Forecasting Bitcoin volatility using machine learning techniques

J. Int. Financ. Markets Inst. Money 97 (2024) 102064

Contents lists available at ScienceDirect

Journal of International Financial Markets, Institutions & Money

journal homepage: www.elsevier.com/locate/intfin

Forecasting Bitcoin volatility using machine learning techniques

Zih-Chun Huang [a], Ivan Sangiorgi [a], Andrew Urquhart [b,*]

[a] ICMA Centre, Henley Business School, University of Reading, Reading, United Kingdom
[b] Birmingham Business School, University of Birmingham, Birmingham, United Kingdom

ARTICLE INFO

JEL classification:
C45
C53
G17

Keywords:
Bitcoin
Volatility forecasting
Machine learning

ABSTRACT

This paper studies the Bitcoin volatility forecasting performance between popular traditional econometric models and machine learning techniques. We compare the 1-day to 2-month ahead forecasting performance of the Long Short-Term Memory (LSTM) and a hybrid Convolutional Neural Network-LSTM (CNN-LSTM) model to the traditional models. We find that neural networks outperform Generalised Autoregressive Conditional Heteroskedasticity (GARCH) models for all forecasting horizons. Furthermore, the LSTM model outperforms the Heterogeneous Autoregressive (HAR) model and by integrating the Markov Transition Field (MTF) into the CNN-LSTM model, we achieve superior forecasting results in the short-term, particularly for the 7-day forecasts.

1. Introduction

Heterogeneous Autoregressive (HAR) models have been shown to outperform many standard volatility forecasting models when forecasting Bitcoin (Shen et al., 2020; Urquhart, 2017b; Zhang and Tan, 2018). However, with their growth and popularity ever-growing, we examine whether machine learning techniques can beat the HAR model in Bitcoin volatility forecasting. Second, we empirically test whether high-frequency Bitcoin data provide valuable information for Bitcoin volatility estimation. We aim to improve Bitcoin volatility forecasting accuracy by employing a hybrid neural network model along with image transformation.

Our primary contributions are threefold. First, we apply machine learning models to forecast Bitcoin volatility using high-frequency data. We use neural network models to extract extra hidden information from the images transformed from the high-frequency Bitcoin volatility data. Instead of incorporating additional factors, such as macroeconomic indicators (Feng et al., 2024) or Google Trends data (Seo and Kim, 2020), we utilise the Markov Transition Field (MTF) technique (Wang et al., 2015), converting ordinary time series into images with transitional probabilities, and demonstrate that MTF can contribute to Bitcoin volatility predictions. Second, to address the concerns of microstructure noises in high-frequency data, we apply the pre-averaging method (Jacod et al., 2009) and indicate that it can reduce noise effectively and prevent the neural networks from over-fitting. Lastly, our analysis expands the usual comparisons with GARCH models by including evaluations against HAR models and shows that the neural network applications outperform short-term and long-term Bitcoin volatility predictions. In contrast to previous studies that employ Bitcoin daily data, such as (Bergsli et al., 2022), we demonstrate that the intra-day Bitcoin time series contains information that can enhance short-term and long-term volatility forecasts. Moreover, because ordinary Bitcoin high-frequency data collection is more accessible and less time-consuming compared to other additional Bitcoin factors, our approach is a more realistic application that can offer valuable insights to both short-term and long-term Bitcoin traders.

Forecasting Bitcoin volatility is a notoriously difficult task caused by several reasons. First of all, Bitcoin is a relatively new investment vehicle introduced in 2008 and considered a speculative asset (Baur et al., 2018; Lee et al., 2020). The market was

* Corresponding author.
E-mail addresses: zih-chun.huang@pgr.reading.ac.uk (Z.-C. Huang), ivan.sangiorgi@icmacentre.ac.uk (I. Sangiorgi), a.urquhart@brum.ac.uk (A. Urquhart).

https://doi.org/10.1016/j.intfin.2024.102064
Received 12 January 2024; Accepted 28 September 2024
Available online 19 October 2024

illiquid in the first few years (Nadarajah and Chu, 2017; Urquhart, 2017a), which resulted in insufficient Bitcoin data for training predictive models. Furthermore, Bitcoin contains a lot of jumps, bubbles, and structural breaks (Ardia et al., 2019; Aysan et al., 2024; Bouri et al., 2019; Li et al., 2022; Shen et al., 2020). Therefore, Bitcoin volatility is more difficult to predict compared to traditional financial assets. Particular interest in accurate Bitcoin volatility predictions has grown since 2017 when Bitcoin derivatives started to trade on the Chicago Mercantile Exchange and Bitcoin entered the mainstream financial sector. In recent years, there have been more institutional investors who tend to deploy capital in digital assets, especially in Bitcoin, one of the largest and most liquid digital assets. For example, according to PwC,[1] in 2021, it is reported that 21% of hedge funds surveyed invest in digital assets and most of them traded Bitcoin. Furthermore, the major investment bank Goldman Sachs[2] and the insurance giant MassMutual[3] have also made investments in Bitcoin. J.P. Morgan also predicted strong demand and a promising future.[4] (Huang et al., 2022) find there has been a consistent increase in the percentage of institutions with exposure to the cryptocurrency market for risk diversification. Consequently, accurate predictions for Bitcoin volatility have become more important. However, the literature regarding precise forecasts of Bitcoin volatility remains quite limited compared to other traditional financial assets. Hence, we aim to provide evidence in Bitcoin volatility forecasting for both Bitcoin traders and institutional investors.

Based on the existing literature, there are three main types of models used to forecast Bitcoin volatility. Firstly, some studies focus on the traditional models, such as HAR-type models (Bergsli et al., 2022; Shen et al., 2020) and GARCH-type models (Chi and Hao, 2021; Katsiampa, 2017; Köchling et al., 2020). Secondly, some researchers employ hybrid models that integrate traditional models and neural networks (Aras, 2021a,b; Kristjanpoller and Minutolo, 2018) resulting in superior predictive capabilities when compared to traditional models. Third, other works test the performance between various neural networks with different input data including Bitcoin raw time series (D'Amato et al., 2022) and Bitcoin data with additional explanatory variables, such as Google Trends and Chicago Board Options Exchange's Volatility Index (VIX) (Seo and Kim, 2020).

Volatility prediction by neural networks has been studied in different financial assets, such as Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). LSTM, a type of recurrent neural network, has been widely applied in modelling sequential data. Studies have shown that incorporating LSTM in models can improve volatility prediction due to the long memory effect in volatility. For instance, Xiong et al. (2015) model S&P500 volatility by LSTM with Google Domestic Trends. Zhou et al. (2019) use the Baidu Index's searching volumes of 28 trending words to enhance the forecasting capability of LSTM on CSI 300[5] daily volatility change. Some researchers also integrate LSTM with traditional GARCH-type models for volatility predictions. For example, Kim and Won (2018) use a hybrid GEW-LSTM model, a combination of LSTM and three GARCH-type models, to predict the volatility of the KOSPI 200 index.[6]

Some researchers incorporate CNN in their volatility models because the convolutional layer in CNN can extract non-linear features by convolution operations. Doering et al. (2017) examine the ability of CNN to forecast volatility and capture market microstructure features by training a deep CNN on a full limit-order book dataset. Vidal and Kristjanpoller (2020) combine LSTM and CNN, retaining the advantages of both neural networks to increase the accuracy of gold volatility prediction. This hybrid model outperforms other models, such as the GARCH, Regression Support Vectors, LSTM, and CNN models.

Although some literature applies machine learning techniques to predict Bitcoin prices (Aggarwal et al., 2020; Atsalakis et al., 2019; Catania and Sandholdt, 2019; Chen et al., 2021; Gradojevic et al., 2023; Ji et al., 2019; Liu et al., 2021; McNally et al., 2018), only a few papers study Bitcoin volatility forecasting using machine learning techniques and Table 1 summarise the related literature. Kristjanpoller and Minutolo (2018) incorporate Artificial neural network (ANN) with the GARCH model and conclude that the proposed model outperforms the GARCH model in Bitcoin volatility forecasting. Seo and Kim (2020) improve volatility prediction by applying neural networks to extract information from Google Trends, VIX index, 1-day lagged weekly volatility, realised volatility, and outputs from GARCH-type models. Feng et al. (2024) build a novel daily dynamic tuning strategy to enhance cryptocurrency volatility forecasting with machine learning. Dudek et al. (2024) compare cryptocurrency volatility forecasting performance between several models, including HAR, GARCH, and LSTM models, and find that no single best method. D'Amato et al. (2022) predict cryptocurrency volatility by using neural networks and daily prices. Peng et al. (2018) combine the GARCH model with Support Vector Regression to estimate volatility and show the outperformance of the combined model. Wang et al. (2023) show that LSTM outperforms the GARCH model and find that internal determinants, such as previous trading information and volatility, are the most crucial features for cryptocurrency volatility forecasting.

In this paper, we aim to improve predictive performance and extend the literature on Bitcoin volatility forecasting. We apply machine learning techniques to estimate Bitcoin volatility and compare the forecasting performance to the traditional models used in the literature on volatility modelling. Moreover, we explore the potential benefits of the time series to image transformation as presented in Vidal and Kristjanpoller (2020) and Wang et al. (2015). This paper follows Vidal and Kristjanpoller (2020) to use a hybrid model combining LSTM and CNN along with image transformation to reserve benefits of both: the capability of capturing

[1] The statistics is from PwC Annual Global Crypto Hedge Fund Report 2021 (https://www.pwc.com/gx/en/financial-services/pdf/3rd-annual-pwc-elwood-aima-crypto-hedge-fund-report-(may-2021).pdf).

[2] https://www.reuters.com/business/finance/exclusive-goldman-sachs-restarts-cryptocurrency-desk-amid-bitcoin-boom-2021-03-01/.

[3] https://www.bloomberg.com/news/articles/2020-12-10/169-year-old-insurer-massmutual-invests-100-million-in-bitcoin?utm_source=newsletter&utm_medium=email&utm_campaign=newsletter_axiosmarkets&stream=business.

[4] https://www.financemagnates.com/cryptocurrency/news/jpmorgan-predicts-600-billion-bitcoin-demand/.

[5] CSI 300, China Securities Index 300, is a capitalisation-weighted stock market index replicating the performance of the top 300 stocks traded on the Shenzhen Stock Exchange and the Shanghai Stock Exchange.

[6] The KOSPI 200 Index is a capitalisation-weighted index of 200 Korean stocks of the Korea Stock Exchange.

Table 1
Summary of literature applying machine learning techniques in the Cryptocurrency volatility forecasting domain.

Authors	Target variables	Dataset	Forecasting models	Evaluation metrics
Feng et al. (2024)	Cryptocurrency volatility	Bitcoin and Ethereum from cryptodatadownload.com, and 19 macroeconomic indicators	HAR-RV model, Lasso, Ridge regression, and Elastic net with tuning strategy	RMSE, MSPE, and MCS
Dudek et al. (2024)	Cryptocurrency volatility	Bitcoin, Ethereum, Litecoin, and Monero historical prices from Kraken	HAR, ARFIMA, GARCH, RR, LASSO, SVR, MLP, FNM, RF, and LSTM	MCS, MSE, and MAE
Wang et al. (2023)	Cryptocurrency volatility	Bitcoin, Ethereum, Litecoin, and Ripple from Coin-API, 6 Blockchain data from Blockchain.com, Google Trends data, Crypto Fear & Greed Index from Bitcoin fear, Policy uncertainty data from Economic policy uncertainty, and 2 financial data from Yahoo YQL Finance API	RF and LSTM	RMSE, MAPE, NMSE, and DA
D'Amato et al. (2022)	Cryptocurrency volatility	Bitcoin, Ripple, and Ethereum daily prices from CoinMarketCap.	JNN, NLNN, and SETAR	MSE, and MAPE
Seo and Kim (2020)	Bitcoin price volatility	Bitcoin from Bitcoincharts, Google trends data, and VIX index	GARCH-type models, ANN, and HONN	RMSE, MAE, and MAPE
Peng et al. (2018)	Cryptocurrency volatility and currency volatility	Bitcoin, Ethereum, and Dash prices from Altcoin Charts, and currency rates from http://fxhistoricaldata.com/	GARCH-type models and GARCH-SVR model	RMSE and MAE
Kristjanpoller and Minutolo (2018)	Bitcoin price volatility and currency price volatility	Variances of the Bitcoin price return, and 7 technical indexes	ANN and GARCH models	MSE and MCS

This table provides the related literature on cryptocurrency volatility forecasting with machine learning techniques. The acronyms explanation is provided in Appendix H.

non-linear features and modelling long memory effect, which exists in Bitcoin time series (Bariviera, 2017; Catania and Grassi, 2022; Phillip et al., 2018). Instead of using deep neural network architecture, we apply a simple layer CNN-LSTM model to avoid overfitting and computational consumption. We train our neural network model with Bitcoin realised volatility estimated from tick-level data while Vidal and Kristjanpoller (2020) employ a pre-trained network. We do not use a pre-trained network because the images transformed from Bitcoin time series do not possess natural edges and angles as in natural images (Wang et al., 2015). Moreover, the images are encoded with Bitcoin high-frequency data, which is more volatile than the daily Gold volatility tested in Vidal and Kristjanpoller (2020). Therefore, we argue that training models based on Bitcoin raw time series rather than a pre-trained deep learning model can be advantageous.

