

RESEARCH

Open Access



Coin impact on cross-crypto realized volatility and dynamic cryptocurrency volatility connectedness

Burak Korkusuz¹ and Mehmet Sahiner^{2*}

*Correspondence:
msahiner001@dundee.ac.uk

¹ Department of Econometrics,
Faculty of Economics
and Administrative Sciences,
Osmaniye Korkut Ata University,
80000 Osmaniye, Turkey

² School of Business, University
of Dundee, Dundee DD1 4HN,
UK

Abstract

This study evaluates the predictive accuracy of traditional time series (TS) models versus machine learning (ML) methods in forecasting realized volatility across major cryptocurrencies—Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and Ripple (XRP). Employing high-frequency data, we analyze cross-cryptocurrency volatility dynamics through two complementary approaches: volatility forecasting and connectedness analysis. Our findings reveal three key insights: (i) TS models, particularly the heterogeneous autoregressive (HAR) model, exhibit superior predictive performance over their ML counterparts, with the long short-term memory (LSTM) model providing competitive yet inconsistent results due to overfitting and short-term volatility challenges; (ii) including lagged realized volatility of large-cap coins improves predictive accuracy for mid-cap coins, especially XRP, whereas forecasts for large-cap coins remain stable, indicating more resilient volatility patterns; and (iii) volatility connectedness analysis reveals substantial spillover effects, particularly pronounced during market turmoil, with large-cap assets (BTC and ETH) acting as primary volatility transmitters and mid-cap assets (XRP and LTC) serving as volatility receivers. These results contribute to the understanding of volatility forecasting and risk management in cryptocurrency markets, offering implications for investors and policymakers in managing market risk and interdependencies in digital asset portfolios.

Keywords: Volatility forecasting, Realized volatility, Bitcoin, Cross-cryptocurrency impact, Dynamic connectedness, Machine learning, Network analysis, Econometric models

JEL Classification: C22, C53, G12

Introduction

Volatility forecasting in cryptocurrency markets poses unique challenges because of its inherent high-risk nature, nonlinearity, and extreme price fluctuations. Traditional time series (TS) models, such as ARMA and HAR, have been widely used in financial forecasting, with the HAR model being particularly effective at capturing long-memory effects, despite not formally belonging to the long-memory class of models (Corsi 2009). However, these models often face limitations in handling the nonlinearities of cryptocurrency markets. To address these complexities, recent studies have increasingly

focused on machine learning (ML) models, such as long short-term memory (LSTM) networks, which can accommodate complex temporal dependencies (i.e. Peng et al. 2018; Bucci 2020; Zhu et al. 2023; Christensen et al. 2023; Wang et al. 2023a; Zhang et al. 2024; Ding et al. 2025). Moreover, the studies of Fischer and Krauss (2018) and Koo and Kim (2024) further demonstrate that ML models can outperform traditional methods to a certain extent. However, despite their promise, ML models also face ongoing concerns regarding overfitting and their ability to effectively capture short-term volatility dynamics.

The predictive accuracy of volatility forecasting models can also be improved by incorporating cross-market information flows, which is particularly important for further amending forecast losses. This can be done by adding lagged exogenous variables from other markets to the primary model, allowing for evaluation of whether such cross-market data improve volatility forecasts. The concept of utilizing information from one market or asset class to forecast another is well established and has been widely explored in the financial forecasting literature. Studies such as Peng et al. (2018), Sin (2013), Degiannakis and Filis (2017), McMillan (2020), Zhang et al. (2020) and Wang et al. (2023b) have demonstrated the significance of cross-market information flow in various asset classes. However, the same depth of research is not present in the cryptocurrency market, where the external factors driving cryptocurrency volatility remain unclear and are still a topic of ongoing debate (see, limited and contradictory findings, among others, Corbet et al. 2018; Kumar and Ajaz 2019; Shahzad et al. 2019; Corbet et al. 2020; Wu et al. 2021; Liang et al. 2022; Wang et al. 2023c; Setiawan and Bhaduri 2023; Köse et al. 2024; Levantesi et al. 2025). While some studies examine the impact of coins on cross-crypto volatility, this area is still underresearched.¹ A few notable exceptions include Guo et al. (2024), who found strong evidence of cross-cryptocurrency return predictability, particularly highlighting Bitcoin's (BTC) lagged returns as a predictor for other cryptocurrencies.

