



## Can machine learning models better volatility forecasting? A combined method

Beining Han, Anqi Liu, Jing Chen & William Knottenbelt

**To cite this article:** Beining Han, Anqi Liu, Jing Chen & William Knottenbelt (14 Sep 2025): Can machine learning models better volatility forecasting? A combined method, The European Journal of Finance, DOI: [10.1080/1351847X.2025.2553053](https://doi.org/10.1080/1351847X.2025.2553053)

**To link to this article:** <https://doi.org/10.1080/1351847X.2025.2553053>



© 2025 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 14 Sep 2025.



[Submit your article to this journal](#)



Article views: 277






[View related articles](#)



[View Crossmark data](#)

# Can machine learning models better volatility forecasting? A combined method

Beining Han <sup>a</sup>, Anqi Liu <sup>a</sup>, Jing Chen <sup>a</sup> and William Knottenbelt<sup>b</sup>

<sup>a</sup>School of Mathematics, Cardiff University, Cardiff, United Kingdom; <sup>b</sup>Department of Computing, Imperial College London, London, United Kingdom

## ABSTRACT

Volatility forecasting for Bitcoin has garnered increasing attention due to heightened investment interest and the inherent risks associated with cryptocurrencies. Traditional forecasting models, such as the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) family models, are widely employed. However, there is a need for careful consideration regarding their ability to capture extreme shocks and the long-term volatile features. In this study, we fit several GARCH models, with the Exponential GARCH model demonstrating the best goodness of fit. We further utilise their volatility observations for an automated forecasting solution, using the Long Short-Term Memory (LSTM) neural network for predictions. Our results indicate a significant clear improvement in volatility forecasting regarding both the model's in-sample and out-of-sample accuracy. Notably, the LSTM model optimises information intake through its short- and long-memory states. Overall, our novel LSTM neural network model is more robust in responding to market shocks and regime changes.

## ARTICLE HISTORY

Received 24 July 2024  
Accepted 12 August 2025

## KEYWORDS

Bitcoin; volatility; forecasting; LSTM; GARCH

## 1. Introduction

Cryptocurrencies represent the most prominent application of decentralised blockchain technology, characterised by key advantages such as low entry barriers, transparency, and efficient transaction costs. In 2022, over 22,000 different 'coins' were traded, including inactive and discontinued ones.<sup>1</sup> As of December 2023, there are more than 9,000 active cryptocurrencies.<sup>2</sup> Some market intelligence predicts that overall capitalisation could reach USD 5.03 trillion by 2028, implying a compound annual growth rate of approximately 30.4%. In light of these, Bitcoin maintains its leading position with a market capitalisation of around USD 561.3 billion as of 2023.<sup>3</sup>

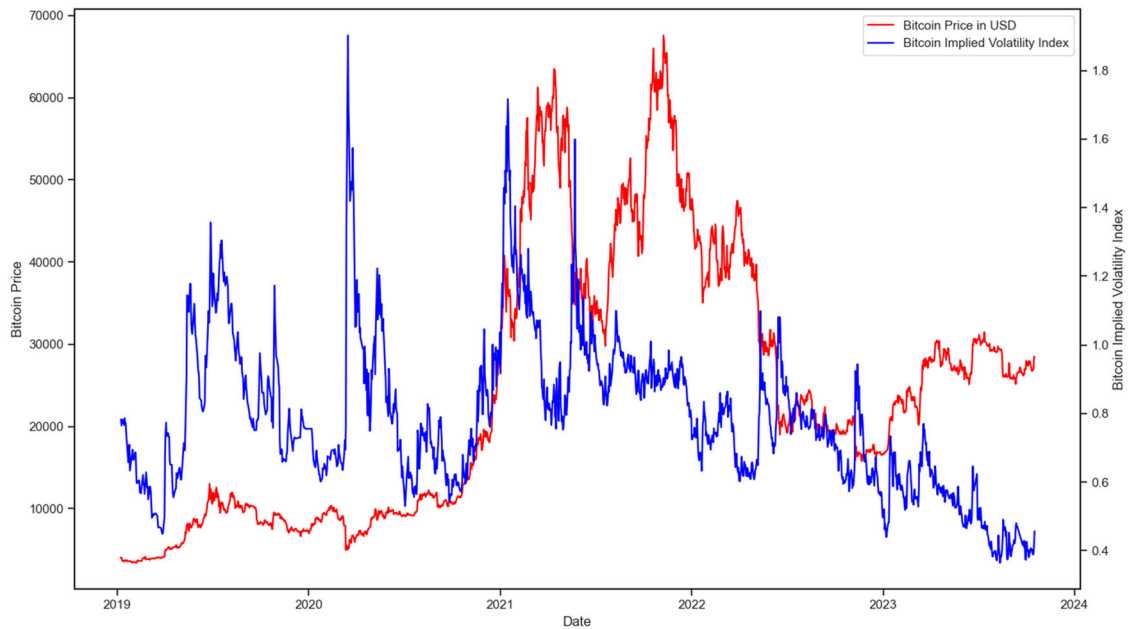
The Bitcoin market has proven to be highly speculative and volatile. Many professionals argue that Bitcoin, along with cryptocurrencies in general, is not isolated from the traditional banking system. For instance, crypto-focused banks, including Silvergate Bank, Silicon Valley Bank, and Signature Bank, were shut down following the bankruptcy of the crypto exchange FTX.<sup>4</sup> These failures have been linked to the banks' crypto holdings and have significantly distorted financial market stability. Consequently, establishing Bitcoin volatility forecasting models is a top priority for investors and provides substantial benefits to the economy.

Volatility forecasting and modelling has accumulated a rich literature based on Engle (1982), which introduces the concept of conditional heteroscedasticity and the general ARCH model. Bollerslev (1986) extended this with half autoregressive conditional heteroscedasticity, leading to the original GARCH model. Together with various extensions, the GARCH family models have become the most widely used technique to address three typical properties of financial time series: asymmetric extreme values, heavy tails, and volatility clustering (see Bollerslev 1986; Engle 1982; Glosten, Jagannathan, and Runkle 1993; Nelson 1991; Zakoian 1994). Efforts

**CONTACT** Jing Chen  chenj60@cardiff.ac.uk

© 2025 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

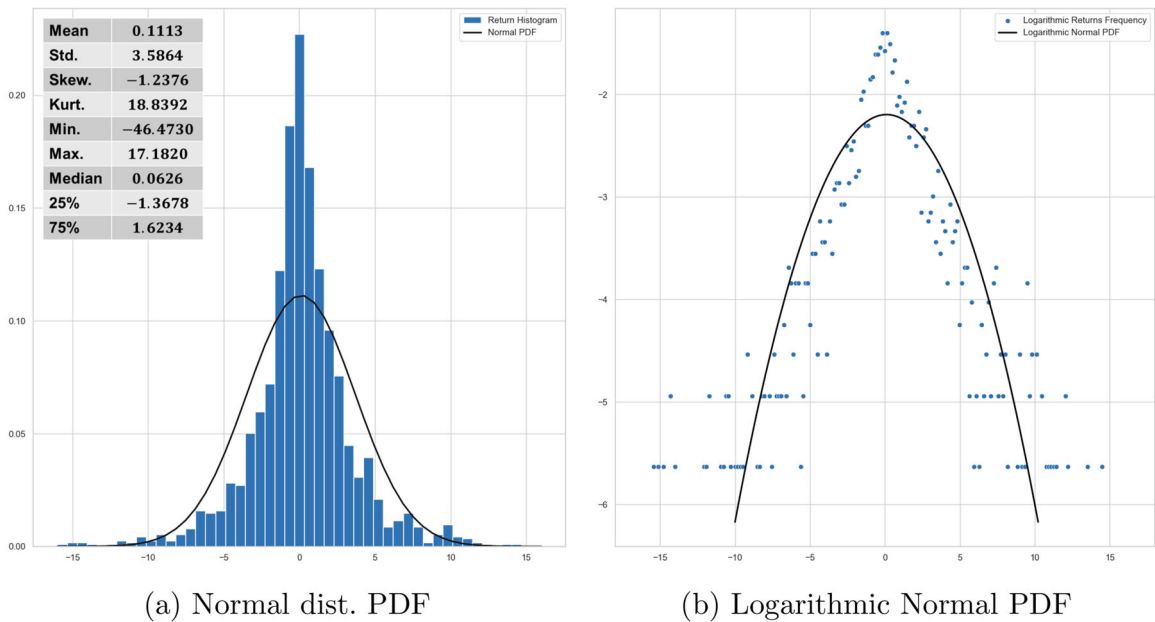
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.



**Figure 1.** Bitcoin prices and implied volatility index (2019–2023).

have also been made to improve realised volatility forecasting based on the GARCH framework by considering intra-day high-frequency price movements. For instance, Hansen, Huang, and Howan Shek (2012) proposed a realised GARCH model in which the Autoregressive Moving Average (ARMA) representation in the realised measure enables estimation of both conditional variance and intra-day realised volatility. Applications of this model to blue-chip stocks and SPDR S&P 500 ETF (SPY) demonstrated superior performance compared to previous GARCH models. Later, Hansen and Huang (2016) adopted the same idea to construct the realised Exponential GARCH model and showed that this technique improves volatility model fitting using eight different realised volatility measures, including the realised kernel, the daily range, and realised variance measures under various sampling frequencies of returns. However, due to microstructure noise, selecting reliable realised measures for such a framework can be critical and challenging, as documented in Andersen, Bollerslev, and Meddahi (2011); Hansen and Huang (2016).

In terms of the volatility forecasting for the Bitcoin market, studies using GARCH family models emerged rapidly. For example, Aras (2021); Dyhrberg (2016); Katsiampa (2017) tested more than 200 variants of GARCH models, and Bergsli et al. (2022) compared the performance of GARCH models with heterogeneous autoregressive (HAR) models, both suggesting that GARCH models are useful and efficient for forecasting Bitcoin volatility. However, as with applications in traditional financial markets, their strong performance is theoretically assured only when the time series do not contain extreme values, as Franses and Van Dijk (1996) suggested. Regular price jumps, bubbles, and crashes in Bitcoin reveal possible inaccuracies in volatility predictions made by GARCH models (see Figure 1). Although the Bitcoin return distribution appears symmetric, it clearly exhibits a high peak, indicating a non-normal, fat-tailed distribution. This is verified by hyperbolic tails in Figure 2(b) and implies a large kurtosis. These features suggest extreme values in the return time series. Nevertheless, GARCH remains valuable as a fundamental volatility modelling and forecasting method. In particular, similarity between the statistical properties of Bitcoin returns and those of classic financial asset returns motivate us to start with the GARCH framework. We also believe it is meaningful to thoroughly explore this forecasting technique before moving on to more complex models or additional modifications. Therefore, the objective of this study is to identify a group of GARCH models that best suit Bitcoin volatility forecasting and bring them together to deliver improved predictions.



**Figure 2.** Statistics of Bitcoin returns. (a) Normal dist. PDF (b) Logarithmic Normal PDF.

To address the challenges in prediction caused by heavy tails and extreme returns, we think deep learning algorithms should be useful for establishing forecasting models built upon the GARCH framework. Rapidly developing deep learning techniques inspired applications of neural networks and other learning models for financial market forecasting. Previous studies have explored hybrid forecasting frameworks such as GARCH-Support Vector Regression(SVR), GARCH-Artificial Neural Network(ANN), and GARCH-LSTM. For example, Li et al. (2009); Peng et al. (2018); Pérez-Cruz, Afonso-Rodriguez, and Giner (2003) replaced the autoregressive part with a SVR model to enhance the model fitting and prediction accuracy. However, these methods introduce a black box into the GARCH framework, which is generally disfavoured in academic finance. A more efficient approach is to use GARCH-type volatilities as inputs for model training (Fu 2023; Kim and Hyun Won 2018; Kristjanpoller and Minutolo 2016; Sun and Yu 2020). We find that SVR and ANN, supported by well-established theoretical foundations, have been extensively explored and have demonstrated practical applications. But in the context of modelling nuanced time series dynamics and achieving accurate predictions in financial markets, these algorithms are limited by their memory-less feature (Kristjanpoller and Minutolo 2016; Sun and Yu 2020).