The forecasting performance is compared to HAR and GARCH-type models evaluated by three loss functions, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Square Logarithmic Error (MSLE). We also conduct pair comparisons using (Diebold and Mariano, 2002) test (DM test) following Fischer and Krauss (2018) and Kim and Won (2018). We find that the LSTM and the hybrid CNN-LSTM model have better forecasting performance than the GARCH-type model with improvements ranging from 5% to 15%, and the hybrid model outperforms other models in the short-term, especially in 7-day ahead forecasts, which is 9.77% better than HAR. Moreover, although machine learning does not always beat the HAR model, the LSTM model dominates the HAR model among all forecasting horizons. Finally, we incorporate images encoded from the ordinary time series into the hybrid model and find that the predictive accuracy increases with respect to LSTM models with the same frequency realised volatility. This infers that non-linear information encoded in the images is beneficial for forecasting Bitcoin volatility.

This paper contributes to the literature in the following ways. First, we fill the gap in the literature by employing high-frequency Bitcoin data for volatility modelling (Esparcia et al., 2023; Katsiampa et al., 2019; Naeem et al., 2022), and study the relatively underexplored domain of using machine learning techniques in this context (Kristjanpoller and Minutolo, 2018; Seo and Kim, 2020). Compared to most traditional assets, Bitcoin trades 24 h a day 7 days a week which enables more interesting forecasting capabilities by avoiding post-market and premarket trading which is usually restricted to certain investors. Although there are concerns about microstructure noises in tick-level data, especially with the highly volatile Bitcoin, we apply the pre-averaging method (Jacod et al., 2009) to cope with this issue. Our preliminary result shows that by using the pre-averaging method, the high-frequency data can reduce a certain amount of noise and avoid over-fitting issues. Additionally, our findings demonstrate that CNN-LSTM and LSTM models when applied to high-frequency data, are particularly efficient in predicting volatility for one-week, one-month, and two-month forecasting horizons. Second, we show the benefits of applying machine learning techniques to Bitcoin volatility forecasting compared to traditional econometric models. Our analysis extends beyond typical performance comparisons of cryptocurrency volatility forecasting between machine learning with GARCH models (Kristjanpoller and Minutolo, 2018; Seo and Kim, 2020) by incorporating evaluations against HAR models, one of the most parsimonious volatility predictive models. To make

Table 2
Bitcoin zero trades percentage.

Year	1-min	2-min	3-min	4-min	5-min	10-min	15-min	20-min	25-min	30-min	35-min	40-min	45-min	50-min	55-min	60-min
2011	99.71%	99.45%	99.21%	98.98%	98.76%	97.66%	96.57%	95.58%	94.53%	93.57%	92.69%	91.85%	91.03%	90.07%	89.11%	88.31%
2012	94.95%	91.57%	88.62%	85.98%	83.51%	73.67%	66.87%	61.40%	56.71%	52.64%	49.42%	46.33%	43.55%	41.11%	39.38%	37.37%
2013	39.16%	27.90%	22.08%	18.40%	15.79%	9.40%	6.76%	5.25%	4.16%	3.48%	2.88%	2.60%	2.17%	1.84%	1.67%	1.40%
2014	**24.24%**	**10.28%**	**5.21%**	**2.95%**	**1.76%**	**0.24%**	**0.09%**	**0.07%**	**0.06%**	**0.05%**	**0.05%**	**0.05%**	**0.03%**	**0.04%**	**0.02%**	**0.02%**
2015	29.08%	14.17%	8.08%	5.19%	3.69%	1.60%	1.31%	1.27%	1.23%	1.23%	1.23%	1.23%	1.22%	1.22%	1.21%	1.22%
2016	33.19%	15.92%	8.63%	4.99%	3.06%	0.46%	0.09%	0.04%	0.02%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2017	8.04%	2.39%	0.98%	0.46%	0.25%	0.03%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2018	3.77%	0.92%	0.31%	0.11%	0.06%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2019	3.28%	0.53%	0.15%	0.05%	0.03%	0.02%	0.01%	0.02%	0.01%	0.01%	0.01%	0.01%	0.00%	0.01%	0.00%	0.00%
2020	1.55%	0.19%	0.11%	0.10%	0.09%	0.08%	0.07%	0.07%	0.07%	0.06%	0.07%	0.06%	0.06%	0.06%	0.06%	0.05%
2021	0.70%	0.09%	0.06%	0.05%	0.05%	0.04%	0.03%	0.03%	0.03%	0.02%	0.03%	0.02%	0.02%	0.02%	0.01%	0.00%

This table shows the zero trades percentage during the whole period of accessible Bitcoin trade data from 2011 to 2021. Zero trades mean there are no transactions during the given sampling frequency. Zero trades percentage in a given year is computed by dividing the number of time intervals with no trades by the total number of time intervals in the given year. The higher zero trade percentage indicates the less liquid the trading activities. We find that the figure drops from 2013 to 2014, therefore, our sampling period starts from the year 2014.

fair comparisons between CNN-LSTM and HAR models, we evaluate the predictive power under ceteris paribus conditions across various forecasting horizons. Third, different from other studies of Bitcoin volatility forecasting with machine learning (D'Amato et al., 2022; Seo and Kim, 2020), we transform Bitcoin time series to Markov Transition Field images (Wang et al., 2015) to improve the predictive performance rather than incorporating several other indicators, such as Google Trends data or implied volatility. This approach reveals that the transitional probability features in Bitcoin high-frequency data can be beneficial for Bitcoin volatility forecasting, which can avoid the time-consuming or hardly accessible collection of additional high-frequency data, and hence be realistic for real-time forecasting applications. The empirical analysis in this paper contributes new evidence to the existing literature on Bitcoin volatility forecasting.

This paper is structured as follows: Section 2 describes the data used in this paper followed by the volatility proxy estimation method in Section 3. Section 4 introduces the image transformation techniques, LSTM, CNN, and the architecture of the proposed hybrid CNN-LSTM model. The experiment results and discussions are given in Sections 5 and 6, respectively. Finally, the conclusion is in Section 7.

2. Data

In this paper, we use Bitcoin tick-level data collected from https://bitcoincharts.com, which provides the entire historical data of several cryptocurrency exchanges. We collect the tick-level Bitcoin exchange rate (BTC/USD) reported on the Bitstamp exchange from November 13, 2011, to December 31, 2021, since it is one of the most popular and long-standing exchanges that has sufficient liquidity (Shen et al., 2020). Different from other financial assets, Bitcoin is traded 24 h a day and 7 days a week, therefore, the data is continuous throughout the time period except for the time period between 2015/01/05 09:12:25 and 2015/01/09 21:05:04 when the Bitstamp exchange was closed due to security breach,[7] hence, we obtain 60.5 million of trades in total. However, the Bitcoin trading activities are illiquid in the first few years, thus, to choose the sampling period, we quantify liquidity by calculating the percentage of zero transactions in each year.[8] Table 2 tabulates the zero trades percentage in each year among different frequencies. There is a significant decline from the zero trades percentage in 2013 to 2014, especially for the sampling frequency lower than 10-min, the percentages drop below 1%. Therefore, the sampling period begins in 2014, since before that point, liquidity was relatively low. As a result, we have 55.76 million tick-level transactions as our dataset.

3. Volatility estimation

Realised volatility (RV) was first proposed by Andersen and Bollerslev (1998), Andersen et al. (1999) and widely used as a volatility proxy not only in traditional financial assets (Andersen et al., 2003; Kim and Won, 2018; Vidal and Kristjanpoller, 2020), but also in cryptocurrencies (Aras, 2021b; Shen et al., 2020). Hence, we apply realised volatility as a proxy for high-frequency variance. However, it has been documented that the realised volatility estimator at high frequency could be affected by microstructure noises (Andersen and Bollerslev, 1997). Therefore, we use the volatility signature plot to decide the sampling frequency, which is computed by the average daily realised volatility among different sampling frequencies from 2014 to 2021 reported in Table 3. The average RV drops 9.24% from 5-min to 10-min frequency and after which, the largest decline is no more

[7] On 6th January 2015, Financial Times reported that Bitstamp suspended services due to cyber attack. The exchange announced that about 5 million dollars (fewer than 19,000 bitcoins) in their "operational wallets" had been stolen by cyber attackers. However, their customers' bitcoins were safe since the majority of bitcoins were held in "cold storage", which means bitcoins are held on computers that are disconnected from the internet (https://www.ft.com/content/668a1b0a-957d-11e4-b3a6-00144feabdc0).

[8] The percentage of zero transactions in each year equals the number of time intervals with zero trades divided by the total number of time intervals in a given year.

Table 3
Average realised volatility.

Frequency	Average realised volatility	Percentage change
1 min	0.027639	–
2 min	0.002726	−90.14%
3 min	0.002529	−7.23%
4 min	0.002413	−4.58%
5 min	0.002381	−1.32%
10 min	**0.002161**	**−9.24%**
15 min	0.002059	−4.76%
20 min	0.001964	−4.61%
25 min	0.001937	−1.34%
30 min	0.001910	−1.43%
35 min	0.001905	−0.26%
40 min	0.001839	−3.47%
45 min	0.001864	1.40%
50 min	0.001814	−2.71%
55 min	0.001851	2.06%
60 min	0.001803	−2.59%

This table presents the average daily realised volatility (RV) of Bitcoin during our sampling period, from 2014 to 2021. The realised volatility of each day is calculated by the sum of the squared log return of the corresponding sampling frequency. The largest average RV decrease is from 5-min to 10-min frequency, with 9.24%, indicating that the realised volatility is more stable using 10-min frequency and we use 10-min as the main frequency in this paper.

Table 4
Summary statistics for annualised 10-min and 1-day realised volatility.

	Panel A: Summary statistics for annualised 10-min RV	
	RV based on original return	RV based on pre-averaged return
Count	420 768	420 768
Mean	12.20	2.07
Std.	5773.24	722.14
Min	0.00	0.00
Max	3 744 851.62	468 339.05
Skewness	648.64	648.29
Kurtosis	420 741.22	420 439.39
	Panel B: Summary statistics for annualised 1-day RV	
	RV based on original return	RV based on pre-averaged return
Count	2922.00	2922.00
Mean	12.20	4.51
Std.	481.29	12.78
Min	0.00	0.00
Max	26 016.85	286.56
Skewness	54.04	11.26
Kurtosis	2920.74	179.11

This table provides the summary statistics of the annualised 10-min and 1-day realised volatility based on the original tick level log return series and the pre-averaged log return series. The 10-min RV is annualised by multiplying $144 * 365$ and the 1-day RV is annualised by multiplying 365, where 144 is the number of 10-min intervals in a day and 365 is the trading day in a year for bitcoin because it is traded 24/7. This table shows that the pre-averaging method decreases the extreme RV, demonstrating its capability of efficiently reducing microstructure noises.

than 5%. This implies that realised volatility is more stable using 10-min frequency and we use 10-min as the main frequency in this paper.[9]

However, as presented in Panel A of Table 4, the average of annualised RV aggregated by the log returns is 12.2, which is still higher than other assets (Christensen and Hansen, 2002; Liu et al., 2015). Therefore, we use pre-averaged realised volatility estimators instead as in Liu et al. (2015) and Alexander et al. (2022) to mitigate possible microstructure noises. Following Jacod et al. (2009), we use the pre-averaging method to smooth the high-frequency log return series in the following way: firstly, the original log return of tick-level data is calculated by

$$r_t = log P_t - log P_{t-1} \tag{1}$$

[9] Recent literature mostly selects 5-min as sampling frequency for calculating daily realised volatility and shows that other frequencies are difficult to significantly beat 5-min RV (Liu et al., 2015). Nevertheless, Bitcoin is a new investment vehicle and extremely volatile, hence we decide our sampling frequency by using the volatility signature plot (Andersen et al., 1999) as discussed.