Adding lagged exogenous variables in univariate volatility forecasting models generally involves examining the relationships between two different variables, often yielding pairwise results. These results, however, only show the interdependence between those two variables, which cannot provide a system-wide understanding of volatility transmission. A more comprehensive method for analyzing information transmission across multiple variables is the volatility spillover index of Diebold and Yilmaz (2009, 2012). This approach, which is based on a multivariate VAR model and forecast error variance decomposition, provides a much clearer measure of how many market shocks are transmitted across different variables, making it a powerful tool for understanding market contagion and risk transmission. In its original form, the method provides a static measure of spillovers.² However, the cryptocurrency market is characterized by its dynamic behaviors rather than static patterns, which are largely influenced by its high volatility. The reason is that, unlike traditional markets, cryptocurrencies undergo

¹ This paper interchangeably uses the terms; cryptocurrencies, coins, and crypto.

² This is because it calculates the spillover index from the variance decomposition of a single VAR model estimated over the full sample period. In this static setup, the spillover index gives a fixed measure of how much of the forecast error variance for each variable is attributable to shocks in other variables, but it does not change over time. It reflects the average spillover effects for the entire sample period.

frequent price fluctuations, where market conditions can change rapidly within a short period of time (see, e.g., Chaim and Laurini 2018; Corbet et al. 2019; Shahzad et al. 2019; Bouri et al. 2020; Bouri et al. 2021).

Building on the recent work of Wang et al. (2023b) and Zhang et al. (2024), which focused solely on stock market volatility forecasting, this study presents a novel perspective by incorporating volatility forecasting with volatility connectedness analysis in the cryptocurrency market. In doing so, we evaluate and compare the predictive performance of traditional time series (TS) models, such as HAR and ARMA, with that of advanced machine learning (ML) models, including LSTM, MLP, and GFN, for forecasting realized volatility in major cryptocurrencies—Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and Ripple (XRP). Additionally, we examine the impact of each coin on cross-cryptocurrency realized volatility by incorporating each coin as a predictor to estimate the realized volatility of others. Beyond model comparison, we also analyze the interconnectedness of realized volatility among these cryptocurrencies via a dynamic VAR model, identifying whether each coin acts as a volatility transmitter or receiver. This interconnectedness is further explored through the lens of complex network theory, providing a comprehensive view of realized volatility spillovers within the cryptocurrency market.

This study makes several important contributions to the emerging literature on cryptocurrency volatility by employing high-frequency cryptocurrency data derived from 5-min intervals and combining two distinct methodologies: volatility forecasting and volatility connectedness. First, using realized volatility measures, we demonstrate that traditional time series (TS) models, particularly the HAR model, consistently outperform ML models in forecasting cryptocurrency realized volatility. Although the well-known LSTM model from the ML family performs well in some cases, it struggles with overfitting and short-term volatility. Second, our findings reveal that lagged realized volatility of large-cap coins improves forecasts for mid-cap coins, whereas large-cap coin volatility remains largely unaffected by external crypto data, suggesting greater stability compared with mid-cap coins. Third, the connectedness analysis highlights a rise in spillover effects, especially during periods of market turbulence. In these dynamics, large-cap cryptocurrencies emerge as primary transmitters of volatility, whereas mid-cap cryptocurrencies tend to be receivers, highlighting the hierarchical nature of volatility transmission. Overall, this study aims to provide deeper insights into the strengths and limitations of various forecasting models and analyze volatility connectedness from the perspective of high-frequency data. By doing so, this work provides critical information for market participants, financial researchers, and risk managers to make informed decisions in highly volatile cryptocurrency markets.

The structure of this paper is as follows: Sect. "Methodology" outlines the methodology, Sect. "Data description and analysis of cryptocurrency data characteristics" describes the data and examines cryptocurrency data characteristics, Sect. "Empirical results" presents empirical results, and Sect. "Conclusion" concludes the study.

Methodology

Realized volatility

Volatility is not directly observable; thus, a proxy for true volatility is necessary. The use of daily squared returns as a proxy was standard practice until Andersen and Bollerslev (1998) demonstrated that daily squared returns are a poor substitute compared with cumulative squared intraday returns. Andersen and Bollerslev (1998) define realized volatility (RV_t) as follows:

$$RV_t = \sum_{n=1}^N r_{t,n}^2 \quad (1)$$

where $r_{t,n}^2$ refers to the log asset squared return on day t , n^{th} intraday interval and where $r_{t,n} = p_{t,n} - p_{t,n-1}$, ($t = 1, \dots, T$; $n = 1, \dots, N$), where T is the total number of trading days, N is the number of intraday intervals and where $p_{t,n}$ is the log asset price.