In recent years, the Long Short-Term Memory (LSTM) architecture introduced by Hochreiter and Schmidhuber (1997) has gained widespread use for handling memory in time series modelling. This training technique is recognised for its capability to adapt to various memory lengths and decay rates, making it particularly useful for financial data. Similar to other neural networks and general deep learning architectures, it also facilitates learning from features that present different memory and statistical properties. For example, an LSTM model can be constructed using volatility estimates, trading volumes, gold prices, bond yields, and various other time series inputs (see Fu 2023; García-Medina and Aguayo-Moreno 2023; Kim and Hyun Won 2018; Wu et al. 2018). Hybrid models combining classic time series models (e.g. AR, ARIMA, GARCH) and LSTM have demonstrated superior performance in numerous financial forecasting studies, particularly in their ability to adapt to extreme return values (Bergsli et al. 2022; Caporale and Zekokh 2019; Katsiampa 2017; Wang, Andreeva, and Martin-Barragan 2023; Zahid, Iqbal, and Koutmos 2022). Some specific examples are the use of GARCH-LSTM to forecast prices, value-at-risks, volatilities, and portfolio risks across various financial markets, including the crypto market (AlMadany et al. 2024; Nsengiyumva, Mung'atu, and Ruranga 2025). For volatility predictions, although most of these studies are interested in realised volatility, Christensen and Prabhala (1998) argue that

implied volatility is a better indicator after addressing overlapping and high autocorrelation issues in the data sampling procedure (also see Canina and Figlewski 1993). We believe that implied volatility is especially important for Bitcoin, given its highly volatile price history and the relative lack of supportive indicators in its derivative markets. Hence, in this paper, we propose to focus on implied volatility forecasting for Bitcoin. This technique will also benefit Bitcoin option investors by assisting with pricing and hedging, as well as providing additional tools to enhance their investment strategies.

Leveraging deep learning architectures to design forecasting models is currently favoured, but it always involves challenges in finding reliable input features. In the examples mentioned above, model inputs typically involve various prices and volatility estimates. Sometimes multi-layered learning architecture is employed, while trade-offs between computational complexity and accuracy are consistently present. Choices can be arbitrary, as we see studies involving rather complex and higher-order models to showcase enhanced forecast accuracy (e.g. Gao, He, and Engin Kuruoglu 2021; Kim and Hyun Won 2018). The advantages and disadvantages of different models add challenges in designing an effective learning architecture. For example, range-based models (e.g. Garman and Klass 1980; Parkinson 1980) only offer effective volatility estimations when leveraging high-frequency data or at least intraday data. The same issues arise for realised variance calibrations (see Barndorff-Nielsen et al. 2011). In this study, we aim to avoid inputs that are overly data-intensive or require high liquidity and trading volumes, enabling broader applications in the cryptocurrency market. Hence, we choose to use GARCH models to provide volatility estimates as input features, which are then refined using LSTM. This is also a commonly used GARCH-LSTM hybrid framework which takes advantage of both the analytical strengths of GARCH-type models and the adaptability of advanced deep learning techniques. We also avoid using higher-order architectures that introduce greater computational complexity and time costs. Given the need for high-quality inputs in such a design, we conducted a careful analysis of GARCH model fitting to produce reliable input generation. The use of widely adopted GARCH-type models and a relatively simple model architecture ensures that our proposed model provides a foundation for future developments in both academia and industry.

To summarise, our study addresses two gaps in Bitcoin volatility forecasting. First, we design a model to predict implied volatility, particularly useful for Bitcoin derivative investments. Our forecasting target is the Bitcoin Implied Volatility Index (BitVol) index,<sup>5</sup> produced by T3 Index, and derived from the prices of tradable Bitcoin options. Second, we introduce a GARCH-LSTM hybrid model with straightforward GARCH-type inputs and manageable computational complexity, showing promise for further applications. We build two GARCH-LSTM models for 1-day and 5-day implied volatility forecasting, respectively. Both models outperform the GARCH family models for in-sample and out-of-sample performance. In particular, as evidenced by the out-of-sample testing dataset, we successfully reduced the prediction percentage error from over 10% (across all GARCH-type models) to 5.70% for the 1-day forecasting and 8.22% for the 5-day forecasting.

The literature has rapidly grown to combine a machine learning model, such as LSTM, with a traditional volatility estimation method like GARCH(1, 1) (Fu 2023; García-Medina and Aguayo-Moreno 2023; Kim and Hyun Won 2018; Wu et al. 2018). However, the common approach is to take a selected GARCH model estimation and plug it into a LSTM model to forecast volatility. This would have several issues on both GARCH and LSTM sides. For the former, there would be concerns of the right choice of a GARCH model that fits the empirical data; market extreme events and so on. For the latter, potential issues include input sensitivity, complexity of LSTM structure, interpretability, life span (shelf life), model accuracy, etc. To best mitigate these issues efficiently and effectively, we adopt this popular hybrid (GARCH-LSTM) structure but with our own innovative twists that bring multiple contributions to the current literature. First, it accounts for computational efficiency as both GARCH models and the LSTM are quick to compute and run. Second, forecasting accuracy is improved due to the reliable ‘preliminary’ forecasting results generated by well-fitted GARCH models, coupled with the proficiency of LSTM in handling sequential data. Third, our model is straightforward for traders to understand, accept, and adopt due to its use of predictions from GARCH-type models and the simplicity of a linear-like, single-layer LSTM neural network. More importantly, compared to previous studies that use endogenous indexes or GARCH parameters as LSTM inputs, our approach builds a stronger theoretical foundation and achieves a better balance between predictability and model complexity, delivering robust predictions with a single-layer LSTM model.

The rest of this paper is organised as follows: Section 2 provides an overview of the volatility research for Bitcoin. Sections 3 and 4 present the data and methods used. Section 5 shows the results, Section 6 discusses the model interpretations, and Section 7 draws the conclusion and discusses future research.

## 2. Literature review

As one of the most important problems in investment research, volatility forecasting has been extensively studied since 1990s (Canina and Figlewski 1993; Engle and Patton 2001). Literature suggests that volatility is technically more predictable than daily returns (Fassas and Siriopoulos 2021; McAleer and Medeiros 2008). GARCH family models are probably the most commonly used methods to estimate and forecast volatility due to their ability to cope with properties of financial data such as volatility clustering, mean-reversion, and asymmetric influence of returns. Alternatively, stochastic volatility (SV) models and heterogeneous autoregressive (HAR) models like the one seen in Corsi (2009); Taylor (2004) can be used. Since Engle (1982) proposed the ARCH model in 1982 and Bollerslev (1986) established the original GARCH model, a wide range of variants of this modelling framework have emerged, including the Glostten-Jagannathan-Runkle GARCH (GJR-GARCH) (Glostten, Jagannathan, and Runkle 1993), threshold GARCH(TGARCH) (Zakoian 1994), Exponential GARCH(EGARCH) (Nelson 1991) and so on, aiming to capture volatility clustering, heteroskedasticity, asymmetric shocks and leverage effects.

The growing interest in managing crypto investments risks has naturally popularised the applications of GARCH models into widespread use. For example, Katsiampa (2017) applies 11 types of GARCH models to estimate Bitcoin's volatility using a sample from 2010 to 2016 and finds that the Autoregressive-GARCH model fits the best. Caporale and Zekokh (2019) examine applying over 1000 GARCH-type models across four cryptocurrencies to generate one-step predictions for Value-at-Risk and Expected Shortfall. Volatility, by definition, is a 'hidden' measure, which is not directly observable. The two most popular volatility measures are implied volatility and realised volatility. Bergsli et al. (2022) compare prediction accuracy between GARCH-type and HAR models for realised volatility of Bitcoin and conclude that HAR models demonstrate superior forecast power over GARCH-type models. On the contrary, Hoang and Baur (2020) verify that GARCH predictions align better with Bitcoin implied volatility derived from option prices than HAR models. Christensen and Prabhala (1998); Hoang and Baur (2020) argue that the former is more insightful as it is derived from option prices and reflects the market's expectations in the near future. However, most existing literature on Bitcoin volatility focuses on realised volatility, as the Bitcoin options market is relatively new and its associated implied volatility index is just developed in recent years.

In general, relying solely on GARCH models does not yield strong performance. A modified or hybrid framework that integrates GARCH with asymmetry models or regime-switching techniques (e.g. Markov-Switching) appears more effective (for example, Ardia et al. 2018; Charles and Darné 2019; Haas, Mittnik, and Paoletta 2004). Moving forward, a growing number of studies have begun incorporating deep learning methods into volatility forecasting models. Aras (2021) suggests a hybrid approach that integrates the GARCH framework with Support Vector Machine after evaluating the forecasting performance of 110 different GARCH-type models. Similar techniques are applied by Kristjanpoller and Minutolo (2016); Sun and Yu (2020) who use GARCH model fitting results as inputs for SVR and ANN, respectively. Seo and Kim (2020) employ a more advanced Higher Order Neural Network (HONN) learning technique to train predictions based on past volatility and GARCH-type volatilities. Due to their memoryless nature, these learning algorithms are not well-suited for handling sequential data and long-term dependencies. This might explain why some of the aforementioned studies prefer using features such as model parameters and prediction errors, which exhibit less memory, as learning inputs. However, as Shen, Wan, and Leatham (2021) pointed out, constructing of learning inputs is crucial for advanced deep learning techniques to achieve effective model training. This argument is supported by the poorer performance of GARCH-recurrent neural network(RNN) models compared to a simple GARCH(1, 1) when realised volatility, derived from daily squared returns, and historical Garman-Klass volatility are used as inputs. Given the current advancements in deep learning algorithms, LSTM is undeniably the most effective approach to tackling this issue. For example, Fu (2023); García-Medina and Aguayo-Moreno (2023); Kim and



Hyun Won (2018); Zahid, Iqbal, and Koutmos (2022) investigate a range of large-scale, multi-layered GARCH-LSTM models and demonstrate their enhanced accuracy in predicting volatility compared to using GARCH models alone. In particular, García-Medina and Aguayo-Moreno (2023) develop a cryptocurrency portfolio strategy where volatility estimation relies on GARCH-LSTM, highlighting the potential of this hybrid framework for more complex applications in the cryptocurrency market. Zahid, Iqbal, and Koutmos (2022) demonstrate the success of integrating a well-fitted GARCH model into LSTM to generate realised volatility predictions; Amirshahi and Lahmiri. (2023) test across 27 cryptocurrencies using the GARCH-LSTM and the GARCH-Feed Forward Neural Networks(DFNN) hybrid frameworks, observing improved volatility prediction in both.

In cryptocurrency, other studies have applied a similar hybrid architecture of deep learning models to address market predictions, including price movements, market volatility, tail risks, sentiment impact, etc. Wu et al. (2018) combine AR and LSTM models to predict the Bitcoin price movements, followed by Gao, He, and Engin Kuruoglu (2021) who find that including GARCH volatility in LSTM model training helps improving Bitcoin price predictions. When Wang, Andreeva, and Martin-Barragan (2023) consider exogenous factors such as the US daily news index, they conclude that a similar hybrid model outperforms traditional forecasting methods.

These studies provide conceptual design and important modelling development of hybrid frameworks, such as GARCH-LSTM. However, we think several fundamental challenges remain unresolved. Existing studies show that multi-layered architectures tend to overfit when the market is highly volatile and exposed to frequent regime shifts, raising concerns about their robustness in real-world applications (Seo and Kim 2020). This point is also supported by Ardia et al. (2018); Haas, Mittnik, and Paoletta (2004), which observe a rapid decline in the forecasting accuracy of such complex hybrid models when market conditions change. Several comparative studies (Bergsli et al. 2022; Franses and Van Dijk 1996; Hoang and Baur 2020) have found that traditional GARCH models can outperform more complex, yet poorly specified, alternatives in highly volatile environments, such as the cryptocurrency market. Additionally, many complex deep learning models require large-scale datasets and intricate hyperparameter tuning, which can be impractical or infeasible for practitioners dealing with fast-evolving markets and limited computational resources (Shen, Wan, and Leatham 2021).