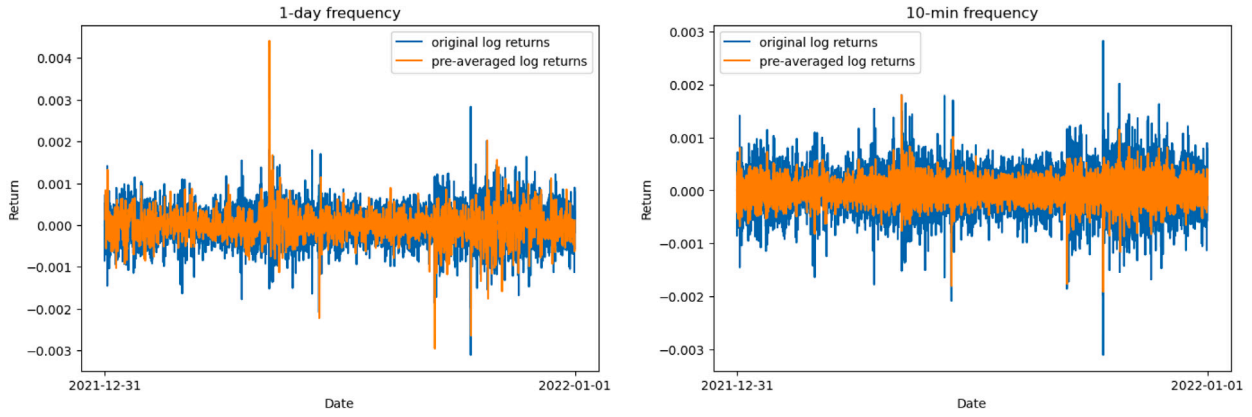

Fig. 1. The comparisons between the pre-averaged log returns and the original log returns. This figure shows the effect of the microstructure noise reduction by the pre-averaging method. We present the difference between the original log returns and the pre-averaged log returns for both 10-min and 1-day frequencies. For simplicity, we only display the last day of our sample period, 31 December 2021, as an example.

where $t-1$ denotes one tick level data before the tick level data at time t. The pre-averaged log returns are calculated by

$$\overline{r_t} = \sum_{i=1}^{k_n} g(\frac{i}{k_n}) r_{t+i} \tag{2}$$

where r_t is the original log return of tick level data, and $g(x) = min(x, 1-x)$ is a weighting function. According to empirical work in Hautsch and Podolskij (2013), the pre-averaging window k_n is chosen to be $\left\lceil \theta\sqrt{n} \right\rceil$, where n is the number of observations in a given window and θ can choose to be 0.4 because Bitcoin can be seen as frequently traded asset.[10] By summing up the squared smoothed log returns $\overline{r_t}$ in the given time interval, we can obtain RV of different frequencies:

$$RV_{t,N} = \sum_{j=1}^{N} \overline{r_{t,j}}^2 \tag{3}$$

where N is the total number of pre-averaged log returns in a given time interval and $\overline{r_{t,j}}$ is the pre-averaged log return calculated above. However, because the 10-min RV is too small, over 97% of RVs are under 0.0001, we employ the basis point of the RV as inputs of the predictive models.

Table 4 presents the summary statistics for the annualised 10-min and 1-day RV based on both the original Bitcoin log return series and the pre-averaged log return series. As discussed, the annualised RV based on the original series is high and volatile, where there is a huge difference between the mean and maximum value for both frequencies. After applying the pre-averaging method, the mean value drops from 12.2 to 2.07 and 4.51 for 10-min and daily frequency, respectively. The extreme value of the pre-averaged RV for both frequencies also declines more than 80% compared to the ordinary RV series. Furthermore, the effect of the noise reduction can also be seen in Fig. 1, which shows the difference between the original log returns and the pre-averaged log returns for both 10-min and 1-day frequency.[11]

4. Methodology

In this paper, we forecast future ahead volatility by Long Short-Term Memory (LSTM) and a hybrid model combining LSTM and Convolutional Neural Network (CNN). LSTM and CNN are widely used in time series classification and image recognition tasks (Rawat and Wang, 2017; Sezer et al., 2020), respectively. To take advantage of CNN, we use images transformed from the pre-averaged RV series as CNN's input. Compared to most of the previous literature, we focus on the high-frequency information included in the Bitcoin time series instead of adding several Bitcoin-related factors to our model. Regarding the model architecture, we use one simple layer rather than multiple layers or even deep neural networks because even though adding layers might improve model performance, it might also lead to overfitting (Shorten and Khoshgoftaar, 2019).

[10] For example, if we have tick level data every second in a 10-min time interval, this means that n is 600 and k_n is $\left\lceil 0.4\sqrt{600} \right\rceil = 10$ when pre-averaging the returns of this 10-min time interval.

[11] We provide the preliminary LSTM results and compare the 10-min ordinary RV and the 10-min pre-averaged RV in Appendix B. Panel A shows the forecasting accuracy of the LSTM model with the ordinary RV as inputs. The in-sample RMSE of the model is around 75 times the out-of-sample RMSE indicating the model is not properly trained using the ordinary RV. In comparison, Panel B presents the performance of LSTM with the pre-averaged RV, which does not have this issue. It is obvious that certain levels of noise are eliminated after smoothing the return series, hence, we use pre-averaged RV as the volatility proxy in this paper.

4.1. Encoding time series to images

To enhance the prediction accuracy, we also transform the RV time series into images to provide transition probability information to our proposed model in addition to the realised volatility time series. We employ a novel transformation framework, Markov Transition Field (MTF), introduced by Wang et al. (2015) because the authors demonstrate that the image classification performance can be improved by applying MTF images, which contain high-level features, such as 2-dimension temporal dependency in transition probability information, that cannot be found in original time series data.

4.1.1. Markov transition field

Markov Transition Field consists of the dynamic transition statistics and preserves temporal information.Given a time series X, which elements, x_i, can be separated into Q quantile bins, which is set to be 4 in our experiments. Then, $Q \times Q$ weighted adjacency matrix W can be constructed with $w_{i,j}$ in the manner of first-order Markov chain along time. $w_{i,j}$ denotes the transition probability of a point in quantile q_i to a point in quantile q_j, where $q \in [1, Q]$. The normalisation process is performed such that $\sum w_{i,j} = 1, \forall i \in [1, Q]$. Finally, Markov Transition Field (MTF) can be constructed as follows

$$MTF = \begin{bmatrix} w_{i,j|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{i,j|x_1 \in q_i, x_n \in q_j} \\ w_{i,j|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{i,j|x_2 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{i,j|x_n \in q_i, x_1 \in q_j} & \cdots & w_{i,j|x_n \in q_i, x_n \in q_j} \end{bmatrix} \tag{4}$$

The MTF encodes the transition probabilities between the different time intervals of a time series. The original time series information is contained in the main diagonal elements, $MTF_{(i,j|i=j)}$, of the matrix. Moreover, the temporal correlation between timestamps is also recorded in elements $MTF_{(i,j|i \neq j)}$. Yosinski et al. (2015) have proved that blurring images can lower computational costs without limiting the expressiveness of the graphs and reduce the noises in input pictures. Therefore, to reduce MTF size and computational cost, we also create different sizes of MTF[12] by averaging the values in each non-overlapping block to compare with the original size MTF image.

4.2. Models

This section will introduce neural networks tested in this paper. The models are used to forecast 1-day, 7-day, 14-day, 30-day, and 60-day ahead volatility. For neural networks, we use LSTM with 10-min RV as inputs and the hybrid CNN-LSTM model with 10-min RV series and its MTF images as inputs. The forecasting task can be described as a function of 10-min RV of the previous one day:

$$RV_D = f(\left[RV_{t-144}, \quad RV_{t-143}, \quad \ldots \quad , RV_{t-1}\right]) \tag{5}$$

where RV_D is daily RV and RV_t is 10-min RV and there are 144 10-min periods in 1-day. We use RV to capture daily volatility to unveil comparisons with traditional models used in the literature, such as GARCH-type (Bollerslev, 1986; Glosten et al., 1993; Zakoian, 1994) and HAR models (Corsi, 2009). To be consistent with the literature and achieve their best performance, instead of using high-frequency data, we use daily log returns and daily RV as inputs of GARCH-type and HAR models, respectively.

4.2.1. Long Short-Term Memory (LSTM)

Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) is designed to deal with the vanishing gradient problem (Pascanu et al., 2013) that Recurrent Neural Network[13] has. This problem is that RNN is difficult to train when the information passes across a long time period. LSTM solves the vanishing gradient problem by replacing each node in the hidden layer with a memory cell. In LSTM, the memory cell contains an input gate, an output gate, and a forget gate, which detailed specification is in Appendix A.2. LSTM and its extensions have been preferred choices for analysing sequential data in a variety of areas, such as handwriting recognition (Graves et al., 2008), speech recognition (Sak et al., 2014), and financial time series analysis (Fischer and Krauss, 2018). Therefore, with its benefits of learning features from the sequential data, we will use the 10-min RV time series as inputs of LSTM and days ahead volatility as the forecasting target.

In our experiments, the architecture of LSTM simply contains a single LSTM layer, one dropout layer with the rate of 0.2 to avoid over-fitting issue (Srivastava et al., 2014), and one output neuron with a sigmoid activation function (Fischer and Krauss, 2018).

4.2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (LeCun et al., 1995) is composed of convolutional layer, pooling layer, and fully connected layer. Convolutional layer aims to extract high-level features, such as edges and colours, from input image data by convolution operation. More detailed specification is in Appendix A.3. Because CNNs have successfully been applied in various areas, such as visual recognition, image classification and handwriting recognition (Krizhevsky et al., 2017), we would like to extract high-level features by CNN with images generated by MTF transformation.

[12] The reduced size has to be the divisors of the original image size. For example, because the original 10-min MTF is 144*144, the reduced image size has to be the divisors of 144.

[13] More details are in Appendix A.1.

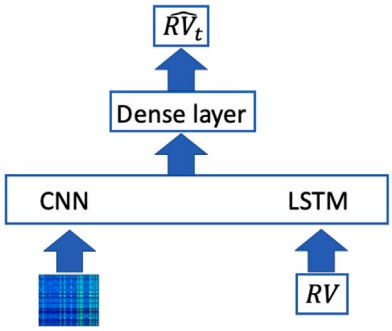

Fig. 2. CNN-LSTM model architecture. This graph shows the architecture of the CNN-LSTM model, in which the CNN model takes the MTF images as inputs and the LSMT model takes the pre-averaged RV time series as inputs. After combining the two models, we use a Dense layer to generate the final output, $\widehat{RV}_t$.

4.2.3. CNN-LSTM model

The architecture of the proposed CNN-LSTM model follows Vidal and Kristjanpoller (2020) with some adjustments. In Fig. 2, we show the general architecture of the CNN-LSTM model. The hybrid model contains two stages: First of all, the 10-min realised volatility time series is transformed into images by applying the Markov Transition Field. The image transformation can not only preserve temporal time series relationships but also contain information on high-level features that the original time series cannot observe. Subsequently, we blur the images to reduce the image size and computational cost as discussed in the above Section 4.1.1. The MTF image matrices are used as CNN's inputs while the original RV time series is fed into the LSTM model.

Secondly, the outputs of the CNN and the LSTM are concatenated and finally, pass to a dense layer[14] to generate the output forecasts, $\widehat{RV}_t$. The main differences are as follows: Firstly, Vidal and Kristjanpoller (2020) use a pre-trained VGG-16 network[15] to extract features from images while we train CNN by our RV images. Moreover, although we both combine CNN and LSTM as the proposed model, we do not apply deep learning, such as VGG-16, because Bitcoin has a limited dataset. Lastly, the dataset they used is daily gold realised volatility. However, we aim to discover daily information from high-frequency data, therefore, our data is 10-min high-frequency Bitcoin data integrated from tick-level data pre-processed by the pre-averaging method.

4.3. Prediction accuracy

Following previous literature related to financial time series forecasting using machine learning techniques, we measure forecasting accuracy by Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Square Logarithmic Error (MSLE). The formula of each criterion is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(ln(1+y_i) - ln(1+\hat{y}_i))^2 \tag{8}$$

To make performance comparison more intuitive, we present the forecasting accuracy by the relative forecasting power between models, which idea originated from the relative RMSE in Bantis et al. (2023).[16] The relative forecasting power is calculated by

$$Relative\ Forecasting\ Power = 1 - \frac{Error_{Model}}{Error_{Benchmark\ Model}} \tag{9}$$

where the *Error* is one of the four criteria, RMSE, MAE or MSLE. The denominator is the benchmark model. By this expression, the comparisons between models can be easily revealed. Positive values indicate that the model outperforms the benchmark model. While negative values mean the benchmark model performs better.