Traditional time series models

HAR model

The HAR-RV model originates from the heterogeneous market hypothesis developed by Müller et al. (1997). This hypothesis posits that different groups of investors exhibit varying risk preferences and respond differently to the same new market information. Inspired by the HARCH model and the heterogeneous market hypothesis, Corsi (2009) proposes the heterogeneous autoregressive (HAR) model for the RV.

$$RV_{t+h}^d = \beta_0 + \beta_d RV_t^d + \beta_w RV_t^w + \beta_m RV_t^m + \varepsilon_{t+h} \quad (2)$$

where RV_t^d is the daily realized volatility; RV_t^w refers to the weekly realized volatility; and RV_t^m indicates the monthly realized volatility. Notably, this study follows the original HAR formulation introduced by Corsi (2009), which models RV in levels rather than in logarithmic form to stabilize variance and ensure positivity. RV_t^w and RV_t^m can easily be calculated as follows:

$$RV_t^w = \frac{1}{5} (RV_{t-5}^d + RV_{t-4}^d + \dots + RV_{t-1}^d) \quad (3)$$

$$RV_t^m = \frac{1}{22} (RV_{t-22}^d + RV_{t-21}^d + \dots + RV_{t-1}^d) \quad (4)$$

This model operates as an additive cascade of various volatility components and aims to capture different time horizons of volatility, specifically daily, weekly, and monthly components. The model's specification integrates these different volatility components to provide a more comprehensive volatility forecasting framework. In Eq. 2, exogenous variables can be added to the equation with the expression $\beta_X X_t^d$. Moreover, the kitchen-sink method, which adds multiple exogenous variables simultaneously into the model, is represented by $\sum_{i=1}^K \beta_i X_{i,t}^d$.

AR, ARMA, and ARFIMA models

The ARFIMA (autoregressive fractionally integrated moving average) model, introduced by Granger and Joyeux in 1980, belongs to the family of long-memory models, meaning that it is effective at capturing the persistence characteristic of volatility over time. The ARFIMA model helps address long-range dependencies in time series data, offering better forecasts by accounting for both short-term fluctuations and long-term persistence. Moreover, Andersen et al. (2003) propose the use of the ARFIMA model specifically for modeling realized volatility. The general form of the ARFIMA (p, d, q) model is described as follows:

$$\varphi(L)(1-L)d(RV_t - \mu) = \theta(L)\varepsilon_t \quad (5)$$

In Eq. (5), $\varphi(L)$ and $\theta(L)$ represent the lag polynomials for the autoregressive (AR) and moving average (MA) components, respectively. The term ε_t refers to the error term, which is assumed to follow a Gaussian white noise distribution with a mean of zero and variance $[N(0, \sigma_\varepsilon^2)]$. The parameter d in the equation represents the fractional differencing, which captures the long-term memory behavior of volatility. While the AR and MA components account for the short memory properties, the fractional differencing term d accounts for modeling the long-term memory in the volatility dynamics. In the ARFIMA specification, the fractional differencing parameter is set at $d \approx 0.40$, which is consistent with Andersen et al. (2003), where values in the range (0, 0.5) capture long memory and $d=0$ corresponds to an ARMA process.

From this point of view, $AR(p)$ is a special case of ARFIMA in which $d=0$ and $q=0$. This means that when the moving average (MA) component and fractional differencing parameter (d) are removed from ARFIMA, the AR model can be obtained. Similarly, ARMA can be derived by removing only the “ d ” parameter from the ARFIMA model equation.

In this study, we use realized volatility data constructed from high-frequency intraday prices but aggregated at the daily level. Given this setup, we deliberately exclude GARCH-type models from our benchmark set for two key reasons. First, traditional GARCH models are designed for modeling volatility directly from return series via maximum likelihood estimation (MLE). Applying these models to realized volatility data introduces estimation challenges and often leads to convergence errors or model misspecification. Second, while the realized GARCH model (Hansen et al. 2012) is specifically designed to incorporate realized measures, it is typically estimated at the same high-frequency level as the input data (e.g., minute-level returns). This makes it incompatible with our framework, which compares models using daily level realized volatility. A comparison between models estimated at different data frequencies would compromise the consistency and fairness of the analysis. Therefore, we focus on comparing traditional time series models (HAR, ARMA) and machine learning models on a unified daily frequency basis.