In response to these unsolved issues in hybrid deep learning design, we propose a GARCH-LSTM architecture suited for practical applications. We bridge the gap in current hybrid model designs through two distinct innovations. The first involves screening, selecting, and calibrating a set of GARCH-type models with a more appropriate distribution (e.g.  $t$ -distribution vs. Normal) to ensure optimal input quality. The second is the use of a single-layer architecture to reduce computational intensity and the need for large datasets. It would also have better interpretability and shelf life. Regarding literature contribution, this study establishes a foundation for improving forecast accuracy through the from GARCH-type models to GARCH-LSTM hybrids. Our proposed single-layer architecture provides a solid baseline for the integration of traditional models with deep learning techniques. Moreover, as the Bitcoin options market is still developing, research on implied volatility in the Bitcoin market is still limited, with only a few studies addressing valuation (Hoang and Baur 2020; Zulfiqar and Gulzar 2021). This paper adds to the growing literature on this emerging field.

### 3. Methodology

We explore the implementation and performance of implied volatility forecasting using GARCH family models. To enhance the forecasting results, we build a Long Short-Term Memory (LSTM) neural network model in which the predictions from GARCH family models and historical implied volatility data are set as features.

#### 3.1. GARCH family models

The GARCH family models are constructed to introduce mathematical techniques that deal with stochastic volatility:

$$\varepsilon_t = \sigma_t Z_t, \quad (1)$$

where  $\varepsilon_t$  represents the return residuals,  $\sigma_t$  denotes the stochastic volatility term, and  $Z_t$  is the noise term with mean zero, variance one. The innovation distribution of  $Z_t$  is usually selected from a normal distribution or a Student's t distribution.

The standard GARCH model is defined by Bollerslev (1986). See below for GARCH(1, 1):

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2, \\ \omega &> 0, \quad \alpha > 0, \quad \beta > 0.\end{aligned}\tag{2}$$

Glosten, Jagannathan, and Runkle (1993) argue that volatility responds to positive and negative residuals at varying levels. Hence, they propose the Glosten-Jagannathan-Runkle GARCH (GJR-GARCH) model by adding a term for negative residuals. We employ the GJR-GARCH(1, 1) version in this paper:

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha \varepsilon_{t-1}^2 + \gamma \varepsilon_{t-1}^2 \mathbb{1}_{\{\varepsilon_{t-1} < 0\}} + \beta \sigma_{t-1}^2, \\ \omega &> 0, \quad \alpha > 0, \quad \beta > 0, \quad \gamma \in \mathbb{R},\end{aligned}\tag{3}$$

where  $\mathbb{1}_A$  is an indicator function such that  $\mathbb{1}_A = 1$  if  $A$  is satisfied, otherwise  $\mathbb{1}_A = 0$ . The GJR-GARCH model reduces to the GARCH model if  $\gamma$  is zero.

To model the asymmetric volatility responses, Zakoian (1994) introduces an alternative method to model the asymmetric responses of positive and negative return shocks, which is the Threshold GARCH (TGARCH) model. In this model, volatility responds to absolute residuals rather than squared residuals, as shown in Equation (4) for TGARCH(1, 1).

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha |\varepsilon_{t-1}| + \gamma |\varepsilon_{t-1}| \mathbb{1}_{\{\varepsilon_{t-1} < 0\}} + \beta \sigma_{t-1}^2, \\ \omega &> 0, \quad \alpha > 0, \quad \beta > 0, \quad \gamma \in \mathbb{R}.\end{aligned}\tag{4}$$

The Exponential GARCH (EGARCH) proposed by Nelson (1991) models the logarithmic variance. This model ensures that the variance remains non-negative. In this paper, we use the symmetric version of EGARCH(1, 1):

$$\begin{aligned}\ln \sigma_t^2 &= \omega + \alpha (|Z_{t-1}| - \mathbb{E}|Z_{t-1}|) + \beta \ln \sigma_{t-1}^2, \\ \omega &> 0, \quad \alpha > 0, \quad \beta > 0.\end{aligned}\tag{5}$$

### 3.2. Long short-term memory (LSTM) neural network model

While GARCH family models provide valuable volatility forecasts, gaps and flaws still require addressing. Observing the variations in the models above, we conclude that no universal form that suits every situation. This presents the first challenge when applying these classic models to a new market like Bitcoin. Moreover, even though we select a model using mathematical techniques, the return time series properties may change over time as the market evolves. To address these issues and build an accurate and sustainable volatility forecasting model, we employ a neural network, called LSTM, to combine outcomes from different GARCH models.

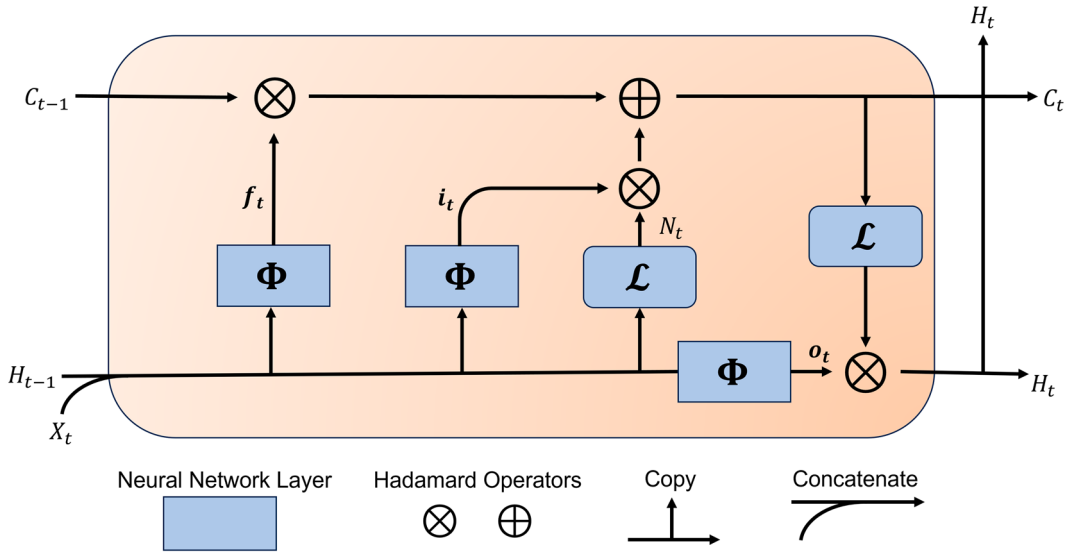
The LSTM model is a type of recurrent neural network (RNN) designed explicitly for sequential data inputs. RNNs are distinguished by their 'memory' – they take information from prior inputs to current nodes, making them effective at dealing with temporal problems. LSTM enhances basic RNNs by involving a forget gate, which discards 'outdated' information from prior nodes (see Figure 3).

In the LSTM, an input  $X_{T \times N}$  is constructed by  $T$  look-back timesteps and  $N$  features. On each timestep, the LSTM cell generates a hidden state  $(H_t)_{1 \times m}$ , where  $m$  is the number of units in the LSTM layer. By processing  $X_t$  from the earliest to the latest timesteps, information flows through the cell state  $\{(C_t)_{1 \times m} : t = 1, 2, \dots, T\}$ . From prior time  $t-1$  to  $t$ , the current cell state takes information from the previous cell state and new input:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes N_t,\tag{6}$$

where  $\otimes$  denotes the Hadamard product,  $f_t \in [0, 1]^m$  is the forget gate such that  $f_t = 0$  means removing everything in  $C_{t-1}$ ,  $N_t$  is new information given by new input and memory carried from previous hidden state,





**Figure 3.** LSTM neural network cell.

$i_t \in [0, 1]^m$  is the input gate such that  $i_t = 1$  means taking all information in  $N_t$ . The new information carries the previous hidden state  $H_{t-1}$  and the new input  $X_t$ :

$$N_t = \mathcal{L}(X_t \cdot u_c + H_{t-1} \cdot w_c + b_c), \quad (7)$$

where  $(u_c)_{N \times m}$  and  $(w_c)_{m \times m}$  are weights of the current input and the previous hidden state, respectively,  $(b_c)_{1 \times m}$  is a bias parameter. The activation function  $\mathcal{L}(x)$  introduces a scaling of the addition and subtraction of information. Now, we focus on the most important part in the LSTM, the output  $H_t$ . It is given by an output gate  $o_t \in [0, 1]^m$  and the cell state  $C_t$ :

$$H_t = o_t \otimes \mathcal{L}(C_t). \quad (8)$$

If  $o_t = 1$ , the LSTM cell sends out all information in the cell state as a prediction; otherwise, ‘discounted’ information is used as a forecast result. Note that the same activation function should be used for the new information  $N_t$  and the output  $H_t$ .

The forget, input and output gates all consider both the new input  $X_t$  and the prior output  $H_{t-1}$ :

$$f_t = \Phi(X_t \cdot u_f + H_{t-1} \cdot w_f + b_f), \quad (9)$$

$$i_t = \Phi(X_t \cdot u_i + H_{t-1} \cdot w_i + b_i), \quad (10)$$

$$o_t = \Phi(X_t \cdot u_o + H_{t-1} \cdot w_o + b_o), \quad (11)$$

where  $(u_f)_{N \times m}$ ,  $(w_f)_{m \times m}$ ,  $(u_i)_{N \times m}$ ,  $(w_i)_{m \times m}$ ,  $(u_o)_{N \times m}$  and  $(w_o)_{m \times m}$  are weights;  $(b_f)_{1 \times m}$ ,  $(b_i)_{1 \times m}$ ,  $(b_o)_{1 \times m}$  are bias parameters; and  $\Phi(\cdot)$  is the Sigmoid function below:

$$\Phi(x) = \frac{1}{1 + \exp(-x)}. \quad (12)$$

The advantage of LSTM lies in its ability to learn how much old information should be stored in the long memory state  $C_t$ , and what can be ignored. Hence, the hidden state  $H_t$  indicates short memory. The structure of the LSTM cell is shown in Figure 3.