[14] Dense layer is used as an output layer in the final stage of this model to project a vector to a one-dimensional feature space through a non-linear transformation.

[15] VGG-16 is a 16-layer CNN, which architecture developed by Simonyan and Zisserman (2014) and was used to win the ImageNet Large Scale Visual Recognition Challenge, a large-scale image classification challenge, in 2014. It is one of the most popular architectures for visual recognition.

[16] The authors compare model performance by the relative RMSE, which is the RMSE of the proposed model divided by the benchmark RMSE. If the value is lower than one, the model is more accurate.

Following Fischer and Krauss (2018)and Kim and Won (2018), we also employ (Diebold and Mariano, 2002) (DM) test to conduct pair comparisons between two models, which the null hypothesis is model i underperforms model j, where i and j are the two competing models. For robustness checks, we compare models using the Superior Predictive Ability (SPA) test (Hansen, 2005) and Model Confidence Set (MCS) introduced by Hansen et al. (2011) with Mean Squared Error (MSE). The SPA test uses bootstrap methods to assess whether one forecasting model significantly outperforms the benchmark models, the HAR model in our case. Rejecting the null hypothesis, the benchmark model is superior to the comparative model, indicating that the comparative model performs better than the HAR model. Following Feng et al. (2024) and Köchling et al. (2020) we apply the Model Confidence Set (MCS) introduced by Hansen et al. (2011). MCS is a model comparison procedure to construct a set of superior models, which starts with the full set of models and eliminates iteratively the comparably worse model. When a model rejects the null hypothesis of equal predictive ability at a given confidence level, it suggests that the model is less likely to be one of the best predictive models.

4.4. Experiment setting

The LSTM and CNN-LSTM models use Bitcoin RV and train with a set of hyper-parameters. First, we split the data into two sets, 90% of training data (for in-sample training) and 10% of testing data (for out-of-sample forecasts). With the 90% and 10% split of data, we have limited out-of-sample data with around 290 days. To address this concern, we conduct robustness checks by splitting the data into 80%/20% and 70%/30% cuts, which results are provided in Appendix D.6. For training the LSTM and CNN-LSTM network, we use Adam, Adaptive Moment Estimation, Kingma and Ba (2014) as optimiser, which has been shown to provide faster convergence and better forecasting performance for LSTM and CNN model (Ahlawat et al., 2020; Chang et al., 2018). We train the LSTM and CNN-LSTM model with hyperparameters defined as follows: the number of batch size,[17] $B \in [10, 32, 64, 128, 256, 512, 1024]$, the number of LSTM neurons,[18] $N \in [1, 5, 10, 15, 20, 25, 30]$, the activation function is Sigmoid,[19] and the number of epochs is set as 200.[20] In the following Section, we select the results for LSTM and CNN-LSTM models that yielded the minimal RMSE among models with all combinations of the specified hyperparameters.

5. Results

In this section, we present the 1-day, 7-day, 14-day, 30-day, and 60-day ahead prediction performance of our models, which we evaluate by comparing out-of-sample prediction errors. Firstly, we provide the results of the benchmark models, HAR and GARCH-type models including GARCH, TGARCH and GJR-GARCH models, followed by the forecasting performance of neural network models. Furthermore, to understand the effectiveness of the hybrid CNN-LSTM model and ensure a fair comparison to HAR model, we include the prediction accuracy of the neural networks based on inputs of the previous 1-day, previous 7-day average, and previous 30-day average. Lastly, we provide results of neural networks employing different input frequencies as robustness checks.[21]

5.1. Benchmark models results

Panel A of Table 5 shows the out-of-sample RMSE of 1-day, 7-day, 14-day, 30-day, and 60-day ahead prediction performance for HAR, GARCH, TGARCH and GJR-GARCH models.[22] We test HAR model with four different daily RV aggregated from squared pre-averaged returns at tick-level, 1-min, 5-min, and 10-min frequencies. The results indicate that HAR models outperform the GARCH models for all forecasting horizons, particularly for 1-day ahead forecasts, with an RMSE of 280.13. In contrast, the RMSEs of the GARCH models are around 335. We observe that HAR model is more suitable for modelling 1-day ahead volatility as its performance drops considerably for short-term and long-term predictions with higher RMSEs. Panel B of Table 5 provides the forecasting power of GARCH-type models relative to the HAR model, which values are all negative indicating that GARCH-type models perform worse than HAR models among all forecasting horizons, especially the 1-day predictions for all GARCH-type models, with forecasting performance circa 19% lower than HAR model. While for 7-day to 60-day ahead predictions, the HAR model outperforms GARCH-type models with RMSEs lower by 3% to 6%. However, it is not surprising that when we extend the forecasting horizon, the RMSE of HAR and GARCH-type models increase because these models are not designed for short and long-term forecasting. Informed by the results of Table 5, we only focus on the forecasting performance relative to HAR model with daily RV calculated by tick-level smoothed returns and GARCH(1, 1) models in the following discussion because HAR model dominates the GARCH-type model and the difference in performance between GARCH-type models is trivial.

[17] Batch size is the size that the number of training examples utilised in one iteration. A model with a larger batch size can train faster but may not capture the nuances in the data.

[18] The model's complexity increases as the number of neurons in a layer grows. However, the complexity is not proven to be related to the model performance. Hence, we test the model with a set of numbers.

[19] Sigmoid is one of the activation functions, which formula is $S(x) = 1/(1+e^{-x})$ where e is Euler's number. We do preliminary tests for Softmax and Sigmoid activation functions as suggested in Fischer and Krauss (2018), but we find better results when using the Sigmoid function.

[20] The number of epochs refers to the total number of iterations of the entire training data through the neural network during the training process. We set it as 200 because when we apply the early stopping technique as in Fischer and Krauss (2018), the training process normally terminates before convergence, which leads to worse predictive performance. We consider the issue might stem from Bitcoin's volatile characteristics.

[21] We also conduct tests for the sample period from 2018 to 2021 as robustness checks. The results are similar to the full sample period and are available upon request.

[22] The MAE and MSLE out-of-sample results are provided in Appendix C. We find qualitatively similar results confirming that HAR outperforms GARCH-type models.

Table 5
Benchmark model RMSE results.

Panel A: Out-of-sample RMSE						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR (tick level)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**280.13**	**315.71**	322.75	324.04	322.53
HAR (1-min)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	316.95	317.14	**317.58**	**318.62**	**320.55**
HAR (5-min)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	308.34	317.22	323.35	326.69	323.72
HAR (10-min)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	294.27	317.54	321.62	327.05	322.65
GARCH(1, 1)	Log return	335.87	335.70	335.77	336.44	337.99
GARCH(2, 1)	Log return	335.87	335.68	335.75	336.43	337.99
GARCH(2, 2)	Log return	335.92	335.63	335.58	336.16	337.55
GARCH(3, 1)	Log return	335.88	335.23	335.27	335.95	337.58
GARCH(3, 2)	Log return	335.88	335.23	335.27	335.95	337.58
TGARCH(1, 1)	Log return	336.00	336.09	336.38	337.44	339.21
TGARCH(2, 1)	Log return	336.05	336.09	336.43	337.50	339.27
TGARCH(2, 2)	Log return	336.05	336.08	336.42	337.49	339.28
TGARCH(3, 1)	Log return	336.19	336.25	336.79	337.98	339.74
TGARCH(3, 2)	Log return	336.19	336.25	336.80	338.03	339.79
GJR-GARCH(1, 1)	Log return	335.26	335.33	335.62	336.70	338.70
GJR-GARCH(2, 1)	Log return	335.27	335.31	335.61	336.70	338.70
GJR-GARCH(2, 2)	Log return	335.22	335.35	335.63	336.68	338.64
GJR-GARCH(3, 1)	Log return	335.31	334.98	335.22	336.23	338.17
GJR-GARCH(3, 2)	Log return	335.31	334.98	335.22	336.23	338.17
Panel B: Relative forecasting power (HAR based)						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR (tick level)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**0.00%**	**0.00%**	0.00%	0.00%	0.00%
HAR (1-min)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	−13.14%	−0.45%	**1.60%**	**1.67%**	**0.61%**
HAR (5-min)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	−10.07%	−0.48%	−0.19%	−0.82%	−0.37%
HAR (10-min)	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	−5.05%	−0.58%	0.35%	−0.93%	−0.04%
GARCH(1, 1)	Log return	−19.90%	−6.33%	−4.03%	−3.83%	−4.80%
GARCH(2, 1)	Log return	−19.90%	−6.33%	−4.03%	−3.82%	−4.80%
GARCH(2, 2)	Log return	−19.92%	−6.31%	−3.98%	−3.74%	−4.66%
GARCH(3, 1)	Log return	−19.90%	−6.18%	−3.88%	−3.68%	−4.67%
GARCH(3, 2)	Log return	−19.90%	−6.18%	−3.88%	−3.68%	−4.67%
TGARCH(1, 1)	Log return	−19.95%	−6.46%	−4.22%	−4.13%	−5.17%
TGARCH(2, 1)	Log return	−19.96%	−6.46%	−4.24%	−4.15%	−5.19%
TGARCH(2, 2)	Log return	−19.96%	−6.45%	−4.23%	−4.15%	−5.20%
TGARCH(3, 1)	Log return	−20.01%	−6.51%	−4.35%	−4.30%	−5.34%
TGARCH(3, 2)	Log return	−20.01%	−6.51%	−4.35%	−4.32%	−5.35%
GJR-GARCH(1, 1)	Log return	−19.68%	−6.21%	−3.99%	−3.91%	−5.01%
GJR-GARCH(2, 1)	Log return	−19.68%	−6.21%	−3.98%	−3.91%	−5.01%
GJR-GARCH(2, 2)	Log return	−19.67%	−6.22%	−3.99%	−3.90%	−5.00%
GJR-GARCH(3, 1)	Log return	−19.70%	−6.10%	−3.86%	−3.76%	−4.85%
GJR-GARCH(3, 2)	Log return	−19.70%	−6.10%	−3.86%	−3.76%	−4.85%

This table presents 1-day, 7-day, 14-day, 30-day, and 60-day ahead forecasting performance of HAR and GARCH, TGARCH, and GJR-GARCH models during the full sample period from 2014 to 2021 by calculating their Root Mean Squared Error (RMSE) of out-of-sample data reported in Panel A. We test the HAR model with four different daily realised volatility calculated by the sum of squared returns at four frequencies, namely tick level, 1-min, 5-min, and 10-min pre-averaged returns. Panel B shows the relative forecasting power compared to the HAR model with tick-level frequency, which is calculated by $1 - RMSE_{Model}/RMSE_{HAR}$ and represents the percentage difference between GARCH-type models and HAR model. Positive relative forecasting power values indicate that the model outperforms the HAR model.

Table 6
Model with 10-min frequency input data evaluated by RMSE.