Machine learning models

Multilayer perceptron (MLP)

Multilayer perceptron (MLP) networks are a type of feedforward neural network commonly used for tasks such as pattern recognition, prediction, and function

approximation. In an MLP, neuron connections are governed by weighted parameters that are iteratively adjusted throughout the learning process. Training in MLPs employs the backpropagation algorithm, which operates through two main phases: a forward pass and a backward pass. In the forward phase, data flow through the network with fixed weights to generate an output. In the backward phase, the algorithm calculates the discrepancy between the actual and predicted outputs, propagating this error back through the network to update the weights. This iterative weight adjustment process is aimed at reducing the mean squared error or total error.

For a single hidden-layer MLP, the model can be mathematically defined as follows:

$$n_{k,t} = w_{k,0} + \sum_{i=1}^i w_{k,i} x_{i,t} \quad (6)$$

$$N_{k,t} = L(n_{k,t}) = \frac{1}{1 + e^{-n_{k,t}}} \quad (7)$$

$$Y_t = \lambda_0 + \sum_{k=1}^k \lambda_k N_{k,t} \quad (8)$$

The training process starts with input vectors $x_{i,t}$, weight vectors $w_{k,i}$, and bias terms $w_{k,0}$. These inputs are then processed via the logistic sigmoid function, represented as $L(n_{k,t}) = \frac{1}{1 + e^{-n_{k,t}}}$, which forms neuron activations $N_{k,t}$ and predicts the output Y_t .

Recurrent neural networks (RNN)

Recurrent neural networks (RNNs) are a specialized form of neural network tailored for sequential data processing. Unlike conventional feedforward networks, RNNs are designed to maintain information from prior inputs through internal memory, enabling them to model both current and past inputs in each computation step. This characteristic makes RNNs particularly effective for tasks with temporal dependencies.

The mathematical formulation for an RNN is as follows:

$$n_{k,t} = w_{k,0} + \sum_{i=1}^i w_{k,i} x_{i,t} + \sum_{k=1}^k \varphi_k n_{k,t-1} \quad (9)$$

$$N_{k,t} = \frac{1}{1 + e^{-n_{i,t}}} \quad (10)$$

$$Y_t = \lambda_0 + \sum_{k=1}^k \lambda_k N_{k,t} \quad (11)$$

where n represents the input features, k is the number of neurons, w denotes weights, and φ captures the influence of the preceding state $n_{k,t-1}$ on the current state.

Long short-term memory (LSTM)

Long short-term memory (LSTM) networks, introduced by Hochreiter and Schmidhuber (1997), address issues in RNNs, such as training instability and the vanishing gradient problem, which can hinder long-sequence learning. The vanishing gradient problem arises when gradients diminish while propagating backward through the network, making it challenging to learn long-term dependencies. LSTM networks mitigate this issue by incorporating a cell state and three gating mechanisms—input, output, and forget gates—that control information flow within the network.

Input Gate: Determines which new information enters the cell state.

Output Gate: Regulates the information used to generate outputs from the cell state.

Forget Gate: Manages which information is removed from the cell state.

This gated structure allows LSTMs to maintain essential information over extended sequences while discarding irrelevant data, making them well suited for complex sequence modeling and time series forecasting tasks.

Radial basis function networks (RBFNs)

Radial basis function networks (RBFNs) are three-layer networks that use radial basis functions as activation units. Originally developed by Lowe and Broomhead (1988), RBFNs aim to improve upon multilayer perceptron (MLP) networks by offering faster training and simpler parameterization.

The process for RBFNs, which is based on McNelis (2005), is defined as follows:

$$n_t = w_0 + \sum_{i=1}^{i^*} w_i x_{i,t} \quad (12)$$

$$R_{k,t} = \phi(n_t; \mu_k) \quad (13)$$

$$\hat{y}_t = \lambda_0 + \sum_{k=1}^{k^*} \lambda_k R_{k,t} \quad (14)$$

In this structure, n_t is a linear combination of input weights w_i and input values $x_{i,t}$, μ_k represents the centers of the radial basis functions, and ϕ their widths. The output $\hat{y}_t$ is estimated via radial basis functions $R_{k,t}$ and adjusted weights λ_k . RBFNs are recognized for their efficiency, particularly in tasks requiring rapid and accurate learning.

Generalized feedforward networks (GFNs)

Generalized feedforward networks (GFNs) extend standard MLP architectures by allowing connections to bypass one or more intermediate layers. This flexible structure enables certain layers to connect directly with the output layer, providing raw information in addition to the processed data from the hidden layers.

The key advantage of GFNs is their adaptability: they provide direct connections for linear data components, whereas hidden layers handle nonlinear processing when needed. Although MLPs theoretically offer similar capabilities, GFNs frequently deliver more efficient and accurate solutions in practical applications.