We utilise a simple LSTM neural network architecture. The sequential input  $X_t$  is constructed from a 30-day look-back period of five features: 1-day forecasts given by GARCH(1, 1), GJR-GARCH(1, 1), TGARCH(1, 1)

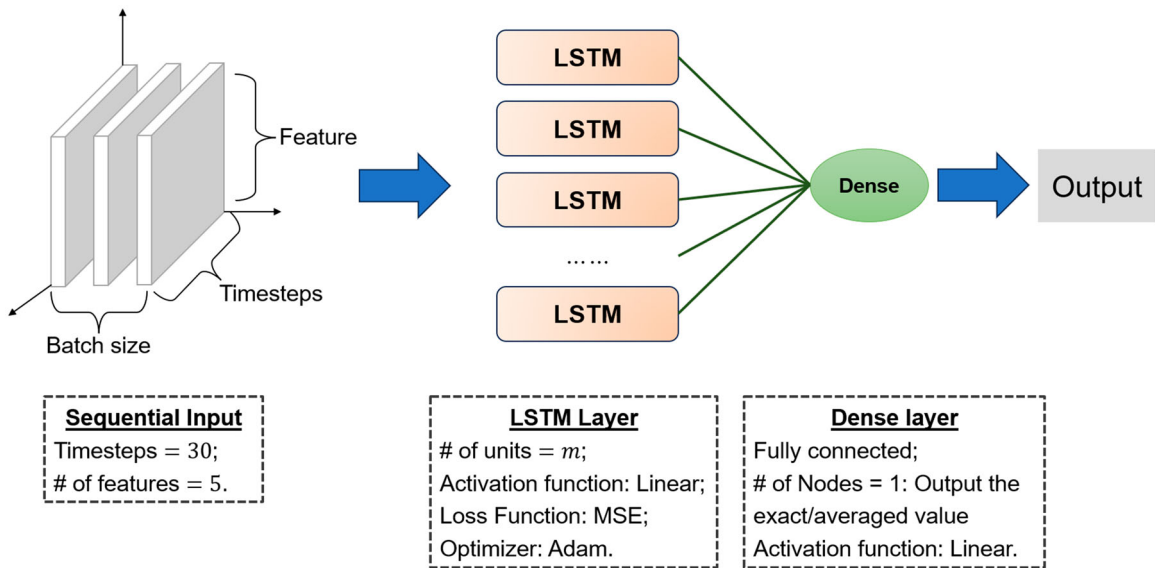


Figure 4. LSTM neural network architecture.

and EGARCH(1, 1), along with past Bitcoin implied volatility (BitVol) values from  $t-30$  to  $t-1$ . The target  $Y_t$  is the BitVol at  $t$ . Other technical specifics are summarised in Figure 4. Here, it is worth noting that we use a linear activation function, instead of the common non-linear activators like tanh and ReLU. This is because all our input features are on the same scale as the target and should not deviate significantly from the target. Thus, training will be more efficient without scaling or shape changes. After comprehensive empirical testing, a 30-day look-back period was selected to determine the optimal length for capturing sufficient temporal information while avoiding excessive noise. We tested several look-back periods, including 7, 14, 30, and 60 days (the results are shown in Appendix), to evaluate their impact on model accuracy and training stability. Among these, the 30-day period consistently provided the best results, balancing the trade-off between capturing relevant volatility patterns and avoiding redundancy or overfitting. Shorter look-back periods, such as 7 and 14 days, lacked sufficient temporal information to identify key trends, resulting in underfitting and limited predictive performance. In contrast, longer periods, such as 60 days, introduced excessive noise and reduced the model's responsiveness to recent market dynamics. Furthermore, the 30-day look-back period aligns naturally with the target variable in our model. Since the implied volatility is derived from 30-day option prices, ensuring that the input features effectively correspond to the output data enhances the model's predictive performance.

The LSTM model optimisation followed a two-step process:

- (1) *Random Search*: We first conducted a random search to explore general hyperparameter ranges and efficiently identify promising configurations.
- (2) *Grid Search*: Using the ranges determined in the random search, we systematically evaluated all possible hyperparameter combinations through grid search. This approach allowed us to select the optimal configuration based on model performance on the validation set.

Table 1 summarises the final hyperparameters chosen for the LSTM model and the ranges tested during tuning.

- *Number of Units*: A total of 8 units provided an optimal trade-off between computational efficiency and the ability to capture complex temporal patterns.

**Table 1.** LSTM hyperparameter tuning results.

Hyperparameter	Optimal value	Tested Range
Number of Units	8	8, 16, 32, 64
Batch Size	32	8, 16, 32, 64
Validation Split	0.2	0.2, 0.15, 0.25
Early Stopping Patience	10	5, 10, 15
Epochs	200	50 to 300
Layers	1	1, 2, 3, 4, 5
Time Steps	30	7, 14, 30, 60
Learning Rate	0.01	0.001, 0.01, 0.05
Dropout Rate	0.2	0.2, 0.3, 0.5

- *Batch Size*: A batch size of 32 balanced training speed and performance, avoiding convergence issues associated with larger or smaller sizes.
- *Validation Split*: A 20% validation split ensured sufficient data for both training and validation while mitigating overfitting.
- *Early Stopping Patience*: A patience value of 10 epochs prevented unnecessary training while ensuring convergence.
- *Epochs*: Training for up to 200 epochs allowed sufficient passes through the data while avoiding overfitting due to early stopping.
- *Layers*: A single-layer LSTM architecture balanced simplicity, efficiency, and performance.
- *Time Steps*: A look-back period of 30 time steps provided sufficient historical context without introducing noise.
- *Learning Rate*: A learning rate of 0.01 ensured stable and efficient weight updates.
- *Dropout Rate*: A 20% dropout rate improved generalisation by reducing overfitting.

We employed a random search strategy to identify suitable training parameters and used the grid search method to derive the optimal hyperparameters.

### 3.3. Performance metrics

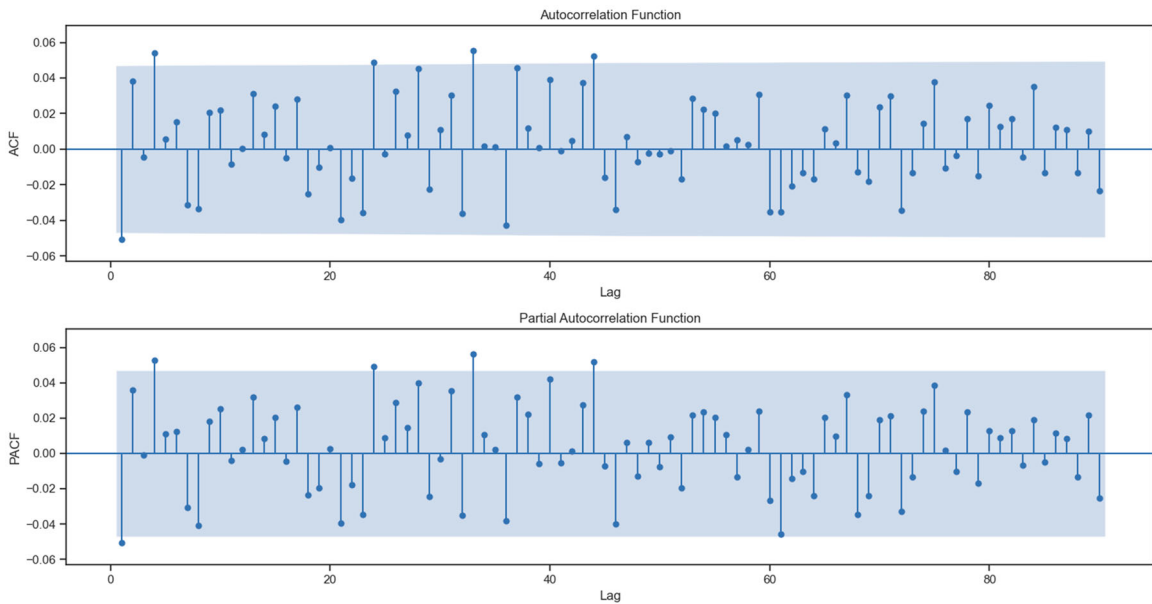
To examine the forecasting results of the models introduced in previous sections, we utilise the performance metrics in Equations (13) and (14), respectively. Recall that we use Mean Squared Error (MSE) as the loss function, which is the square Root of Mean Squared Error (RMSE). Hence, RMSE is theoretically minimised when training the model.

$$\text{RMSPE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{Y_i - \hat{Y}_i}{\hat{Y}_i} \right)^2} \quad (13)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (14)$$

where  $\{Y_i : i = 1, 2, \dots, n\}$  represents the forecast values and  $\{\hat{Y}_i : i = 1, 2, \dots, n\}$  denotes the target values.

Root Mean Squared Percentage Error (RMSPE) would be more effective than other measures in capturing the degree of errors in relation to the desired target values. Additionally, RMSPE would penalise significant errors more severely than standard RMSE, which this paper requires. Therefore, RMSPE takes precedence in this paper when comparing the performance of model predictions.



**Figure 5.** ACF and PACF of returns.

#### 4. Data

Bitcoin daily prices are obtained from Yahoo Finance.<sup>6</sup> We choose the Bitcoin Implied Volatility (BitVol) index as the outcome for our targeted volatility predictions. BitVol, produced by the T3 Index,<sup>7</sup> is a daily updated index tracking expected 30-day implied volatility in Bitcoin. Our data spans from 7 January 2019 to 17 October 2023, in total 1745 trading days. The Bitcoin market opens 24/7 and does not close on weekends or public holidays. The Bitcoin daily closing price in U.S. dollars is recorded in the GMT zone.

The BitVol index is sourced from the T3 Index. We split the dataset into training and test data using an 80–20 ratio. The training period runs from 7 January 2019 to 1 November 2022, comprising 1395 observations. The testing period is from 2 November 2022 to 16 October 2023, with 349 observations. Note that the training period for the LSTM is 30 days shorter, as we take sequential inputs with 30 timesteps.

The BitVol index contains missing data on some weekends and holidays before June 2020. The missing values constitute a small portion of the entire dataset (i.e. 5.57%) and are only present in the first half of our training dataset. We expect them to have minimal impact on our model training. Therefore, we adopt a simple and effective method – linear interpolation – to fill in the missing values. The completed BitVol index series is used as the target. This treatment is also based on the assumption that market perception of volatility evolves sequentially during periods when options are not tradable.

To ensure consistent data input for the LSTM model, we used conditional variance from GARCH models and the BitVol index, which are on the same scale. The data was converted into a percentage format to avoid issues caused by differing scales during model training.

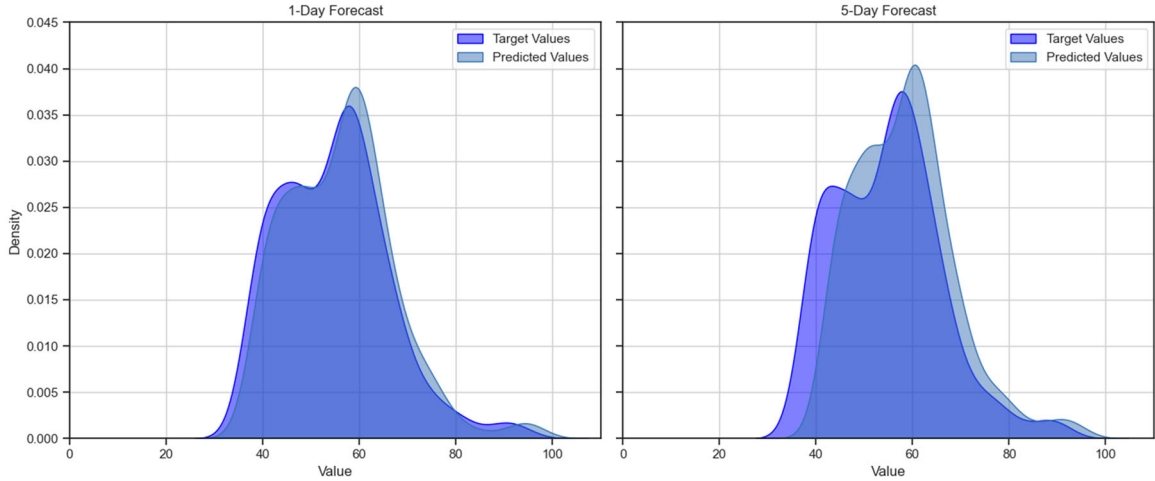
#### 5. Results

We first determine the mean process for return residuals  $\varepsilon_t$ . We examine the autocorrelation function (ACF) and partial autocorrelation function (PACF) in Figure 5, confirming no autocorrelations in returns and return residuals. Hence, we find return residuals through demeaning:  $\varepsilon_t = x_t - \mu$ , where  $\mu$  is the average return in the training dataset.

The calibration results of GARCH family models, and the degrees of freedom of the Student-T innovation distribution of each model are presented in Table 2. We use these models to conduct 1-day volatility forecasts

**Table 2.** Calibration results of GARCH family models.

Model	$\omega$	$\alpha$	$\beta$	$\gamma$	$\nu$
GARCH	0.236	0.070	0.930		2.0961
GJR-GARCH	0.191	0.077	0.937	−0.028	2.994
TGARCH	0.095	0.079	0.927	−0.012	3.154
EGARCH	0.066	0.166	0.988		2.805

**Figure 6.** Comparison of target and predicted value distributions.

and 5-day average volatility forecasts (i.e.  $\frac{\sum_{i=1}^5 \sigma_{t+i}}{5}$ ). The 1-day ahead volatility  $\sigma_{t+1}$  is computed using  $\varepsilon_t$  and  $\sigma_t$  which is known at  $t$ . However, volatility from 2 days ahead will require non-observable noise terms. Two techniques are used to address this issue: one is the simulation technique that draws  $Z_t$  from corresponding innovation distributions; the other is the bootstrap technique that draws noise terms from random sampling based on the training dataset. Although these techniques can yield further volatility forecasting, we do not expect either to produce a reliable result in the long-run. Hence, we limit the experiment to a short term of 5 days. It is worth noting that bootstrap results are not available for the training dataset due to the lack of random samples.