Panel A: Out-of-sample RMSE						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**280.13**	315.71	322.75	324.04	322.53
LSTM	10-min RV in Past 1 day	305.83	312.89	318.70	318.65	320.49
MTF-144-CNN + LSTM	10-min RV in Past 1 day	299.77	308.25	317.08	317.75	319.76
MTF-72-CNN + LSTM	10-min RV in Past 1 day	301.13	306.76	317.05	317.78	319.75
MTF-48-CNN + LSTM	10-min RV in Past 1 day	303.11	301.07	317.07	316.67	319.75
MTF-36-CNN + LSTM	10-min RV in Past 1 day	303.09	301.85	317.04	317.62	319.73
MTF-24-CNN + LSTM	10-min RV in Past 1 day	302.77	290.28	**314.65**	318.18	319.73
MTF-18-CNN + LSTM	10-min RV in Past 1 day	307.89	305.69	317.07	**308.98**	319.75
MTF-16-CNN + LSTM	10-min RV in Past 1 day	302.95	293.75	317.06	316.97	**319.66**
MTF-12-CNN + LSTM	10-min RV in Past 1 day	304.37	300.08	317.08	317.73	319.71
MTF-9-CNN + LSTM	10-min RV in Past 1 day	307.23	310.02	317.07	316.93	319.75
MTF-8-CNN + LSTM	10-min RV in Past 1 day	308.75	286.64	317.08	317.63	319.67
MTF-6-CNN + LSTM	10-min RV in Past 1 day	305.05	**284.88**	317.04	317.60	319.83
MTF-4-CNN + LSTM	10-min RV in Past 1 day	305.44	284.95	317.08	317.31	319.73
GARCH(1, 1)	Log return	335.87	335.70	335.77	336.44	337.99
Panel B: Relative forecasting power (HAR based)						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**0.00%**	0.00%	0.00%	0.00%	0.00%
LSTM	10-min RV in Past 1 day	−9.17%	0.89%	1.26%	1.66%	0.63%
MTF-144-CNN + LSTM	10-min RV in Past 1 day	−7.01%	2.36%	1.76%	1.94%	0.86%
MTF-72-CNN + LSTM	10-min RV in Past 1 day	−7.50%	2.83%	1.77%	1.93%	0.86%
MTF-48-CNN + LSTM	10-min RV in Past 1 day	−8.20%	4.64%	1.76%	2.27%	0.86%
MTF-36-CNN + LSTM	10-min RV in Past 1 day	−8.20%	4.39%	1.77%	1.98%	0.87%
MTF-24-CNN + LSTM	10-min RV in Past 1 day	−8.08%	8.05%	**2.51%**	1.81%	0.87%
MTF-18-CNN + LSTM	10-min RV in Past 1 day	−9.91%	3.17%	1.76%	**4.65%**	0.86%
MTF-16-CNN + LSTM	10-min RV in Past 1 day	−8.15%	6.96%	1.76%	2.18%	**0.89%**
MTF-12-CNN + LSTM	10-min RV in Past 1 day	−8.65%	4.95%	1.76%	1.95%	0.87%
MTF-9-CNN + LSTM	10-min RV in Past 1 day	−9.68%	1.80%	1.76%	2.19%	0.86%
MTF-8-CNN + LSTM	10-min RV in Past 1 day	−10.22%	9.21%	1.76%	1.98%	0.88%
MTF-6-CNN + LSTM	10-min RV in Past 1 day	−8.90%	**9.77%**	1.77%	1.99%	0.84%
MTF-4-CNN + LSTM	10-min RV in Past 1 day	−9.04%	9.74%	1.76%	2.08%	0.87%
GARCH(1, 1)	Log return	−19.90%	−6.33%	−4.03%	−3.83%	−4.80%

This table presents the model comparisons between the HAR model and neural network models, including LSTM and CNN-LSTM models. Panel A is evaluated by Root Mean Squared Error (RMSE) of out-of-sample data. Panel B presents the model performance measured by the relative forecasting power calculated by one minus RMSE of each model divided by RMSE of the HAR model. Positive values indicate that the model outperforms the HAR model. While negative values mean the HAR model performs better. The models are named with MTF-size-CNN+LSTM, which size means CNN input image size. For example, the input image size of the MTF-144-CNN+LSTM model is 144*144. Input series explanation: (1) 10-min RV in the past 1 day means that the model forecasts days ahead based on 10-min RV in the previous 1-day. The amount of data for 10-min RV in 1 day is 144 because there are 144 10-min daily. (2) Past 7 and 30 day average RV are calculated by rolling 7 and 30-day average realised volatility.

5.2. Model performance comparisons

5.2.1. Comparisons with benchmark models

Table 6 presents the relative forecasting power and RMSE of the hybrid CNN-LSTM model compared to HAR model.[23] There are three main results: first of all, it is noticeable that by employing 10-min frequency data instead of using standard inputs in the HAR model, 1-day, 7-day average, and 30-day average data, or feeding log returns as in GARCH-type models, the CNN-LSTM models rank at the top and outperform LSTM in 7-day to 60-day ahead prediction. Therefore, the CNN-LSTM model that uses high-frequency data outperforms models that use daily data. Furthermore, our model is a better choice for short-term volatility forecasting, especially in 7-day ahead forecasting. The best performance for the 7-day ahead prediction generated by the hybrid CNN-LSTM model is 9.77% higher than the HAR model. However, the HAR model with daily data outperforms the neural network model with high-frequency data for 1-day ahead prediction.

Second, compared to LSTM, the hybrid CNN-LSTM exploits the benefits of image classification on MTF images encoded by the transition probability information. The result shows that the prediction accuracy of the hybrid CNN-LSTM model along with image

[23] Appendix D.1 and Appendix D.2 provide MAE and MSLE results as well as the 1-day, 7-day, 14-day, 30-day, and 60-day ahead forecasting power of our model compared to HAR model and its ranking by the order of RMSE, MAE, and MSLE, respectively. The average rank is the order of the average four rankings. If the models have the same average rank, we then rank them by RMSE.

transformation can be improved in all forecasting targets. This is because the high-level features, such as 2-dimensional temporal dependency, can be discovered by CNN with time series to image transformation. However, the performance of CNN-LSTM models that make use of different sizes of images varies across different forecasting horizons.[24] The highest predictive power is achieved with an image size of 144*144 for 1-day ahead, 6*6 for 7-day ahead, 24*24 for 14-day ahead, 18*18 for 30-day ahead, and 16*16 for 60-day ahead prediction. Therefore, we cannot conclude which image size can be the best estimator among different forecasting horizons. Nevertheless, we find the improved performance of the CNN-LSTM model when utilising larger image sizes for one-day ahead forecasts. For 7-day to 60-day ahead prediction, the predictive power of the model with the reduced image, 16*16, stays in the top 5 when considering the average rank. This suggests that the original MTF image with size reduction can provide better forecasts. We consider the reason to be the effectiveness of noise reduction by both the pre-averaging method and image size reduction. Lastly, our model outperforms GARCH-type models by improvements ranging from 5% to 15% of RMSE, consistent with most of the literature on neural networks application in volatility forecasting, although the difference is relatively small compared to other assets, such as Gold (Vidal and Kristjanpoller, 2020).

Appendix D.3 presents the DM test results for 14-day, 30-day, and 60-day ahead forecasting performance as a robustness check.[25] Most of the pair comparisons between CNN-LSTM and the GARCH(1, 1) model can significantly reject the null hypothesis, indicating that the CNN-LSTM model performs better than the GARCH(1, 1) model. In the case of the HAR model, null hypothesis rejection is pronounced, particularly when considering the 30-day forecasting horizon. This is consistent with the results from RMSE and the average ranking shown in Table 6 and Appendix D.2, respectively. Appendix D.4 presents SPA results, showing the consistent results that CNN-LSTM models are superior to the HAR model in short-term and long-term predicting horizons.

Appendix D.5 provides MCS results, which are in general consistent with our findings. Additionally, we find that except for the 14-day ahead forecasting horizon, we can hardly exclude any model during the MCS procedure, indicating that most of the models possess distinctive information relevant to Bitcoin volatility forecasting. However, only CNN-LSTM with an image size of 24, LSTM, and HAR models construct superior model set in 14-day ahead volatility prediction, meaning that these models contain the most information needed in this regard. Although we cannot significantly exclude the HAR model from the model confidence set, the HAR model cannot significantly beat the CNN-LSTM models. Further, the HAR model performance drops when the horizon increases and the CNN-LSTM models with an MCS p-value of 1 are more likely to be one of the best models. Moreover, the average rank calculated from four error measurements (Appendix D.2) and the SPA robustness check (Appendix D.4) show that the CNN-LSTM models perform better than the HAR model from 7 to 60-day prediction horizons. We believe these tests show CNN-LSTM can outperform the HAR model in short and long-term forecasts. Moreover, to address the concern of limited out-of-sample datapoints, we also conduct robustness checks with different training and testing data cuts, with 80%/ 20% and 70%/ 30% proportions. Appendix D.6 shows the model outperforms at 7-day ahead predictions and is consistent with our main findings.

Nevertheless, the model with 10-min frequency data underperforms the HAR model in 1-day ahead volatility forecasting, thus, Table 7 compares the performance of the neural networks to the HAR model, ceteris paribus, to make fair comparisons.[26] By giving the same information set as possible, LSTM outperforms the HAR model across all forecasting targets with particularly notable results in 7-day ahead prediction, where LSTM has a 1.56% improvement over the HAR model. Moreover, we include MTF images derived from the past 7-day and 30-day RV time series into the CNN-LSTM model, which means the images have information on the past 7-day and 30-day transition probability, respectively. Surprisingly, this model performs better than both HAR and LSTM with 1-day, 7-day average, and 30-day average inputs only in predicting 7-day to 60-day ahead volatility circa 1%–2%. This indicates that the transition probability for the past 7-day and 30-day contains valuable information that can benefit volatility predictions beyond one-week. Moreover, for 1-week to 2-month ahead prediction, rather than using MTF generated from past days' RV, 10-min RV is more beneficial when we compare the results of Tables 7 to 6, meaning high-frequency RV is a better choice when predicting short-term and long-term Bitcoin volatility compared to daily RV.

Table 8 presents the DM test for model pair comparisons. We find the results consistent with Table 7. All the models outperform GARCH(1, 1) models in terms of prediction accuracy. Moreover, the CNN-LSTM model augmented with past 7-day and 30-day transition probability images demonstrates superior performance in short-term forecasts, ranging from 14-day to 60-day ahead predictions, compared to the LSTM model. These results highlight the potential benefits of incorporating image-based features derived from past time series data into forecasting models, particularly for short-term forecasting objectives. Moreover, we present MCS results in Appendix E.3, which align with our findings. For 1-day ahead prediction, LSTM is one of the superior models compared to other models, evidenced by the rejection of the null hypothesis denoting that the models are excluded from the model confidence set. While for 1-week to 2-month forecasting horizons, the GARCH(1, 1) model rejects the null hypothesis, suggesting that it is unlikely to be one of the top-performing predictive models.

[24] MTF is the transition probability from a 10-min RV to another, thus, the original image size of MTF is 144*144 for 10-min RV because there are 144 10-min in a day. The blurred images are compressed from the 144*144 images, which image size has to be the divisor of the original image size. For example, because the original 10-min MTF is 144*144, the reduced image size has to be the divisors of 144.

[25] We discuss DM test results for 14-day to 60-day ahead forecasts because the predictive performance of RMSEs superior in 14-day to 60-day. While 1-day and 7-day DM test results are also provided in Appendix D.3.

[26] Appendix E.1 and Appendix E.2 tabulate MAE and MSLE results as well as the 1-day, 7-day, 14-day, 30-day, and 60-day ahead forecasting power of our model compared to the HAR model and its ranking by the order of RMSE, MAE, and MSLE, respectively. The average rank is the order of the average four rankings. If the models have the same average rank, we then rank them by RMSE rank.

Table 7
Model with input upon ceteris paribus as HAR model evaluated by RMSE.

Panel A: Out-of-sample RMSE						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	280.13	315.71	322.75	324.04	322.53
LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**278.85**	**310.79**	319.43	319.45	320.33
MTF-30-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 30 day MTF	284.52	312.75	**316.29**	318.15	**318.20**
MTF-7-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 7 day MTF	298.60	312.37	317.02	**318.00**	318.54
GARCH(1, 1)	Log return	335.87	335.70	335.77	336.44	337.99
Panel B: Relative forecasting power (HAR based)						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	0.00%	0.00%	0.00%	0.00%	0.00%
LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**0.46%**	**1.56%**	1.03%	1.42%	0.68%
MTF-30-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 30 day MTF	−1.57%	0.94%	**2.00%**	1.82%	**1.34%**
MTF-7-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 7 day MTF	−6.60%	1.06%	1.78%	**1.86%**	1.24%
GARCH(1, 1)	Log return	−19.90%	−6.33%	−4.03%	−3.83%	−4.80%

This table presents the results of our model compared to the HAR model, ceteris paribus. Panel A is evaluated by Root Mean Squared Error (RMSE) of out-of-sample data. Panel B presents the model performance by the relative forecasting power calculated by one minus RMSE of each model divided by the RMSE of the HAR model, respectively. Positive values indicate that the model outperforms the HAR model. While negative values mean HAR performs better. The models named MTF-7-CNN+LSTM and MTF-30-CNN+LSTM are the CNN-LSTM models with MTF image inputs transformed from past daily RV time series, where the images include previous 7-day and 30-day information, respectively. Input series explanation: Past 1 day RV is the realised volatility of the past 1 day. Past 7 and 30 day average RV are calculated by rolling 7 and 30-day average realised volatility.

