org/10.1371/journal.pone.0245904>.
- Nelson, Daniel B. 1991. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society* 59: 347–70. <https://doi.org/10.2307/2938260>.
- Pabuçcu, Hakan, Ongan Serdar, and Ayse Ongan. 2020. Forecasting the movements of Bitcoin prices: An application of machine learning algorithms. *Quantitative Finance and Economics* 4: 679–92. <https://doi.org/10.3934/QFE.2020031>.
- Peng, Yaohua, Pedro Henrique Melo Albuquerque, Jader Martins Camboim de Sá, Ana Julia Akaishi Padula, and Mariana R. Montenegro. 2018. The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. *Expert Systems with Applications* 97: 177–92. <https://doi.org/10.1016/j.eswa.2017.12.004>.
- Phillip, Andrew, Jennifer S. K. Chan, and Shelton Peiris. 2020. A new look at cryptocurrencies. *Economics Letters* 16: 6–9. <https://doi.org/10.1016/j.econlet.2017.11.020>.
- Qiu, Jiayu, Bin Wang, and Changjun Zhou. 2020. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* 15: e0227222. <https://doi.org/10.1371/journal.pone.0227222>.
- Schuster, Mike, and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45: 2673–81. <https://doi.org/10.1109/78.650093>.
- Seo, Monghwan, and Geonwoo Kim. 2020. Hybrid Forecasting Models Based on the Neural Networks for the Volatility of Bitcoin. *Applied Sciences* 10: 4768. <http://doi.org/10.3390/app10144768>.
- Shang, Han L., and Steven Haberman. 2018. Model confidence sets and forecast combination: An application to age-specific mortality. *Genus* 74: 1–23.
- Shen, Ze, Qing Wan, and David J. Leatham. 2021. Bitcoin Return Volatility Forecasting: A Comparative Study between GARCH and RNN. *Journal of Risk and Financial Management* 14: 1–18. <https://doi.org/10.3390/jrfm14070337>.
- Sieglmann, Hava T., and Eduardo D. Sontag. 1991. Turing computability with neural nets. *Applied Mathematics Letters* 4: 77–80. [https://doi.org/10.1016/0893-9659\(91\)90080-F](https://doi.org/10.1016/0893-9659(91)90080-F).
- Sim, Hyun S., Hae I. Kim, and Jae J. Ahn. 2019. Is deep learning for image recognition applicable to stock market prediction? *Complexity* 2019: 4324878. <https://doi.org/10.1155/2019/4324878>.
- Singh, Ritika, and Shashi Srivastava. 2017. Stock prediction using deep learning. *Multimedia Tools and Applications* 76: 18569–84. <https://doi.org/10.1007/s11042-016-4159-7>.
- Struga, Kejsi, and Olti Qirici. 2018. Bitcoin Price Prediction with Neural Networks. In *Recent Trends and Applications in Computer Science and Information Technology (RTA-CSIT)*, Tirana, Albania, May 21–22, pp. 41–49. Available online: <https://www.semanticscholar.org/paper/Bitcoin-Price-Prediction-with-Neural-Networks-Struga-Qirici/78227a1267464c132236b0bf25a0db812788b864> (accessed on 1 January 2022).
- Verma, Sauraj. 2021. Forecasting volatility of crude oil futures using a GARCH–RNN hybrid approach. *Intelligent Systems in Accounting, Finance and Management* 28: 130–42. <https://doi.org/10.1002/isaf.1489>.
- Vidal, Andres, and Werner Kristjanpoller. 2020. Gold volatility prediction using a CNN–LSTM approach. *Expert Systems with Applications* 157: 113481. <https://doi.org/10.1016/j.eswa.2020.113481>.
- Vo, Nhi N. Y., Xue-Zhong He, Shaowu Liu, and Guandong Xu. 2019. Deep learning for decision making and the optimization of socially responsible investments and portfolio. *Decision Support Systems* 124: 113097. <http://doi.org/10.1016/j.dss.2019.113097>.
- Wellenreuther, Claudia, and Jan Voelzke. 2019. Speculation and volatility A time-varying approach applied on Chinese commodity futures markets. *Journal of Futures Markets* 39: 405–17. <https://doi.org/10.1002/fut.21984>.
- Zahid, Mamoon, Farhat Iqbal, Abdul Raziq, and Naveed Sheikh. 2022. Modeling and Forecasting the Realized Volatility of Bitcoin using Realized HAR–GARCH-type Models with Jumps and Inverse Leverage Effect. *Sains Malaysiana* 51: 929–42. <http://doi.org/10.17576/jsm-2022-5103-25>.
- Zhang, Zihao, Stefan Zohren, and Stephen Roberts. 2019. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing* 67: 3001–12. <https://doi.org/10.1109/TSP.2019.2907260>.