The GARCH family models exhibited poor performance in some cases due to their inability to adapt to sudden shifts in volatility, particularly during high-stress market periods. The reliance of GARCH models on linear relationships and past variance often leads to underestimation or overestimation of volatility during sharp market movements. The LSTM model, however, demonstrated significant improvements, especially in long-term (5-day) forecasts. This improvement can be attributed to the LSTM's ability to capture short-term dependencies through its gate mechanism and long-term memory states. These features enable it to retain relevant recent information from historic volatility while linking it within extended volatility patterns from GARCH models. This dual capability allows the model to dynamically adjust to evolving market conditions, unlike the static assumptions of GARCH models.

To further justify the superiority of the GARCH-LSTM model beyond traditional loss function metrics, we compared the distributions of the predicted and target volatility values. Figure 6 illustrates the density estimates for 1-day and 5-day forecast horizons. In the 1-day forecast, the distribution of the predicted values aligns closely with the target distribution, effectively capturing both skewness and kurtosis. This suggests that the model is well-suited for short-term volatility predictions and retains the ability to represent the underlying data accurately. In the 5-day forecast (right panel), the predicted distribution also matches the target values with slightly increased deviations, which are expected due to the compounding uncertainties in large forecast horizons. Nevertheless, the overlap between the predicted and target distributions remains substantial, highlighting the robustness of the model even in extended forecasts. This analysis demonstrates that the GARCH-LSTM model

**Table 3.** 1-day implied volatility forecast performance.

	Training dataset		Testing dataset	
	RMSPE	RMSE	RMSPE	RMSE
GARCH	16.22%	15.86	12.71%	6.96
GJR-GARCH	16.01%	14.88	10.90%	6.02
TGARCH	16.85%	15.26	10.51%	5.83
EGARCH	14.30%	12.61	12.49%	7.28
LSTM	5.13%	5.08	5.70%	3.13

**Table 4.** 5-day average implied volatility forecast performance.

	Training dataset		Testing dataset			
	RMSPE	RMSE	RMSPE		RMSE	
			Sim.	Boot.	Sim.	Boot.
GARCH	15.79%	15.67	14.76%	14.35%	8.09	7.91
	$(8.63 \times 10^{-3})$	$(6.83 \times 10^{-1})$	$(5.73 \times 10^{-3})$	$(3.62 \times 10^{-4})$	$(3.35 \times 10^{-1})$	$(2.30 \times 10^{-2})$
GJR-GARCH	15.65%	14.68	12.51%	12.24%	6.99	6.88
	$(7.14 \times 10^{-3})$	$(5.15 \times 10^{-1})$	$(8.61 \times 10^{-3})$	$(2.86 \times 10^{-4})$	$(4.70 \times 10^{-1})$	$(1.70 \times 10^{-2})$
TGARCH	16.49%	15.18	11.22%	11.17%	6.19	6.18
	$(1.46 \times 10^{-4})$	$(9.12 \times 10^{-3})$	$(1.35 \times 10^{-3})$	$(1.24 \times 10^{-4})$	$(7.16 \times 10^{-2})$	$(6.75 \times 10^{-3})$
EGARCH	188791%	$1.42 \times 10^5$	1518.46%	13.27%	$8.53 \times 10^2$	7.97
	$(1.26 \times 10^{108})$	$(1.03 \times 10^{110})$	$(1.83 \times 10^{28})$	$(2.04 \times 10^{-4})$	$(8.45 \times 10^{29})$	$(1.32 \times 10^{-2})$
LSTM	6.46%	6.22	8.22%		4.47	

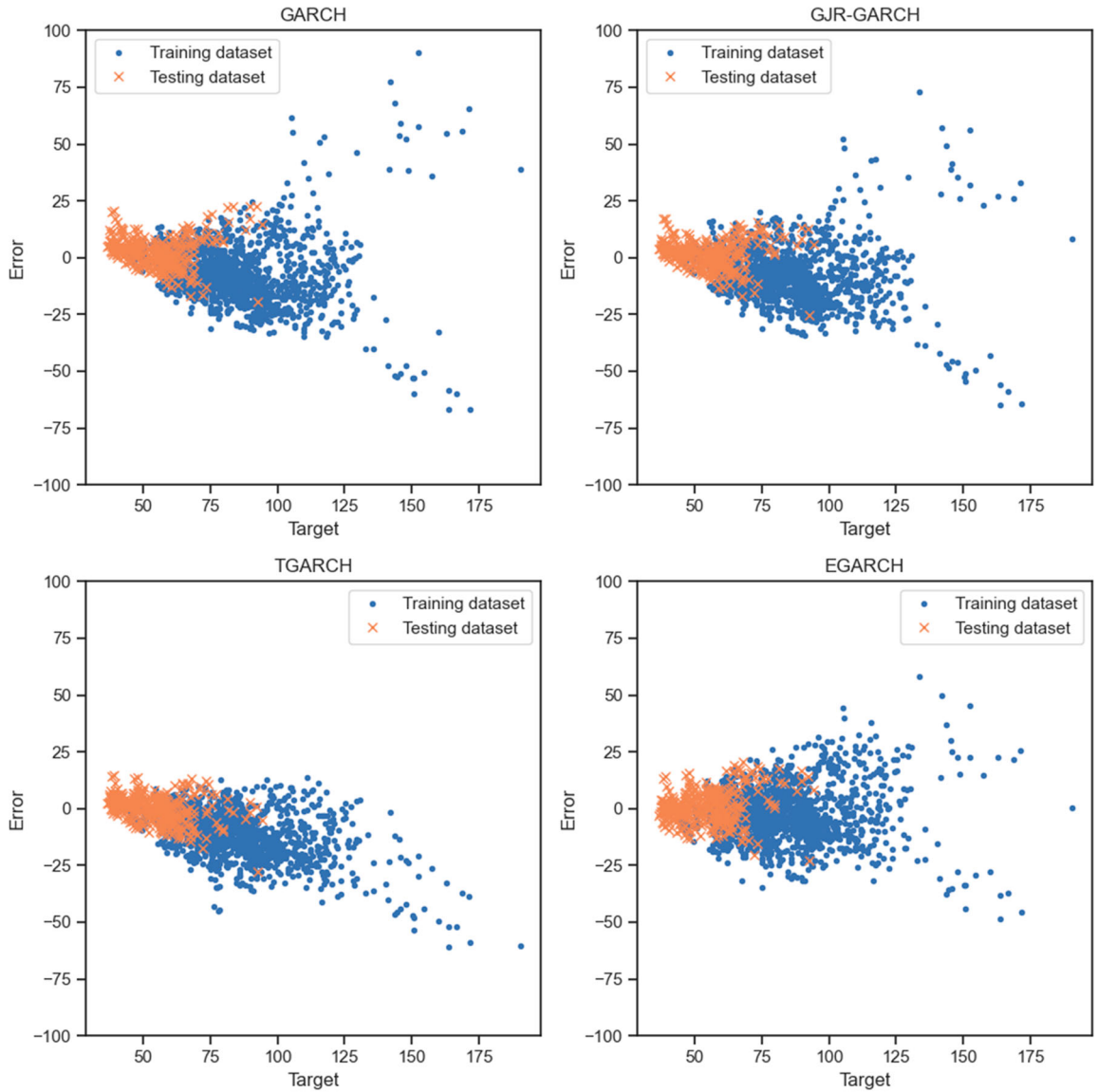
Note: The values in the table are the median and standard deviation of performance metrics, with the latter in parentheses.

not only achieves lower error metrics but also accurately reproduces the distributional properties of the target data. This distributional fit further supports the model's ability to effectively capture the underlying dynamics of market volatility, making it a reliable tool for financial forecasting.

The performance metrics are in Tables 3 and 4. For the 5-day forecasting, we carry out 1000 runs to examine the robustness of the methods. The values in Table 4 represent the median and standard deviation of performance metrics, with the latter in parentheses. We do not find clear differences in 1-day forecasting using these GARCH-type models. EGARCH performs best for the training dataset, but does not stand out in the testing stage. We observed an unusual result where all models exhibit much better performance in testing, particularly with regard to the RMSE metric. In Figure 7, we examine the error  $Y_i - \hat{Y}_i$  for each target  $\hat{Y}_i$  and confirm that the 'better' performance results from coincidentally avoiding high volatility levels during the testing period. All four models 'fail' when volatility exceeds 120(%), with GARCH, GJR-GARCH and EGARCH either over- or under-predicting the volatility. TGARCH exhibits an even more harmful systemic error, primarily producing under-predictions; moreover, as volatility increases, so does the bias. We believe the same explanation applies to the performance of 5-day forecasting, although we cannot observe it due to the unstable outcomes resulting from the simulation method. Another important observation is that, in the 5-day forecasting given by simulation methods, EGARCH becomes unstable, the instability in multi-step EGARCH forecasts is primarily driven by the interaction between the model's high persistence ( $\beta \approx 1$ ) and moderate shock sensitivity ( $\alpha$ ), combined with the random sampling of residuals from a heavy-tailed Student's *t* distribution. Extreme values drawn during simulation lead to exponentially amplified volatility due to the EGARCH model's logarithmic variance formulation. This effect compounds over 5 steps of forecasting, resulting in significant deviations. In contrast, bootstrap-based forecasting methods and 1-day forecasts avoid this instability. Bootstrap methods resample historical residuals, limiting extreme values to those observed in historical data and preventing the introduction of artificial outliers. The 1-day forecasting results do not suffer from compounding effects, as they involve only a single residual draw.

To further validate the forecasting performance, we also assess the statistical differences between models using the Diebold-Mariano (DM) test (Diebold and Mariano 1995; Harvey, Leybourne, and Newbold 1997).





**Figure 7.** GARCH family 1-day errors vs. Implied volatility.

The Diebold-Mariano test evaluates the null hypothesis that two forecasting models have equal predictive accuracy, based on a user-specified loss function (here, the mean squared error, MSE). For two sets of forecast errors, the DM test constructs the loss differential series  $d_t = L(e_{A,t}) - L(e_{B,t})$ , where  $L(\cdot)$  denotes the loss function (MSE in this study), and  $e_{A,t}$  and  $e_{B,t}$  are the forecast errors from models  $A$  and  $B$  at time  $t$ , respectively.

The DM test statistic is given by:

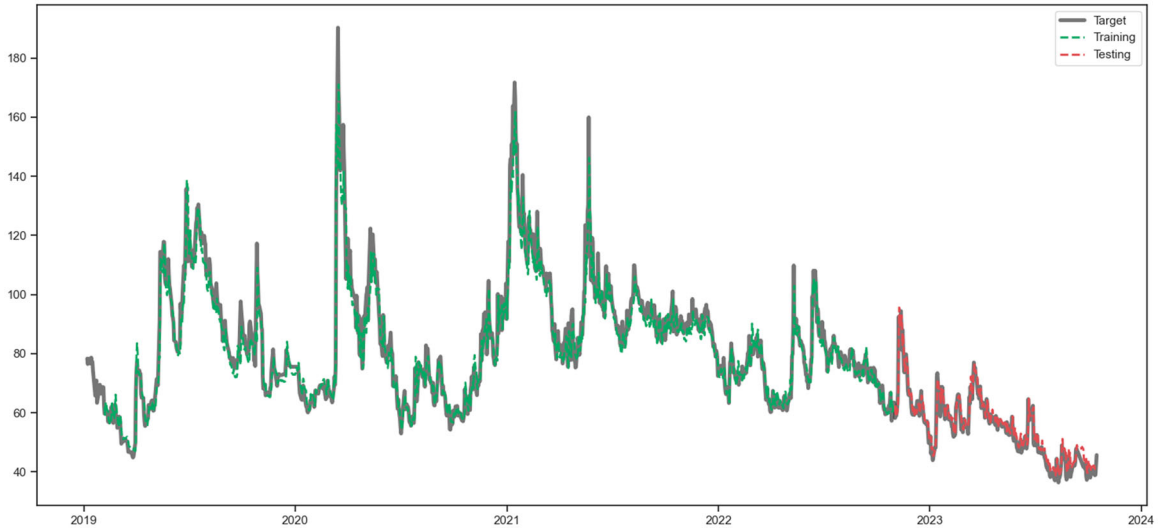
$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{f}_d(0)}{T}}}, \quad (15)$$

where  $\bar{d}$  is the mean of the loss differential series,  $T$  is the sample size, and  $2\pi\hat{f}_d(0)$  is a consistent estimate of the spectral density of  $d_t$  at frequency zero, adjusted for autocorrelation as proposed by Harvey, Leybourne, and Newbold (1997).