5.2.2. Comparisons between different frequencies

In this section, we test the impact of inputs with different frequencies on the performance of our models. In addition, we want to know whether there is a suggested sampling frequency choice for the proposed model. Table 9 compares the relative performance power of LSTM given different inputs including 5-min, 10-min, 30-min, and 1-h RV in the past 1 day. As discussed, the LSTM with HAR model inputs outperforms the HAR model in all forecasting tasks with all positive relative forecasting power figures and ranks at the top,[27] especially for 14-day ahead prediction. This suggests that by giving the same information set, the neural network extracts higher-order features that the HAR model cannot, leading to higher forecasting performance. Moreover, the LSTM employing the 30-min RV ranks top in 7-day forecasting horizons, while the model using the HAR model's input, 5-min and 1-h RV in the past one day outperforms other models in 30-day and 60-day forecasting horizons, respectively. Hence, although there is no one dominant frequency data among others in terms of predictive performance, the LSTM model integrating high-frequency RV can provide more accurate long-term forecasts when compared to LSTM relying on daily RV.

Generally, regarding the one-week to one-month ahead forecasts, LSTM has higher prediction accuracy compared to the HAR model by 1% to 2%. We also compare CNN-LSTM model performance by giving different time series and MTF image inputs, which results are presented in Table 10.[28] The CNN-LSTM model with high-frequency data outperforms the HAR model when forecasting 7-day to 60-day ahead volatility, especially for 7-day ahead prediction, with forecasting power 3% to 5% higher than the HAR model. The short-term predictive outperformance indicates that the model can capture temporal dependencies in Bitcoin volatility. Our analysis reflects the Bitcoin market response time to information, suggesting that Bitcoin investors require a week or longer to fully react to information. Furthermore, the CNN-LSTM model with high-frequency data dominates the model with daily data

[27] The ranking for LSTM with various frequencies is provided in Appendix F.2.
[28] The MAE and MSLE and rankings are provided in Appendix G.1 and Appendix G.2, respectively.

Table 8
DM test for different days ahead forecasting performance of various models with HAR model's input.

1-day ahead	i	j=	HAR	LSTM	MTF-30-CNN + LSTM	MTF-7-CNN + LSTM	GARCH(1, 1)
	HAR		–	0.8261	0.3653	0.0637*	0.0048***
	LSTM		0.1739	–	0.3264	0.0530*	0.0048***
	MTF-30-CNN + LSTM		0.6347	0.6736	–	0.0154**	0.0110**
	MTF-7-CNN + LSTM		0.9363	0.9470	0.9846	–	0.0158**
	GARCH(1, 1)		0.9952	0.9952	0.9890	0.9842	–
7-day ahead	i	j=	HAR	LSTM	MTF-30-CNN + LSTM	MTF-7-CNN + LSTM	GARCH(1, 1)
	HAR		–	0.7886	0.8160	0.8239	0.0693*
	LSTM		0.2114	–	0.2969	0.3047	0.0915*
	MTF-30-CNN + LSTM		0.1840	0.7031	–	0.5535	0.0652*
	MTF-7-CNN + LSTM		0.1761	0.6953	0.4465	–	0.0812*
	GARCH(1, 1)		0.9307	0.9085	0.9348	0.9188	–
14-day ahead	i	j=	HAR	LSTM	MTF-30-CNN + LSTM	MTF-7-CNN + LSTM	GARCH(1, 1)
	HAR		–	0.7680	0.8964	0.8255	0.1086
	LSTM		0.2320	–	0.7870	0.6689	0.0238**
	MTF-30-CNN + LSTM		0.1036	0.2130	–	0.3301	0.0535*
	MTF-7-CNN + LSTM		0.1745	0.3311	0.6699	–	0.0846*
	GARCH(1, 1)		0.8914	0.9762	0.9465	0.9154	–
30-day ahead	i	j=	HAR	LSTM	MTF-30-CNN + LSTM	MTF-7-CNN + LSTM	GARCH(1, 1)
	HAR		–	0.8744	0.8988	0.9151	0.1471
	LSTM		0.1256	–	0.6669	0.7173	0.0549*
	MTF-30-CNN + LSTM		0.1012	0.3331	–	0.6207	0.0888*
	MTF-7-CNN + LSTM		0.0849*	0.2827	0.3793	–	0.0795*
	GARCH(1, 1)		0.8529	0.9451	0.9112	0.9205	–
60-day ahead	i	j=	HAR	LSTM	MTF-30-CNN + LSTM	MTF-7-CNN + LSTM	GARCH(1, 1)
	HAR		–	0.9701	0.8554	0.7915	0.0829*
	LSTM		0.0299**	–	0.7542	0.6770	0.0731*
	MTF-30-CNN + LSTM		0.1446	0.2458	–	0.3387	0.0964*
	MTF-7-CNN + LSTM		0.2085	0.3230	0.6613	–	0.1116
	GARCH(1, 1)		0.9171	0.9269	0.9036	0.8884	–

This table provides the Diebold and Mariano (DM test) p-values for pair comparisons between benchmark models and machine learning models among different forecasting horizons. The DM test evaluates the null hypothesis: the forecasting accuracy of model i underperforms model j. Upon ceteris paribus as the HAR model, LSTM takes inputs of the past 1-day RV, 7-day average RV and 30-day average RV. The MTF-7-CNN+LSTM and MTF-30-CNN+LSTM are the CNN-LSTM models combined by the LSTM part with HAR-like inputs and the CNN part with MTF images transformed from past daily RV time series, where the images include previous 7-day and 30-day information, respectively.

* $p < 0.1$.
** $p < 0.05$.
*** $p < 0.01$.

when forecasting one-week and one-month ahead volatility. As shown in Tables 9 and 10, we cannot find a consistent choice of frequencies as inputs for forecasting both short and long-term, which is not in line with the existing literature that suggests 5-min frequency (Liu et al., 2015; Shen et al., 2020). Nevertheless, we find that using MTF images transformed from high-frequency data increases the forecasting accuracy, indicating that unrevealed features are covered in the images that the original time series does not have, even though the image is transformed from the original time series. To conclude, the proposed hybrid model generally outperforms the HAR model in short-term forecasts, especially for 7-day ahead prediction.

6. Discussion

In this paper, we show that MTF images transformed from high-frequency realised volatility can benefit short-term and long-term volatility precision tasks. As stated in Wang et al. (2015), MTF image encodes transitional probability between timesteps, which unfortunately is difficult to trace back to understand the exact characteristics that increase the predictive power. Nevertheless, we consider the information provided by MTF images from several perspectives. First, MTF images can capture how frequently the time series resides in particular levels of volatility. Thus, if the time series often fluctuates around a specific volatility level during a period, the transitions to and from the state will dominate the MTF and highlight the level in the image. In Fig. 3, we can also observe the outperformance originating from this regard. The CNN-LSTM prediction errors, the difference between the predicted and true RV, are smaller when more jumps cluster together but larger when a sudden jump exists, indicating the model performs better when the volatility clusters in a period rather than sudden jumps. Second, the MTF image indirectly reflects the moments of volatility. For the second moment, variance, a wider spread in the transition probabilities suggests a higher variance in the realised volatility, indicating that the series frequently moves between a broader range of states. Moreover, the MTF images can show skewness when the MTF image is biased, for instance, when the volatility predominantly transitions to higher levels rather than returning to lower

Table 9
LSTM RMSE performance comparisons between different frequencies.

Panel A: Out-of-sample RMSE						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	280.13	315.71	322.75	324.04	322.53
LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**278.85**	310.79	319.43	319.45	320.33
LSTM	Past 1 day RV	286.08	310.09	**317.14**	316.11	**319.58**
LSTM	1-h RV in Past 1 day	303.09	307.09	319.83	**314.82**	320.94
LSTM	30-min RV in Past 1 day	305.83	**305.44**	319.58	318.30	321.71
LSTM	10-min RV in Past 1 day	305.83	312.89	318.70	318.65	320.49
LSTM	5-min RV in Past 1 day	306.50	313.18	318.92	317.31	321.27
GARCH(1, 1)	Log return	335.87	335.70	335.77	336.44	337.99
Panel B: Relative forecasting power (HAR based)						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	0.00%	0.00%	0.00%	0.00%	0.00%
LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**0.46%**	1.56%	1.03%	1.42%	0.68%
LSTM	Past 1 day RV	−2.12%	1.78%	**1.74%**	2.45%	**0.92%**
LSTM	1-h RV in Past 1 day	−8.20%	2.73%	0.91%	**2.85%**	0.49%
LSTM	30-min RV in Past 1 day	−9.17%	**3.25%**	0.98%	1.77%	0.25%
LSTM	10-min RV in Past 1 day	−9.17%	0.89%	1.26%	1.66%	0.63%
LSTM	5-min RV in Past 1 day	−9.41%	0.80%	1.19%	2.08%	0.39%
GARCH(1, 1)	Log return	−19.90%	−6.33%	−4.03%	−3.83%	−4.80%

This table compares the LSTM model forecasting performance with different frequency data as inputs evaluated by Root Mean Squared Error (RMSE) of out-of-sample data. The relative forecasting power is calculated by one minus RMSE of HAR model divided by RMSE of HAR model. Positive values indicate that the model outperforms HAR model. While negative values mean HAR performs better. Input series explanation: (1) 5-min/10-min/30-min/1-h RV in the past 1 day means that the model forecasts days ahead based on 5-min/10-min/30-min/1-h RV in the previous 1-day. (2) Past 7 and 30 day average RV are calculated by rolling 7 and 30-day average realised volatility.

ones. This can indicate the asymmetry effect towards one tail in the volatility time series. MTF may also indicate high kurtosis if the volatility more frequently transitions to extreme states (either high or low) than expected in a normal distribution, implying the presence of heavy tails in the volatility time series. As the summary statistics in Table 4, the Bitcoin volatility exhibits heavy tails and a skewed distribution. We consider the MTF image may be able to capture the heavy tails and a skewed distribution as well as the volatility clustering effects, leading to outperformance in the volatility prediction.

However, an empirical investigation of the link between statistical properties of the volatility time series distribution and the extracted MTF transitional probabilities goes beyond the scope of this paper. Besides, we compare the RMSE of the HAR and CNN-LSTM models to gain insights into why the CNN-LSTM model excels in both short-term and long-term forecasts. Using 1-day and 7-day ahead forecasting errors as examples, Fig. 4 presents the RMSE for both HAR and CNN-LSTM models. The results indicate that while HAR generally performs better in 1-day forecasts, the CNN-LSTM model outperforms in 7-day forecasts. We can observe that in 1-day ahead forecasting, the HAR model is more effective at handling Bitcoin's volatility spikes, likely originating from its design, which incorporates 7-day and 30-day average volatility to mitigate the impact of sudden changes. However, for 7-day ahead forecasts, the CNN-LSTM model can reduce the influence of the volatility spikes compared to 1-day ahead predictions. Moreover, the CNN-LSTM model surpasses the HAR model, particularly when dealing with volatility clustering, which the HAR model struggles to capture. This advantage is due to the memory unit of the LSTM, which allows the CNN-LSTM model to outperform in both short-term and long-term forecasts.

Further, we do not extend our study to other cryptocurrencies for several reasons. First, neural networks build models based on the training dataset, which is limited to other cryptocurrencies. We find literature using Ethereum data starting around 2018 (Feng et al., 2024; Wang et al., 2023). However, because we take 10-min volatility time series as our model input instead of daily data, we need to consider the liquidity issue. In Section 2 and Table 2, we quantify liquidity by calculating the percentage of zero transactions in each year. Even though we can trace the tick-level trades back to 2011, we find Bitcoin is more liquid after 2014, where the percentage drops to 0.24%. Other cryptocurrencies are not as Bitcoin having accessible and liquid high-frequency data from 2014. For example, the high-frequency Ethereum data covered in Brauneis et al. (2021) and Feng et al. (2024) is from around 2018, which is half of our Bitcoin dataset. Hence, we do not have sufficient training and testing data for neural networks. Second, Bitcoin covered 50% of market capitalisation and there is evidence of cryptocurrency comovement after 2017 (De Pace and Rao, 2023), which we consider Bitcoin can represent the market. Future studies in a few years time, where longer sample periods for other cryptocurrencies are available and the market is more liquid, would be more appropriate, but at this time, is infeasible.

Table 10
CNN-LSTM RMSE performance comparisons between different frequencies.