Modular feedforward networks (MFNs)

Modular feedforward networks (MFNs) adapt the traditional feedforward design to reduce complexity and improve robustness. Prompted by challenges such as slow convergence and difficulty in learning weights, MFNs segment tasks into smaller, independent modules, each contributing autonomously to the final output without interacting during computation. This modular structure enables efficient handling of complex tasks by delegating subtasks to individual network modules.

Optimization process

This study leverages several neural network architectures implemented in MATLAB and Python, employing the following configurations:

- *Hidden layers*: Hidden layers connect the input and output layers and handle essential computations. Although theoretically, a single hidden layer can approximate continuous functions, one or two layers are commonly used in practice for optimal results (Thomas et al. 2017). This study uses up to two hidden layers per model.
- *Epochs*: Each epoch represents a full pass through the training dataset, set to a maximum of 1,000. Early stopping is applied if no improvement occurs after 100 epochs, helping to prevent overfitting.
- *Weights*: Weights are crucial in transforming inputs within hidden layers. This study applies both batch and online modes for weight updates, with batch mode accumulating weight changes across the dataset and online training updating weights after each data presentation.
- *Activation function*: Activation functions restrict the output values to be divided into linear and nonlinear types. Given the nonlinear nature of financial markets (Franses & Van Dijk 2000), this study uses the hyperbolic tangent (tanh) function, which ranges from -1 to 1, providing robust performance for time series forecasting (Farzad et al. 2019).
- *Learning rule*: The Levenberg–Marquardt (LM) algorithm, developed by Levenberg (1944) and Marquardt (1963), is employed to increase network performance by minimizing error functions. Known for rapid and stable convergence, this method is particularly effective for training networks that rely on higher-order adaptive algorithms (Yu & Wilamowski 2018).

For reproducibility and comparability across models, we also conducted grid-based hyperparameter tuning for key architectural components. The specific hyperparameters and value ranges are summarized in Table 1 below:

All the models were trained via multiple random seeds to control for stochastic variation in weight initialization. Forecasting performance was averaged across multiple runs and assessed via a wide array of loss functions (e.g., RMSE, MAE, MSLE, and Theil's U) to provide a robust and holistic comparison. The full results are presented in Sect. "[Empirical results](#)".

Table 1 Hyperparameter tuning grid for neural network models

Model type	Hyperparameter	Values tested
MLP/MFN/GFN	Number of hidden layers	(1, 2)
	Neurons per layer	(32, 64, 128)
	Activation function	(tanh)
RNN/LSTM	Time lag window (sequence)	(5, 10, 15) days
	Number of hidden layers	(1, 2)
	Neurons per layer	(32, 64, 128)
	Dropout rate (LSTM)	(0.1, 0.2, 0.3)
RBFN	Spread/center initialization	Determined via k-means clustering and manual calibration
All models	Learning algorithm	(Levenberg–Marquardt)
	Batch vs. online training	Both

The hyperparameter ranges were chosen to ensure a balance between model complexity and generalizability, in line with standard practices in financial time-series forecasting. Configurations such as 1–2 hidden layers, 32–128 neurons, and the tanh activation function are widely used for capturing nonlinear dynamics.

The Levenberg–Marquardt algorithm was selected for its fast and stable convergence. Dropout was applied to LSTM models for regularization, and both batch and online training modes were tested. For further reference on these choices, please see Thomas et al. (2017), Farzad et al. (2019), Buccì (2020), Yu & Wilamowski (2018), Srivastava et al. (2023), and Kristjanpoller (2024).

**Fig. 1** Working principle of the rolling window mechanism

Rolling window forecasting method

The working principle of the rolling window for volatility forecasting is illustrated in Fig. 1. The estimation sample is rolled forward by adding a new observation while dropping the most distant observation. By doing so, the size of the in-sample window used in the estimation process remains at a fixed length. The first four lines of Fig. 1 illustrate how the loop command functions, whereas the last line displays the predefined training and test samples, which are color coded in green for the time horizon between 1 and 8 and orange for the horizon between 9 and 12. Notably, forecasts extending beyond one step are more likely to be inaccurate because of insufficient data for future predictions. Therefore, forecasts beyond one step ahead are marked as "bad predictions" in the figure.