**Table 5.** Diebold–Mariano test results.

Model	1-Day		5-Day	
	<i>In-sample</i>	<i>Out-of-Sample</i>	<i>In-sample</i>	<i>Out-of-Sample</i>
GARCH	5.41*** (3.21e−08)	6.92*** (2.07e−11)	6.12*** (4.91e−09)	5.76*** (1.83e−08)
GJR-GARCH	5.58*** (1.76e−08)	7.00*** (1.25e−11)	5.21*** (2.35e−07)	4.63*** (4.99e−06)
EGARCH	6.03*** (4.95e−09)	8.62*** (2.31e−16)	6.45*** (1.08e−09)	5.99*** (5.33e−09)
TGARCH	5.77*** (9.80e−09)	7.85*** (5.22e−14)	3.11*** (1.85e−04)	1.94* (5.30e−02)

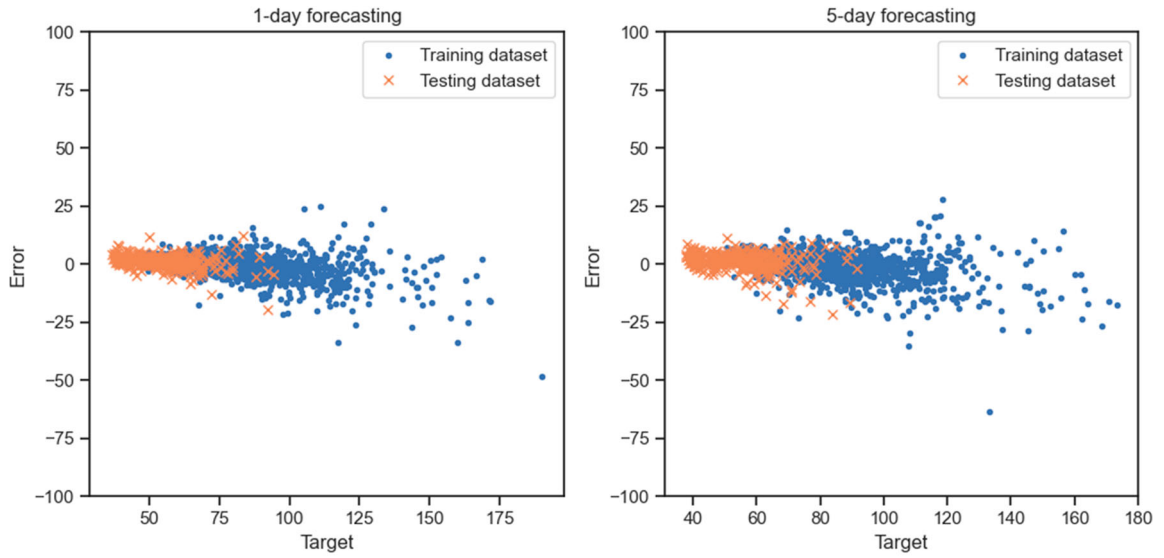
Note: This table reports both DM statistics and their *p*-values to demonstrate both in-sample and out-of-sample forecasting performance at 1-day and 5-day levels. \*, \*\*, and \*\*\* denote significance at the 10%, 5%, and 1% levels, respectively.

**Figure 8.** LSTM 1-day implied volatility forecasting.

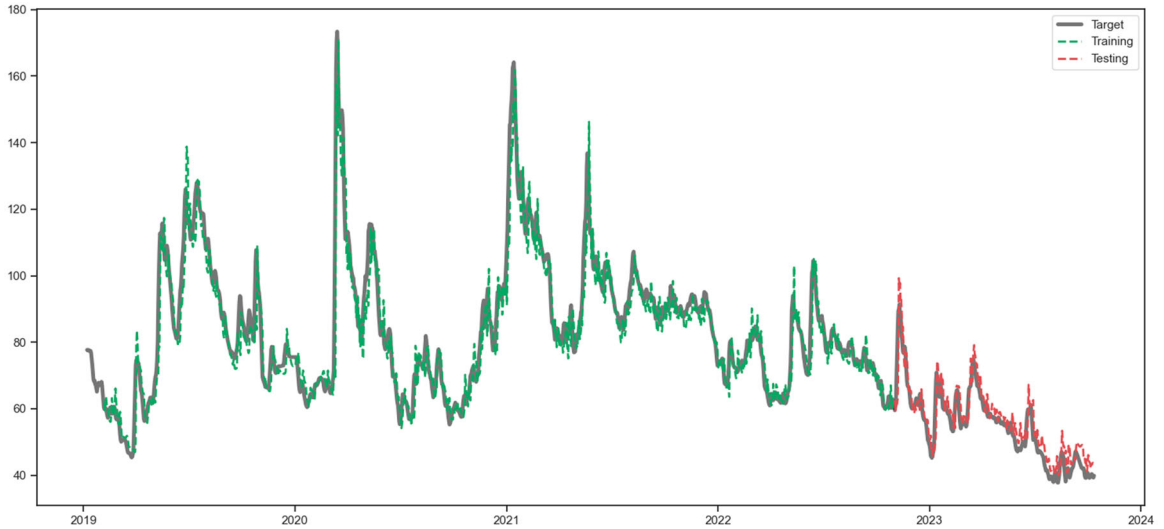
We compare the four fitted GARCH models to our hybrid models in pairs and for both 1- and 5-day forecasts. Table 5 reports the DM statistics and their *p*-values for both in-sample and out-of-sample forecasting at 1-day and 5-day levels.

A higher positive value of the DM statistic indicates a higher forecast error variance of the tested model. We obtain positive and significant DM statistics across all pairwise comparisons, which suggests that our LSTM-based hybrid model outperforms all four GARCH models with better forecast accuracy. In addition, the significance levels of DM statistics are strong (at 1% level) in most cases, further indicating that the hybrid model is consistently better than the GARCH models in forecasting Bitcoin volatility.

In comparison, we also investigate the errors produced by the LSTM model and GARCH-type models (see Figures 7 and 9). The deep learning technique effectively eliminates systemic errors introduced in the inputs and ensures a more accurate prediction close to the target values. In Figures 8 and 10, we find that, overall, the forecasting provided by the LSTM models is accurate. While it is challenging to train a 5-day forecasting model as effectively as the 1-day model. Lacking updated information, the model shows signs of deviating from the target shortly within a year of testing. Also, the model may fail to catch up with the market fluctuations even in the short term. For example, we observed some unstable predictions in 2021 due to abrupt market changes triggered by external events, which were not well represented in the training dataset. To mitigate such occurrences, expanding the training dataset to include more extreme market events could improve the robustness of the model. Furthermore, an attention mechanism could help the model focus on significant market events, enhancing accuracy during anomalies.



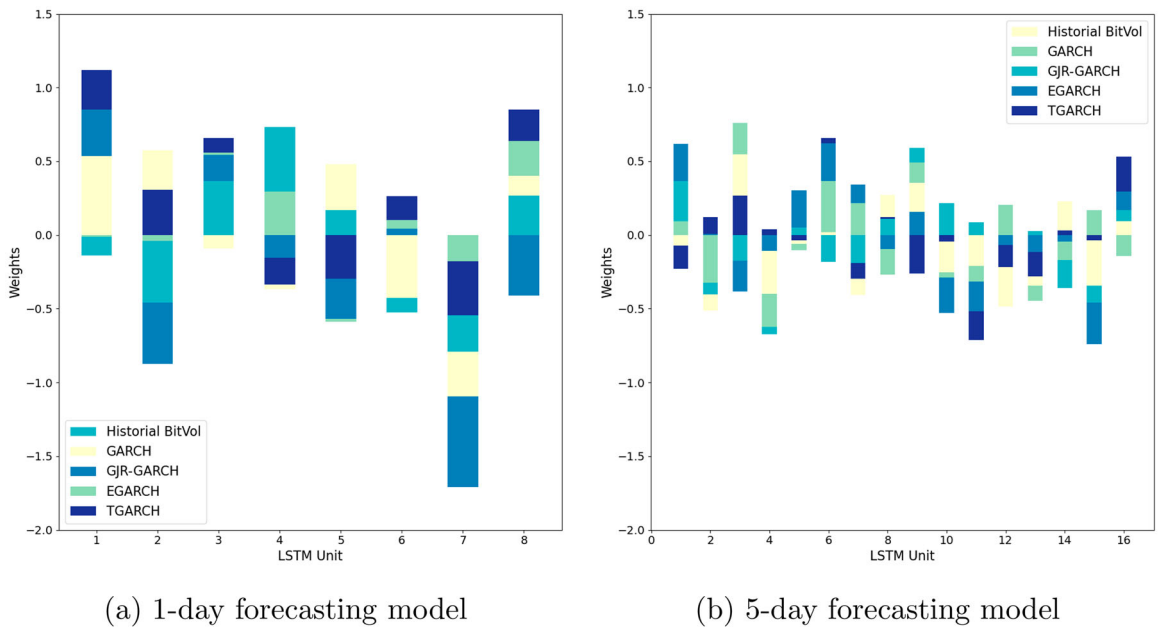
**Figure 9.** LSTM errors vs. Implied volatility.



**Figure 10.** LSTM 5-day implied volatility forecasting.

## 6. Discussion

In studies on deep learning applications, we notice little focus on interpreting how models work. Due to the model's complexity, extracting the impact of a feature or delineating the memory decaying patterns for LSTM is challenging. However, we still believe exhibiting deeper insights into our model is worthwhile. We examine the weights  $u_c$  that carry input  $X_t$  to new information in each LSTM unit (see Figures 11 and 12). We observe that each LSTM unit handles a specific linear combination of the inputs. In Figure 11(a), the 1st and 7th units primarily focus on positive and negative weighted averages, respectively. We note that the 1-day forecasting model applies 'strong' negative weights to the GJR-GARCH input (e.g. the 2nd, 7th, and 8th units), whereas this is not the case for the 5-day forecasting model. Another noteworthy finding is that the weights in the 5-day



**Figure 11.** LSTM new information weights on inputs. (a) 1-day forecasting model (b) 5-day forecasting model.

**Table 6.** LSTM features impact.

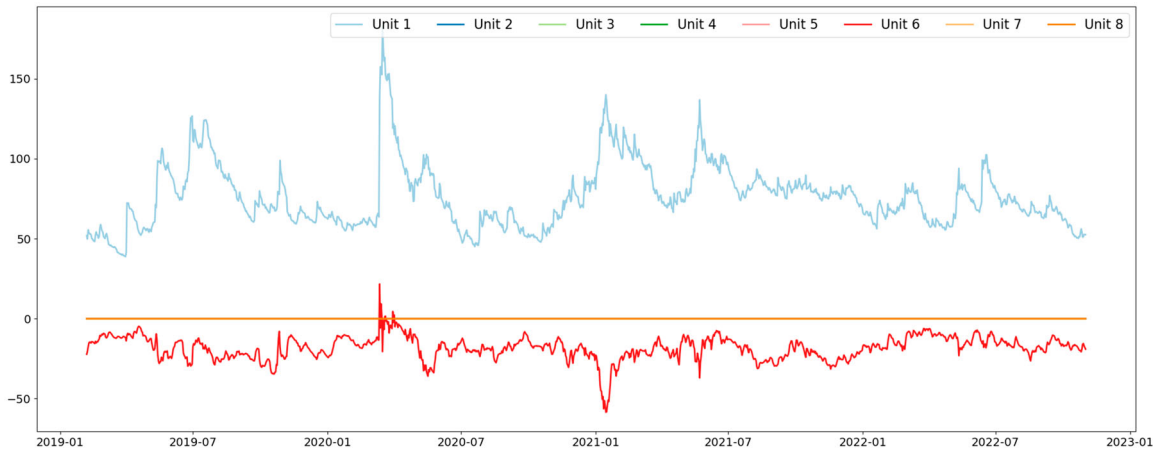
	Hist. BitVol	GARCH	GJR-GARCH	TGARCH	EGARCH
1-day model	0.7133	0.2638	−0.0427	−0.0645	0.1642
5-day model	0.7231	0.1382	0.1621	0.1232	0.0073

forecasting model are much smaller than those in the 1-day forecasting model, indicating that the former ‘tunes’ the inputs more conservatively.