Panel A: Out-of-sample RMSE						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**280.13**	315.71	322.75	324.04	322.53
MTF-30-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 30 day MTF	284.52	312.75	**316.29**	318.15	**318.20**
MTF-7-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 7 day MTF	298.60	312.37	317.02	318.00	318.54
MTF-24-CNN + LSTM	1-h RV in Past 1 day	295.16	306.14	317.07	316.55	318.95
MTF-48-CNN + LSTM	30-min RV in Past 1 day	297.60	308.50	317.08	**315.37**	319.68
MTF-144-CNN + LSTM	10-min RV in Past 1 day	299.77	308.25	317.08	317.75	319.76
MTF-288-CNN + LSTM	5-min RV in Past 1 day	302.59	**301.84**	317.07	316.03	319.73
GARCH(1, 1)	Log return	335.87	335.70	335.77	336.44	337.99
Panel B: Relative forecasting power (HAR based)						
Model	Input	1-day ahead	7-day ahead	14-day ahead	30-day ahead	60-day ahead
HAR	Past 1 day RV + Past 7 day average RV + Past 30 day average RV	**0.00%**	0.00%	0.00%	0.00%	0.00%
MTF-30-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 30 day MTF	−1.57%	0.94%	**2.00%**	1.82%	**1.34%**
MTF-7-CNN + LSTM	Past 1 day RV + Past 7 day average RV + Past 30 day average RV and Past 7 day MTF	−6.60%	1.06%	1.78%	1.86%	1.24%
MTF-24-CNN + LSTM	1-h RV in Past 1 day	−5.37%	3.03%	1.76%	2.31%	1.11%
MTF-48-CNN + LSTM	30-min RV in Past 1 day	−6.24%	2.28%	1.76%	**2.68%**	0.88%
MTF-144-CNN + LSTM	10-min RV in Past 1 day	−7.01%	2.36%	1.76%	1.94%	0.86%
MTF-288-CNN + LSTM	5-min RV in Past 1 day	−8.02%	**4.39%**	1.76%	2.47%	0.87%
GARCH(1, 1)	Log return	−19.90%	−6.33%	−4.03%	−3.83%	−4.80%

This table compares the CNN-LSTM model forecasting performance with different frequency RV and corresponding MTF images as inputs, which is evaluated by Root Mean Squared Error (RMSE) of out-of-sample data. The relative forecasting power is calculated by one minus RMSE of HAR model divided by RMSE of HAR model. Positive values indicate that the model outperforms HAR model. While negative values mean HAR performs better. Input series explanation: (1) 5-min/10-min/30-min/1-h RV in the past 1 day means that the model forecasts days ahead based on 5-min/10-min/30-min/1-h RV in the previous 1-day. The amount of data for 5-min/10-min/30-min/1-h RV in 1-day are 288, 144, 48, and 24, respectively. Therefore, the image sizes transformed from the RV time series are also 288*288, 144*144, 48*48, and 24*24, respectively. (2) Past 7 and 30 day average RV are calculated by rolling 7 and 30-day average realised volatility. MTF-7-CNN+LSTM and MTF-30-CNN+LSTM are the CNN-LSTM models with MTF image inputs transformed from past daily RV time series, where the images include previous 7-day and 30-day information, respectively.

7. Conclusion

Bitcoin operates as a 24/7 global trading market, distinguishing it from traditional financial markets and rendering it highly volatile. The growing participation of institutional investors highlights the necessity for precise Bitcoin volatility prediction to effectively manage the risk of investments. In this paper, with Bitcoin's 24/7 characteristics, we utilise Bitcoin's high-frequency RV to enhance volatility prediction.

We apply popular machine learning techniques, LSTM and CNN, to forecast days ahead volatility of Bitcoin based on high-frequency data from 2014 to 2021 and compare to the traditional models including HAR and GARCH-type models. There are two main contributions in this paper: first of all, we add evidence to neural network applications in financial time series prediction. We show that neural networks dominate the GARCH-type model, with performance improvements ranging from 5% to 15%, which is consistent with previous literature. However, compared to the HAR model, only LSTM with the HAR model's inputs can outperform all predicting time horizons. The hybrid CNN-LSTM model with high-frequency data inputs provides superior results particularly in one week ahead forecasting horizons, achieving 9.77% of improvement compared to HAR. Secondly, we explore the potential of time series to image transformation, using the Markov Transition Field (MTF). In previous literature, LSTM is a widely used neural network in financial time series prediction since it can capture memory effects in time series. In addition to LSTM, we also combine CNN with a Dense layer which takes MTF images transformed from time series as inputs. We find that the forecasting performance is improved by including MTF images, which means that time series to image transformations can capture the 2-dimension temporal dependencies different from the raw time series. Moreover, we show that high-frequency RV is better than daily data for short-term and long-term Bitcoin volatility prediction.

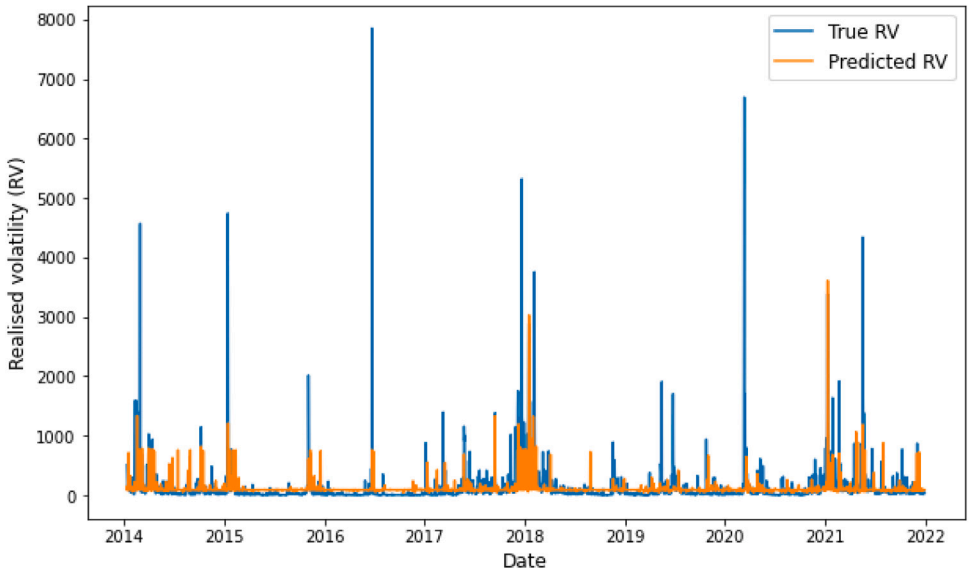

Fig. 3. CNN-LSTM 7-day RMSE overtime. This graph shows the CNN-LSTM prediction errors across our sample period. CNN-LSTM models are superior at a 7-day ahead forecasting horizon and the CNN-LSTM model with an input image size of 16 outperforms other CNN-LSTM models, therefore, we select the model as an example to discuss the under- and out-performance. We observe that the underperformance mainly happens when there is a sudden huge volatility jump in RV. Nevertheless, when more jumps cluster together, the model tends to well capture the effect.

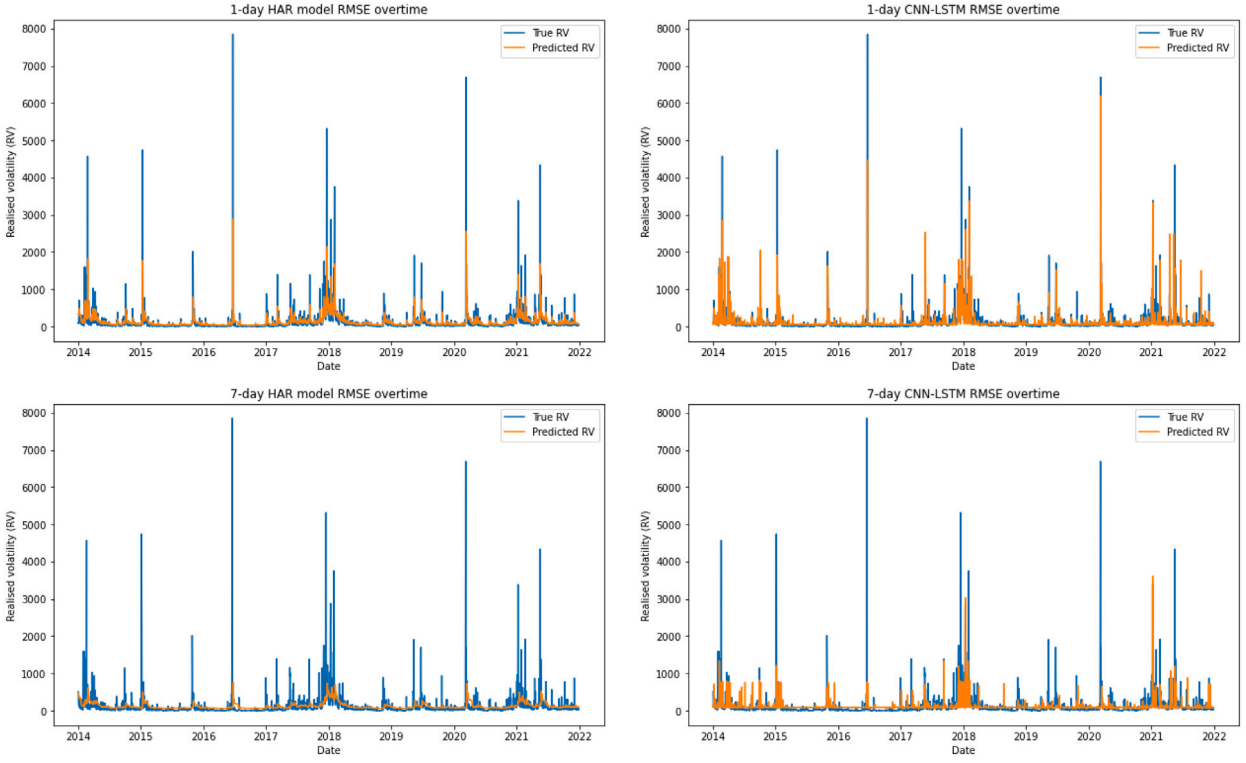

Fig. 4. HAR and CNN-LSTM performance visualisation. This graph shows the HAR and CNN-LSTM model prediction errors across our sample period. We select CNN-LSTM model with an input image size of 16 as in Fig. 3 as an example to discuss the under- and out-performance compared to the HAR model. We observe that the HAR model is better at handling volatility spikes in 1-day ahead forecasts, while the CNN-LSTM model outperforms in dealing with the volatility clustering effect.

Overall, our study adds to the existing literature which suggests that neural networks have the potential to outperform the GARCH model. We find that our neural network model is superior in 7-day ahead forecasts and discover the potential of the time series to image transformation for high-frequency data in comparison to the HAR model. The 7-day ahead predictive superiority suggests that the CNN-LSTM models with their ability to capture transition probabilities patterns and temporal time series dependencies, are particularly useful at forecasting volatility clustering in the weekly Bitcoin market. The effectiveness also reflects the Bitcoin market's response time to information and indicates that investors need time to interpret and react to such information fully. Given that Bitcoin is a relatively new and highly volatile financial asset attracting significant retail investor interest, the response time of a week or even longer is not surprising.

Our analyses reveal important implications. From a financial economics perspective, these results underscore the utility of advanced machine learning models in capturing the complex dynamics of the volatility in cryptocurrency markets, where traditional models may fall short. First, neural networks can effectively handle nonlinear relationships among high-frequency Bitcoin time series, which traditional models may miss. The capability can result in more accurate predictions, especially in the volatile Bitcoin market. Second, machine learning models can incorporate different types of inputs compared to traditional models, such as MTF images that provide information on transition probability, which cannot be directly observed in the ordinary time series. The integration can improve out-of-sample forecasting accuracy, highlighting the hybrid CNN-LSTM model's ability to adapt to changing market conditions. As a result, our analyses of popular neural networks with high-frequency Bitcoin RV can improve predictive accuracy, leading to support for Bitcoin traders, institutional investors and policymakers from several perspectives. Accurate volatility predictions can help traders formulate trading strategies and make informed trading decisions. Furthermore, because Bitcoin is highly volatile compared to traditional financial assets, precise volatility forecasts can reduce the risk of unexpected losses from a risk management perspective. For instance, investors can more precisely hedge volatility, leading to more efficient risk management and asset allocation (Wang et al., 2023). Traders can incorporate CNN-LSTM volatility forecasts into short-term trading strategies, allowing for more responsive and precise position adjustments during periods of anticipated volatility. Financial institutions can design more accurate derivatives linked to Bitcoin price volatility (Wang et al., 2023). Furthermore, investors could integrate these improved forecasts into their portfolio optimisation processes, leading to better asset allocation decisions and improved risk-adjusted returns. Finally, this paper can assist policymakers in monitoring the cryptocurrency market more effectively by providing more accurate forecasts, allowing them to prevent bubbles, and supporting and strengthening the financial stability of cryptocurrencies. Accurate forecasts can help reduce the likelihood of market disruptions by allowing for timely interventions during periods of anticipated market stress. Our research can also help to enhance further regulatory frameworks aimed at protecting investors.