Loss functions

Table 2 lists the loss functions used in this study for evaluating the accuracy of various forecasting models. Loss functions play a critical role in assessing the accuracy of forecasting models by quantifying the error between actual values (y_t) and predicted values (f_t). The mean error (ME) calculates the average of the differences, indicating the direction of bias in predictions (whether the model tends to over- or underpredict). The root mean square error (RMSE), which squares the differences before averaging and taking the square root, emphasizes larger errors and is sensitive to outliers. On the other hand, the mean absolute error (MAE) avoids squaring and focuses on the average magnitude of errors, ensuring a more robust assessment when outliers are present.

For more robust measures, the median error (MDE) and median absolute error (MDAE) focus on the median of the errors and absolute errors, respectively, making them less sensitive to extreme values than their mean-based counterparts. These median-based metrics help reduce the influence of outliers, offering a clearer view of the typical performance of the model.

The mean absolute error relative to the mean (MAM) and median absolute error relative to the median (MDAMD) normalize the errors by dividing by the mean or median of the actual values, making them useful for comparing forecasts across datasets with different scales. Finally, the mean squared logarithmic error (MSLE) is particularly useful when dealing with exponentially growing data, as it computes the squared differences after taking the logarithm of both actual and forecasted values. This makes MSLE more appropriate for cases where relative errors for smaller values are crucial.

The accuracy of forecasts can be measured via Theil's U statistic, where lower values indicate better forecast performance. This statistic ranges from 0, with 0 representing a perfectly accurate forecast. Theil's U is essentially the ratio of the RMSE of the forecast model to the RMSE of a naive model that predicts the next value as equal to the current

Table 2 Loss functions for evaluating the accuracy of forecasting models

Metric	Name	Formula
ME	Mean error	$mean(y_t - f_t)$
RMSE	Root mean squared error	$\sqrt{mean((y_t - f_t)^2)}$
MAE	Mean absolute error	$mean(y_t - f_t)$
MDE	Median error	$median(y_t - f_t)$
MDAE	Median absolute error	$median(y_t - f_t)$
MAM	Mean absolute error relative to the mean of realizations	$mean \frac{ y_t - f_t }{mean(y)}$
MDAMD	Median absolute error relative to the median of realizations	$median \frac{ y_t - f_t }{median(y)}$
MSLE	Mean squared logarithmic error	$mean([\log(y_t + 1) - \log(f_t + 1)]^2)$
U	Theil's U	$\sqrt{mean(\frac{(f_{t+1} - y_{t+1})}{y_t})^2} / [(\frac{(y_{t+1} - y_t)}{y_t})^2]^{-1}$
UM	Bias proportion	$\frac{(\bar{f} - \bar{y})^2}{MSE}$
UR	Regression proportion	$\frac{(s_f - \Upsilon s_y)^2}{MSE}$
UD	Disturbance proportion	$\frac{(1 - \Upsilon^2)s_y^2}{MSE}$

y_t represents actual values, and f_t represents predicted values

$\bar{f}$ and $\bar{y}$ are the sample means of the forecasts and the actual values, respectively. s_f and s_y are the respective standard deviations (using T in the denominator), and Υ is the sample correlation between y and f

value ($y_{t+1} = y_t$). A naive model yields a Theil's U value of 1; thus, values less than 1 reflect an improvement over this baseline, whereas values above 1 suggest poorer performance.

Theil et al. (1966) also proposed a decomposition of the MSE into three nonnegative components, which help assess forecast quality from different angles, such as UM, UR, and UD. These components provide insight into different aspects of the errors.

Hansen et al. (2011) proposed the widely recognized model confidence set (MCS) procedure for identifying a group of superior forecasting models through a sequential elimination algorithm. At a chosen confidence level, this algorithm iteratively removes models with inferior predictive performance on the basis of a specified loss function without relying on a predefined benchmark. Models that fail to demonstrate adequate predictive accuracy are excluded from the initial pool of candidates. Among the various test statistics available for the MCS procedure, Hansen et al. (2003) recommend the range and semiquadratic statistics as the most effective. The core steps of the MCS procedure are summarized as follows. Let $L_{i,k}$ denote the criterion of model i and let $d_{i,j,k} = L_{i,k} - L_{j,k}$ denote the differential. The null hypothesis of the procedure is $H_{0,M} = E(d_{i,j,k}) = 0, \text{ for } i, j \in M, M \subset M^0$, and the null hypothesis is tested against the alternative hypothesis $H_{1,M} = E(d_{i,j,k}) \neq 0, \text{ for some } i, j \in M$.