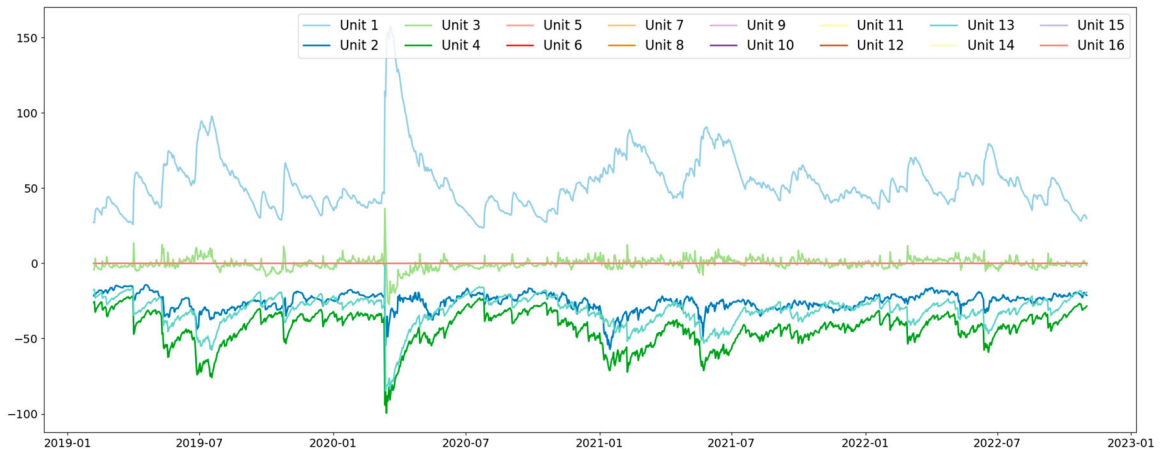
According to this observation, we run a rough analysis of the feature impact by multiplying  $u_c$ —excluding the columns of units that only give outputs equal to or close to 0 by the dense layer weights of the same units. We obtained the overall feature weights in Table 6. We conclude that the forecasting results are mainly driven by the historical BitVol index. Minor adjustments are made using different GARCH family models, with GARCH contributing the most. In the 1-day forecasting model, GJR-GARCH and TGARCH are used as a kind of ‘reversal’ indicator. EGARCH shows almost no impact in the 5-day forecasting model, indicating that it is unsuitable for long-run predictions.

## 7. Conclusion

This paper combines deep learning techniques with classic GARCH-type models to develop a hybrid forecasting model that offers more accurate volatility predictions in the Bitcoin market. Among various time series models, the GARCH family is widely used for volatility forecasting due to their stable performance and efficiency. However, our application results for GARCH(1, 1), GJR-GARCH(1, 1), TGARCH(1, 1) and EGARCH(1, 1) demonstrate that these models do not yield accurate forecasting results and often exhibit systemic errors. Consequently, we involve the LSTM model to enhance forecasting precision. By training a model capable of selectively retaining and discarding memories, our LSTM neural network reduces the 1-day and 5-day forecasting errors from over 10% to 5.70% and 8.22%, respectively. This improvement significantly enhances overall forecasting performance and provides an effective solution to the GARCH model’s tendency to under-predict extreme volatility.



(a) 1-day forecasting model



(b) 5-day forecasting model

**Figure 12.** LSTM layer outputs of training dataset. (a) 1-day forecasting model (b) 5-day forecasting model.

We conclude with three key findings from our experiment. Firstly, the Bitcoin market exhibits similar time series properties to the stock market, indicating that classic modelling techniques such as GARCH models are applicable. However, the performance of these models for volatility forecasting varies. Moreover, GARCH-type models do not effectively manage the more frequent high-volatility conditions in the Bitcoin market. Secondly, although none of the GARCH-type models we applied are adaptive to extreme market shocks, they can serve as stable input features in deep learning models. Our model demonstrates superior performance compared to most previous literature addressing similar forecasting issues, attributed to the effective input from GARCH models and historical BitVol. Lastly, the LSTM neural network possesses advantages in financial time series forecasting due to its inherent design for handling sequential data. As the market absorbs shocks and information from the past, a model that strategically builds long and short memories is beneficial for forecasting. This explains why the LSTM can refine inaccurate and outdated volatility from various sources into more accurate and robust predictions. The findings suggest several promising avenues for future research. Extending the hybrid GARCH-LSTM approach to other financial instruments, such as different cryptocurrencies or exchange markets, could provide valuable insights into its generalisation across different asset classes and volatility regimes. Techniques such

as differencing or integrating ARIMA models could help manage trends and seasonality, enabling the hybrid model to address the challenge of non-stationary time series data. However, our study also reveals certain limitations. Traditional GARCH models struggle to adapt to sudden, extreme market movements, highlighting their limitations during high-stress periods. The LSTM model also requires high-quality input data and is prone to overfitting if not optimised. Consequently, while the hybrid model is effective, it relies heavily on the quality of GARCH inputs and may exhibit reduced robustness when faced with unpredictable market shocks. To enhance the model, high-frequency data that capture rapid market fluctuations could be particularly valuable for short-term forecasting, enabling more responsive and actionable predictions. Moreover, incorporating alternative features such as trading volume or sentiment indicators derived from social media and news could enrich the model's understanding of market dynamics, especially in the highly reactive Bitcoin markets. Another key area for improvement is the integration of exogenous variables. Including macroeconomic indicators such as interest rates, inflation, and regulatory developments could provide a broader context for volatility predictions, thereby improving the robustness and accuracy of the model. Techniques such as feature selection or attention mechanisms could dynamically assess the importance of these variables to refine the model's predictive performance. Additionally, incorporating explainable AI frameworks could enhance the interpretability of the model, allowing investors and researchers to better understand the relationship between Bitcoin market conditions and predicted volatility. Thus, our study contributes to developing more robust forecasting models capable of addressing the unique challenges of cryptocurrency markets. By bridging the gap between traditional econometric methods and modern machine learning approaches, the proposed hybrid model sets the stage for further advances in financial modelling and offers a versatile framework that can be adapted to a wide range of market conditions.

## Notes

1. <https://commonslibrary.parliament.uk/research-briefings/cbp-8780/>
2. <https://www.statista.com/statistics/863917/number-crypto-coins-tokens/>
3. See <https://fintechmagazine.com/articles/top-10-cryptocurrencies-in-2023-by-market-capitalisation>.
4. <https://www.cnbc.com/2023/03/12/signature-svb-silvergate-failures-effects-on-crypto-sector.html>
5. <https://t3index.com/indexes/bit-vol/>
6. <https://finance.yahoo.com/quote/BTC-USD/>
7. <https://t3index.com/indexes/bit-vol/>

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Notes on contributors

*Beining Han* is a PhD candidate in Financial Mathematics at Cardiff University, where his research focuses on risk in cryptocurrency markets. He received his BSc in Financial Mathematics from Zhengzhou University, Henan, China, in 2021 and his MSc in Applied Statistics and Operational Research from Cardiff University, Cardiff, U.K., in 2022. His research interests lie at the intersection of quantitative finance and emerging crypto assets, with a particular emphasis on volatility forecasting, market risk, and the role of sentiment in cryptocurrency markets.

*Anqi Liu* received her B.Sc. degree in Mathematics and Applied Mathematics from Northwest University, Xi'an, China, in 2011, and her M.Sc. and Ph.D. degrees in Financial Engineering from Stevens Institute of Technology, USA, in 2013 and 2017, respectively. She is currently Senior Lecturer in Financial Mathematics in the School of Mathematics at Cardiff University. Her expertise lies in quantitative and computational finance, with a primary research focus on cryptocurrency markets. Her broader research interests include trading behaviour simulation, Hawkes processes in finance, and financial networks and system modelling. She has been Guest Editor for The Journal of Futures Markets. She has led and co-led FinTech projects funded by the UKFin+ Network and the Alan Turing Institute, contributing to the development of meaningful academic-industrial collaborations.

*Jing Chen* received the B.Sc. degree in computer science from Lanzhou Jiaotong University, Lanzhou, China, in 2001, the M.Sc. and Ph.D. degree in finance from the University of Aberdeen, Aberdeen, U.K., in 2007 and 2011, respectively. She is a professor in financial mathematics with Cardiff University, Cardiff, U.K. In the academic community, she is known for her expertise in interdisciplinary approaches of financial modelling (e.g., Hawkes processes and network approaches) that tackles modern finance problems.



Her research provides a methodological foundation for social-technical issues in financial technology (FinTech), financial market behaviour, and regulatory impacts. Prof. Chen holds editorships for The European Journal of Finance (EJF), Journal of Forecasting, International Review of Economics and Finance, IMA Journal of Management Mathematics, and Cogent Economics and Finance. She leads and co-leads national scale projects including the Engineering and Physical Sciences Research Council (EPSRC) UKFin+ Network and AI for Collective Intelligence (AI4CI), and many other projects with a wide range of stakeholders including Wales Data Nation Acceleration Program, Turing Institute, Office for National Statistics, Royal Statistical Society, Nationwide, FCA etc.

**William Knottenbelt** is Professor of Applied Quantitative Analysis in the Department of Computing at Imperial College London. He has a broad research interest in the application of mathematical modelling techniques to real-world systems. He is Director of the Imperial College Centre for Cryptocurrency Research and Engineering, where he contributes to research on cryptocurrencies, blockchains, distributed ledgers, and smart contracts. William has co-authored more than 200 scientific papers, serves as an editor of Performance Evaluation Journal, and has chaired numerous conferences and workshops on quantitative modelling and cryptocurrencies, including ACM/SPEC ICPE 2024 and IEEE ICBC 2024. In addition to his academic work, he actively supports innovation and entrepreneurship, serving as a technical advisor to several startups in fintech and blockchain.