CRediT authorship contribution statement

Zih-Chun Huang: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Ivan Sangiorgi:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Andrew Urquhart:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

Acknowledgements

The authors thank and appreciate the valuable feedback from discussants and participants in the Cryptocurrency Research Conference 2022.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.intfin.2024.102064.

Data availability

The authors do not have permission to share data.

References

Aggarwal, D., Chandrasekaran, S., Annamalai, B., 2020. A complete empirical ensemble mode decomposition and support vector machine-based approach to predict bitcoin prices. J. Behav. Exp. Finance 27, 100335.
Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., Yoon, B., 2020. Improved handwritten digit recognition using convolutional neural networks (CNN). Sensors 20 (12), 3344.
Alexander, C., Heck, D.F., Kaeck, A., 2022. The role of binance in bitcoin volatility transmission. Appl. Math. Finance 29 (1), 1–32.
Andersen, T.G., Bollerslev, T., 1997. Intraday periodicity and volatility persistence in financial markets. J. Empir. Financ. 4 (2–3), 115–158.
Andersen, T.G., Bollerslev, T., 1998. Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. Internat. Econom. Rev. 39 (4), 885–905.
Andersen, T.G., Bollerslev, T., Diebold, F.X., Labys, P., 1999. Realized Volatility and Correlation. LN Stern School of Finance Department Working Paper 24.
Andersen, T.G., Bollerslev, T., Diebold, F.X., Labys, P., 2003. Modeling and forecasting realized volatility. Econometrica 71 (2), 579–625.
Aras, S., 2021a. On improving GARCH volatility forecasts for bitcoin via a meta-learning approach. Knowl.-Based Syst. 230, 107393.
Aras, S., 2021b. Stacking hybrid GARCH models for forecasting bitcoin volatility. Expert Syst. Appl. 174, 114747.
Ardia, D., Bluteau, K., Rüede, M., 2019. Regime changes in bitcoin GARCH volatility dynamics. Finance Res. Lett. 29, 266–271.
Atsalakis, G.S., Atsalaki, I.G., Pasiouras, F., Zopounidis, C., 2019. Bitcoin price forecasting with neuro-fuzzy techniques. European J. Oper. Res. 276 (2), 770–780.

Aysan, A.F., Caporin, M., Cepni, O., 2024. Not all words are equal: Sentiment and jumps in the cryptocurrency market. J. Int. Financ. Mark. Inst. Money 91, 101920.
Bantis, E., Clements, M.P., Urquhart, A., 2023. Forecasting GDP growth rates in the United States and Brazil using google trends. Int. J. Forecast. 39 (4), 1909–1924.
Bariviera, A.F., 2017. The inefficiency of bitcoin revisited: A dynamic approach. Econom. Lett. 161, 1–4.
Baur, D.G., Hong, K., Lee, A.D., 2018. Bitcoin: Medium of exchange or speculative assets?. J. Int. Financ. Mark. Inst. Money 54, 177–189.
Bergsli, L.Ø., Lind, A.F., Molnár, P., Polasik, M., 2022. Forecasting volatility of bitcoin Res. Int. Bus. Finance 59, 101540.
Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity J. Econometrics 31 (3), 307–327.
Bouri, E., Gil-Alana, L.A., Gupta, R., Roubaud, D., 2019. Modelling long memory volatility in the bitcoin market: Evidence of persistence and structural breaks Int. J. Finance Econ. 24 (1), 412–426.
Brauneis, A., Mestel, R., Riordan, R., Theissen, E., 2021. How to measure the liquidity of cryptocurrency markets? J. Bank. Financ. 124, 106041.
Catania, L., Grassi, S., 2022. Forecasting cryptocurrency volatility Int. J. Forecast. 38 (3), 878–894.
Catania, L., Sandholdt, M., 2019. Bitcoin at high frequency J. Risk Financ. Manag. 12 (1), 36.
Chang, Z., Zhang, Y., Chen, W., 2018. Effective adam-optimized LSTM neural network for electricity price forecasting In: 2018 IEEE 9th International Conference on Software Engineering and Service Science. ICSESS, IEEE, pp. 245–248.
Chen, W., Xu, H., Jia, L., Gao, Y., 2021. Machine learning model for bitcoin exchange rate prediction using economic and technology determinants Int. J. Forecast. 37 (1), 28–43.
Chi, Y., Hao, W., 2021. Volatility models for cryptocurrencies and applications in the options market J. Int. Financ. Mark. Inst. Money 75, 101421.
Christensen, B.J., Hansen, C.S., 2002. New evidence on the implied-realized volatility relation Eur. J. Finance 8 (2), 187–205.
Corsi, F., 2009. A simple approximate long-memory model of realized volatility J. Financ. Econom. 7 (2), 174–196.
D'Amato, V., Levantesi, S., Piscopo, G., 2022. Deep learning in predicting cryptocurrency volatility Phys. A 596, 127158.
De Pace, P., Rao, J., 2023. Comovement and instability in cryptocurrency markets Int. Rev. Econ. Finance 83, 173–200.
Diebold, F.X., Mariano, R.S., 2002. Comparing predictive accuracy J. Bus. Econom. Statist. 20 (1), 134–144.
Doering, J., Fairbank, M., Markose, S., 2017. Convolutional neural networks applied to high-frequency market microstructure forecasting In: 2017 9th Computer Science and Electronic Engineering. CEEC, IEEE, pp. 31–36.
Dudek, G., Fiszeder, P., Kobus, P., Orzeszko, W., 2024. Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study Appl. Soft Comput. 151, 111132.
Esparcia, C., Escribano, A., Jareño, F., 2023. Did cryptomarket chaos unleash silvergate's bankruptcy? investigating the high-frequency volatility and connectedness behind the collapse J. Int. Financ. Mark. Inst. Money 89, 101851.
Feng, L., Qi, J., Lucey, B., 2024. Enhancing cryptocurrency market volatility forecasting with daily dynamic tuning strategy Int. Rev. Financ. Anal. 94, 103239.
Fischer, T., Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions European J. Oper. Res. 270 (2), 654–669.
Glosten, L.R., Jagannathan, R., Runkle, D.E., 1993. On the relation between the expected value and the volatility of the nominal excess return on stocks J. Finance 48 (5), 1779–1801.
Gradojevic, N., Kukolj, D., Adcock, R., Djakovic, V., 2023. Forecasting bitcoin with technical analysis: A not-so-random forest? Int. J. Forecast. 39 (1), 1–17.
Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J., 2008. A novel connectionist system for unconstrained handwriting recognition IEEE Trans. Pattern Anal. Mach. Intell. 31 (5), 855–868.
Hansen, P.R., 2005. A test for superior predictive ability J. Bus. Econom. Statist. 23 (4), 365–380.
Hansen, P.R., Lunde, A., Nason, J.M., 2011. The model confidence set Econometrica 79 (2), 453–497.
Hautsch, N., Podolskij, M., 2013. Preaveraging-based estimation of quadratic variation in the presence of noise and jumps: theory, implementation, and empirical evidence J. Bus. Econom. Statist. 31 (2), 165–183.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory Neural Comput. 9 (8), 1735–1780.
Huang, X., Lin, J., Wang, P., 2022. Are institutional investors marching into the crypto market? Econom. Lett. 220, 110856.
Jacod, J., Li, Y., Mykland, P.A., Podolskij, M., Vetter, M., 2009. Microstructure noise in the continuous case: the pre-averaging approach Stochastic Process. Appl. 119 (7), 2249–2276.
Ji, S., Kim, J., Im, H., 2019. A comparative study of bitcoin price prediction using deep learning Mathematics 7 (10), 898.
Katsiampa, P., 2017. Volatility estimation for bitcoin: A comparison of GARCH models Econom. Lett. 158, 3–6.
Katsiampa, P., Corbet, S., Lucey, B., 2019. High frequency volatility co-movements in cryptocurrency markets J. Int. Financ. Mark. Inst. Money 62, 35–52.
Kim, H.Y., Won, C.H., 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models Expert Syst. Appl. 103, 25–37.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization arXiv preprint arXiv:1412.6980.
Köchling, G., Schmidtke, P., Posch, P.N., 2020. Volatility forecasting accuracy for bitcoin Econom. Lett. 191, 108836.
Kristjanpoller, W., Minutolo, M.C., 2018. A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis Expert Syst. Appl. 109, 1–11.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks Commun. ACM 60 (6), 84–90.
LeCun, Y., Bengio, Y., et al., 1995. Convolutional networks for images, speech, and time series Handb. Brain Theory Neural Netw. 3361 (10), 1995.
Lee, A.D., Li, M., Zheng, H., 2020. Bitcoin: Speculative asset or innovative technology? J. Int. Financ. Mark. Inst. Money 67, 101209.
Li, Y., Zhang, W., Urquhart, A., Wang, P., 2022. The role of media coverage in the bubble formation: evidence from the bitcoin market J. Int. Financ. Mark. Inst. Money 80, 101629.
Liu, M., Li, G., Li, J., Zhu, X., Yao, Y., 2021. Forecasting the price of bitcoin using deep learning Finance Res. Lett. 40, 101755.
Liu, L.Y., Patton, A.J., Sheppard, K., 2015. Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes J. Econometrics 187 (1), 293–311.
McNally, S., Roche, J., Caton, S., 2018. Predicting the price of bitcoin using machine learning In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing. PDP, IEEE, pp. 339–343.
Nadarajah, S., Chu, J., 2017. On the inefficiency of bitcoin Econom. Lett. 150, 6–9.
Naeem, M.A., Iqbal, N., Lucey, B.M., Karim, S., 2022. Good versus bad information transmission in the cryptocurrency market: Evidence from high-frequency data J. Int. Financ. Mark. Inst. Money 81, 101695.
Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks In: International Conference on Machine Learning. PMLR, pp. 1310–1318.
Peng, Y., Albuquerque, P.H.M., de Sá, J.M.C., Padula, A.J.A., Montenegro, M.R., 2018. The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression Expert Syst. Appl. 97, 177–192.
Phillip, A., Chan, J.S., Peiris, S., 2018. A new look at cryptocurrencies Econom. Lett. 163, 6–9.
Rawat, W., Wang, Z., 2017. Deep convolutional neural networks for image classification: A comprehensive review Neural Comput. 29 (9), 2352–2449.
Sak, H., Senior, A.W., Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling In: Interspeech. pp. 338–342.
Seo, M., Kim, G., 2020. Hybrid forecasting models based on the neural networks for the volatility of bitcoin Appl. Sci. 10 (14), 4768.

Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019 Appl. Soft Comput. 90, 106181.
Shen, D., Urquhart, A., Wang, P., 2020. Forecasting the volatility of bitcoin: The importance of jumps and structural breaks Eur. Financial Manag. 26 (5), 1294–1323.
Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning J. Big Data 6 (1), 1–48.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition arXiv preprint arXiv:1409.1556.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting J. Mach. Learn. Res. 15 (1), 1929–1958.
Urquhart, A., 2017a. Price clustering in bitcoin Econ. Lett. 159, 145–148.
Urquhart, A., 2017b. The volatility of bitcoin Available at SSRN 2921082.
Vidal, A., Kristjanpoller, W., 2020. Gold volatility prediction using a CNN-LSTM approach Expert Syst. Appl. 157, 113481.
Wang, Y., Andreeva, G., Martin-Barragan, B., 2023. Machine learning approaches to forecasting cryptocurrency volatility: Considering internal and external determinants Int. Rev. Financ. Anal. 90, 102914.
Wang, Z., Oates, T., et al., 2015. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks In: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence.
Xiong, R., Nichols, E.P., Shen, Y., 2015. Deep learning stock volatility with google domestic trends arXiv preprint arXiv:1512.04916.
Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H., 2015. Understanding neural networks through deep visualization arXiv preprint arXiv:1506.06579.
Zakoian, J.-M., 1994. Threshold heteroskedastic models J. Econom. Dynam. Control 18 (5), 931–955.
Zhang, X., Tan, Y., 2018. Deep stock ranker: A LSTM neural network model for stock selection In: Data Mining and Big Data: Third International Conference, DMBD 2018, Shanghai, China, June 17–22, 2018, Proceedings 3. Springer, pp. 614–623.
Zhou, Y.-L., Han, R.-J., Xu, Q., Jiang, Q.-J., Zhang, W.-K., 2019. Long short-term memory networks for CSI300 volatility prediction with baidu search volume Concurr. Comput.: Pract. Exper. 31 (10), e4721.