VAR model

The vector autoregression (VAR) model, developed by Sims in 1980, is an influential econometric tool designed to capture the linear relationships among multiple time series variables. This model is particularly valuable for studying how several endogenous variables interact with and influence each other over time. To estimate the parameters effectively, the VAR approach is executed in two stages, taking into account heteroscedasticity and ensuring a balance between complexity and parsimony. The general form of a VAR model with p -th order and N -variables can be specified as follows for a covariance-stationary system:

$$R_t = \mu_0 + \sum_{p=1}^p \varnothing_p R_{t-p} + \varepsilon_t \quad (15)$$

where R_t represents a vector of N variables, each corresponding to the volatility of returns from stock markets at a given time t . The term $\varnothing_p$ is an $N \times N$ matrix of coefficients, which captures the relationships between the variables across time. Finally, ε_t denotes a vector of disturbances, with each element being independently and identically distributed (i.i.d.) and having a covariance matrix Σ .

The vector autoregression (VAR) methodology is essential because of its moving average (MA) representation, which plays a key role in comprehending impulse response functions. The MA representation expresses the VAR model as follows:

$$R_t = \mu_0 + \sum_{i=0}^{\infty} K_i \varepsilon_{t-i} \quad (16)$$

where K_i is the $N \times N$ coefficient matrix, defined recursively as:

Table 3 Summary statistics from observations 2018-05-15 to 2023-01-24

Statistics/variable	RV_BTC	RV_ETH	RV_LTC	RV_XRP
Mean	2.818	4.588	5.766	6.597
Median	1.377	2.437	3.205	2.535
Minimum	0.002	0.001	0.060	0.006
Maximum	189.56	157.22	292.55	286.50
Standard deviation	6.917	8.192	11.434	16.133
5% percentage	0.147	0.434	0.644	0.461
95% percentage	9.297	15.196	17.732	23.771
Missing observation	0.000	0.000	0.000	0.000

$$K_i = \emptyset_1 K_{i-1} + \emptyset_2 K_{i-2} + \dots + \emptyset_p K_{i-p} \quad (17)$$

with K_0 being the identity matrix $N \times N$ and $K_i = 0$ for $i < 0$.

This recursive formulation enables the exploration of the system's dynamic behavior by observing how disturbances in one variable propagate to others over time via impulse response functions. This feature makes VAR particularly effective for uncovering causal links and dependencies in multivariate time series data.

In summary, the VAR approach provides a robust framework for modeling the interactions of multiple time series, providing in-depth insights into the system's dynamic structure and the causal relationships among variables. This is especially valuable for studying the volatility of financial markets and other intricate systems where variables are interrelated.

Data description and analysis of cryptocurrency data characteristics

This study uses high-frequency cryptocurrency data (of Bitcoin, Ethereum, Litecoin, and Ripple) downloaded from the “CryptoDataDownload” website, covering a nearly five-year period from May 15, 2018, to January 24, 2023. The intraday dataset, containing almost half a million (493,968) observations, is originally available at 5-min intervals. Hol and Koopman (2002) recommend using a frequency interval between 5 and 30 min for precisely measuring realized volatility. Liu et al. (2015) further support this by comparing over 400 realized measures. They find that it is challenging to significantly outperform the five-minute realized variance, which has led to a broad agreement among scholars and practitioners to adopt the 5-min realized variance as a standard for target volatility.

Having obtained this large volume (nearly half a million) of high-frequency data, the intraday dataset at the five-minute interval is aggregated into daily realized variance values, yielding 1716 daily observations. For forecasting, approximately the first 1000 ($\approx 60\%$) daily data points are used for in-sample estimation, whereas the remaining 716 ($\approx 40\%$) are allocated as the out-of-sample forecasting period.³ A one-step-ahead rolling window forecasting technique is employed to generate volatility forecasts.

³ We also test the model performance using an 80%-20% training-test as well as 60%-40% sample split.