## ORCID

Beining Han  <http://orcid.org/0009-0007-8980-1259>

Anqi Liu  <http://orcid.org/0000-0002-9224-084X>

Jing Chen  <http://orcid.org/0000-0001-7135-2116>

## References

- AlMadany, Nehal N., Omar Hujran, Ghazi Al Naymat, and Aktham Maghyereh. 2024. "Forecasting Cryptocurrency Returns Using Classical Statistical and Deep Learning Techniques." *International Journal of Information Management Data Insights* 4 (2):100251. <https://doi.org/10.1016/j.jjimei.2024.100251>.
- Amirshahi, Bahareh, and Salim Lahmiri. 2023. "Hybrid Deep Learning and GARCH-family Models for Forecasting Volatility of Cryptocurrencies." *Machine Learning with Applications* 12:100465. <https://doi.org/10.1016/j.mmlwa.2023.100465>.
- Andersen, Torben G., Tim Bollerslev, and Nour Meddahi. 2011. "Realized Volatility Forecasting and Market Microstructure Noise." *Journal of Econometrics* 160 (1): 220–234. <https://doi.org/10.1016/j.jeconom.2010.03.032>.
- Aras, Serkan. 2021. "On Improving GARCH Volatility Forecasts for Bitcoin via a Meta-learning Approach." *Knowledge-Based Systems* 230:107393. <https://doi.org/10.1016/j.knosys.2021.107393>.
- Ardia, David, Keven Bluteau, Kris Boudt, and Leopoldo Catania. 2018. "Forecasting Risk with Markov-Switching GARCH Models: A Large-Scale Performance Study." *International Journal of Forecasting* 34 (4): 733–747. <https://doi.org/10.1016/j.ijforecast.2018.05.004>.
- Barndorff-Nielsen, Ole E., Peter R. Hansen, Asger Lunde, and Neil Shephard. 2011. "Subsampling realised kernels." *Journal of Econometrics* 160 (1): 204–219.
- Bergsli, Lykke Øverland, Andrea Falk Lind, Peter Molnár, and Michał Polasik. 2022. "Forecasting Volatility of Bitcoin." *Research in International Business and Finance* 59:101540. <https://doi.org/10.1016/j.ribaf.2021.101540>.
- Bollerslev, Tim. 1986. "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics* 31 (3): 307–327. [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1).
- Canina, Linda, and Stephen Figlewski. 1993. "The Informational Content of Implied Volatility." *Review of Financial Studies* 6 (3): 659–681. <https://doi.org/10.1093/rfs/5.3.659>.
- Caporale, Guglielmo Maria, and Timur Zekokh. 2019. "Modelling Volatility of Cryptocurrencies Using Markov-Switching GARCH Models." *Research in International Business and Finance* 48:143–155. <https://doi.org/10.1016/j.ribaf.2018.12.009>.
- Charles, Amélie, and Olivier Darné. 2019. "The Accuracy of Asymmetric GARCH Model Estimation." *International Economics* 157:179–202. <https://doi.org/10.1016/j.inteco.2018.11.001>.
- Christensen, B. J., and N. R. Prabhala. 1998. "The Relation between Implied and Realized Volatility." *Journal of Financial Economics* 50 (2): 125–150. [https://doi.org/10.1016/S0304-405X\(98\)00034-8](https://doi.org/10.1016/S0304-405X(98)00034-8).
- Corsi, Fulvio. 2009. "A Simple Approximate Long-Memory Model of Realized Volatility." *Journal of Financial Econometrics* 7 (2): 174–196. <https://doi.org/10.1093/jfinrec/nbp001>.
- Diebold, Francis X., and Roberto S. Mariano. 1995. "Comparing Predictive Accuracy." *Journal of Business & Economic Statistics* 13 (3): 253–263. <https://doi.org/10.1080/07350015.1995.10524599>.
- Dyhrberg, Anne Haubo. 2016. "Bitcoin, Gold and the Dollar—A GARCH Volatility Analysis." *Finance Research Letters* 16:85–92. <https://doi.org/10.1016/j.frl.2015.10.008>.
- Engle, R. F., and A. J. Patton. 2001. "What Good Is a Volatility Model?" *Quantitative Finance* 1 (2): 237–245. <https://doi.org/10.1088/1469-7688/1/2/305>.
- Engle, Robert F. 1982. "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica: Journal of the Econometric Society* 50 (4): 987–1007. <https://doi.org/10.2307/1912773>.

- Fassas, Athanasios P., and Costas Siriopoulos. 2021. "Implied Volatility Indices – A Review." *The Quarterly Review of Economics and Finance* 79:303–329. <https://doi.org/10.1016/j.qref.2020.07.004>.
- Franses, Philip Hans, and Dick Van Dijk. 1996. "Forecasting Stock Market Volatility Using (Non-linear) Garch Models." *Journal of Forecasting* 15 (3): 229–235. [https://doi.org/10.1002/\(ISSN\)1099-131X](https://doi.org/10.1002/(ISSN)1099-131X).
- Fu, X. 2023. "Oil Price Forecasting Model Based on GARCH-LSTM Model." *Frontiers in Business, Economics and Management* 10 (3): 28–31. <https://doi.org/10.54097/fbem.v10i3.11205>.
- Z. Gao, Y. H., and E. E. Kuruoglu. 2021. "A Hybrid Model Integrating LSTM and Garch for Bitcoin Price Prediction. 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). Gold Coast, Australia. 2021.10.1109/MLSP52302.2021.9596429.
- García-Medina, A., and E. Aguayo-Moreno. 2023. "LSTM–GARCH Hybrid Model for the Prediction of Volatility in Cryptocurrency Portfolios." *Computational Economics* 63 (4): 1511–1542. <https://doi.org/10.1007/s10614-023-10373-8>.
- Garman, Mark B., and Michael J. Klass. 1980. "On the Estimation of Security Price Volatilities from Historical Data." *Journal of Business* 53 (1): 67–78. <https://doi.org/10.1086/jb.1980.53.issue-1>.
- 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 (5): 1779–1801. <https://doi.org/10.1111/jofi.1993.48.issue-5>.
- Haas, Markus, Stefan Mittnik, and Marc S. Paoletta. 2004. "A New Approach to Markov-Switching GARCH Models." *Journal of Financial Econometrics* 2 (4): 493–530. <https://doi.org/10.1093/jfinfec/nbh020>.
- Hansen, Peter Reinhard, and Zhuo Huang. 2016. "Exponential GARCH Modeling with Realized Measures of Volatility." *Journal of Business & Economic Statistics* 34 (2): 269–287. <https://doi.org/10.1080/07350015.2015.1038543>.
- Hansen, Peter Reinhard, Zhuo Huang, and Howard Howan Shek. 2012. "Realized GARCH: A Joint Model for Returns and Realized Measures of Volatility." *Journal of Applied Econometrics* 27 (6): 877–906. <https://doi.org/10.1002/jae.v27.6>.
- Harvey, David, Stephen Leybourne, and Paul Newbold. 1997. "Testing the Equality of Prediction Mean Squared Errors." *International Journal of Forecasting* 13 (2): 281–291. [https://doi.org/10.1016/S0169-2070\(96\)00719-4](https://doi.org/10.1016/S0169-2070(96)00719-4).
- Hoang, Lai T., and Dirk G. Baur. 2020. "Forecasting Bitcoin Volatility: Evidence from the Options Market." *Journal of Futures Markets* 40 (10): 1584–1602. <https://doi.org/10.1002/fut.v40.10>.
- Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long short-term Memory." *Neural Computation* 9 (8): 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- 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 Young, and Chang Hyun 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. <https://doi.org/10.1016/j.eswa.2018.03.002>.
- Kristjanpoller, W., and M. C. Minutolo. 2016. "Forecasting Volatility of Oil Price Using an Artificial Neural Network–GARCH Model." *Expert Systems with Applications* 65:233–241. <https://doi.org/10.1016/j.eswa.2016.08.045>.
- Li, Nan, Xun Liang, Xinli Li, Chao Wang, and Desheng Dash Wu. 2009. "Network Environment and Financial Risk Using Machine Learning and Sentiment Analysis." *Human and Ecological Risk Assessment* 15 (2): 227–252. <https://doi.org/10.1080/10807030902761056>.
- McAleer, Michael, and Marcelo C. Medeiros. 2008. "Realized Volatility: A Review." *Econometric Reviews* 27 (1–3): 10–45. <https://doi.org/10.1080/07474930701853509>.
- Nelson, Daniel B. 1991. "Conditional Heteroskedasticity in Asset Returns: A New Approach." *Econometrica* 59 (2): 347–370. <https://doi.org/10.2307/2938260>.
- Nsengiyumva, Elysee, Joseph K. Mung'atu, and Charles Ruranga. 2025. "Hybrid GARCH-LSTM Forecasting for Foreign Exchange Risk." *FinTech* 4 (2): 22. <https://doi.org/10.3390/fintech4020022>.
- Parkinson, Michael. 1980. "The Extreme Value Method for Estimating the Variance of the Rate of Return." *Journal of Business* 53 (1): 61–65. <https://doi.org/10.1086/jb.1980.53.issue-1>.
- Peng, Yaohao, Pedro Henrique Melo Albuquerque, Jader Martins Camboim de Sá, Ana Julia Akaishi Padula, and Mariana Rosa Montenegro. 2018. "The Best of Two Worlds: Forecasting High Frequency Volatility for Cryptocurrencies and Traditional Currencies with Support Vector Regression." *Expert Systems with Applications* 97:177–192. <https://doi.org/10.1016/j.eswa.2017.12.004>.
- Pérez-Cruz, Fernando, Julio A. Afonso-Rodríguez, and Javier Giner. 2003. "Estimating GARCH Models using Support Vector Machines." *Quantitative Finance* 3 (3): 163–172. <https://doi.org/10.1088/1469-7688/3/3/302>.
- Seo, Monghwan, and Geonwoo Kim. 2020. "Hybrid Forecasting Models Based on the Neural Networks for the Volatility of Bitcoin." *Applied Sciences* 10 (14): 4768. <https://doi.org/10.3390/app10144768>.
- 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 (7): 337. <https://doi.org/10.3390/jrfm14070337>.
- Sun, H., and B. Yu. 2020. "Forecasting Financial Returns Volatility: A GARCH-SVR Model." *Computational Economics* 55 (2): 451–471. <https://doi.org/10.1007/s10614-019-09896-w>.
- Taylor, James W. 2004. "Volatility Forecasting with Smooth Transition Exponential Smoothing." *International Journal of Forecasting* 20 (2): 273–286. <https://doi.org/10.1016/j.ijforecast.2003.09.010>.
- Wang, Yijun, Galina Andreeva, and Belen Martin-Barragan. 2023. "Machine Learning Approaches to Forecasting Cryptocurrency Volatility: Considering Internal and External Determinants." *International Review of Financial Analysis* 90:102914. <https://doi.org/10.1016/j.irfa.2023.102914>.

- Wu, Chih-Hung, Chih-Chiang Lu, Yu-Feng Ma, and Ruei-Shan Lu. 2018. "A New Forecasting Framework for Bitcoin Price with LSTM." 2018 IEEE International Conference on Data Mining Workshops (ICDMW). Singapore. 2018. <https://doi.org/10.1109/ICDMW.2018.00032..>
- Zahid, Mamoon, Farhat Iqbal, and Dimitrios Koutmos. 2022. "Forecasting Bitcoin Volatility Using Hybrid GARCH Models with Machine Learning." *Risks* 10 (12): 237. <https://doi.org/10.3390/risks10120237>.
- Zakoian, Jean-Michel. 1994. "Threshold Heteroskedastic Models." *Journal of Economic Dynamics and Control* 18 (5): 931–955. [https://doi.org/10.1016/0165-1889\(94\)90039-6](https://doi.org/10.1016/0165-1889(94)90039-6).
- Zulfiqar, Noshaba, and Saqib Gulzar. 2021. "Implied Volatility Estimation of Bitcoin Options and the Stylized Facts of Option Pricing." *Financial Innovation* 7 (1): 67. <https://doi.org/10.1186/s40854-021-00280-y>.

## Appendix. Look-back period testing results

Table A1 presents the performance metrics for various look-back periods tested during the empirical evaluation. The model tuning process was conducted consistently across all models, following the same process described in the Methodology section.

**Table A1.** Performance metrics for different look-back periods.

Look-Back Period	1-Day Forecast		5-Day Forecast	
	RMSE	RMSPE (%)	RMSE	RMSPE (%)
7 days	0.079	11.34	0.156	14.67
14 days	0.066	9.89	0.138	12.82
30 days	<b>0.056</b>	<b>8.23</b>	<b>0.112</b>	<b>10.45</b>
60 days	0.072	11.67	0.145	13.79

The table highlights the superior performance of the 30-day look-back period for both 1-day and 5-day forecasts. Shorter periods, such as 7 and 14 days, fail to capture sufficient temporal context, resulting in underfitting and poor predictive accuracy. Conversely, longer periods, such as 60 days, introduce excessive noise and redundancy, reducing responsiveness to recent market changes.