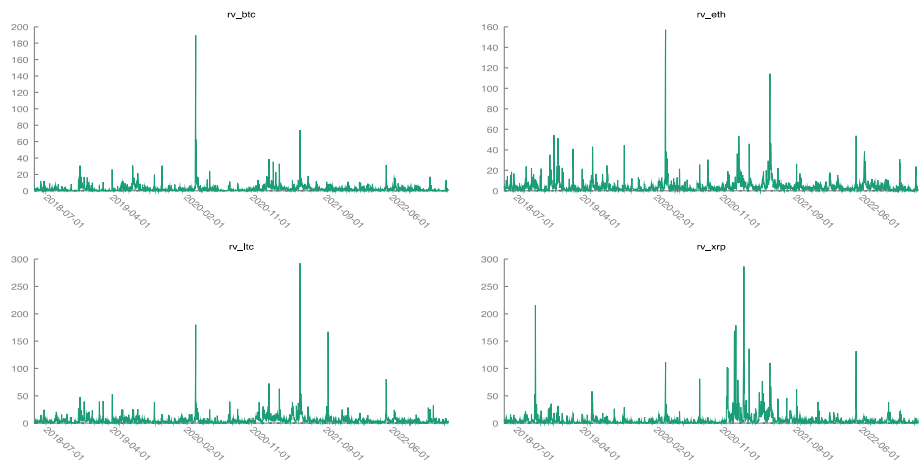


Fig. 2 Line graphs of realized volatility for cryptocurrency data

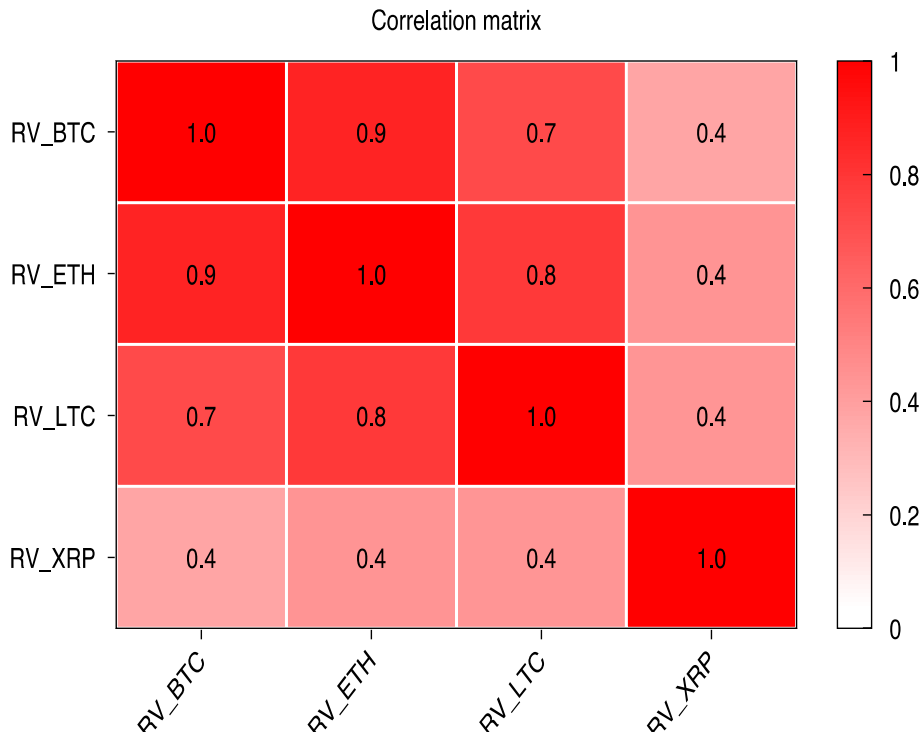


Fig. 3 Correlation matrix of realized volatility for the selected coins

In Table 3, the summary statistics for the realized variance of BTC, ETH, LTC, and XRP reveal notable patterns. BTC has the lowest mean (2.818) and median (1.377), indicating less overall variance than the other variables. Conversely, XRP has the highest mean (6.597) and maximum (286.50) values, reflecting greater realized variance and volatility. The volatility of these realized variances is further emphasized by the standard deviations, with XRP exhibiting the highest deviation and BTC the lowest deviation. Overall, the cryptocurrencies exhibit significant variability, with XRP being the most volatile.

The 5% and 95% percentage statistics reflect the lower and upper bounds of realized variance distributions, providing insight into the tails of the variance series for the four cryptocurrencies. BTC has the lowest 5% percentile (0.147) and 95% percentile (9.297), indicating lower extreme volatility in comparison to the other cryptocurrencies. The XRP has the highest 95% percentile (23.771), suggesting greater extreme volatility on the upper end, followed by LTC and ETH. These percentiles illustrate the wide range and tail behavior of volatility in cryptocurrency markets, with XRP and LTC showing the highest tail risk (Fig. 2).

Figure 3 shows the correlation matrix that demonstrates the relationships between the four cryptocurrencies, such as BTC, ETH, LTC, and XRP. BTC and ETH exhibit the strongest correlation (0.9), indicating a high degree of comovement between their realized variances. Similarly, ETH and LTC also show a strong correlation (0.8), suggesting that their volatility patterns tend to move in tandem. BTC and LTC follow, with a moderate correlation of 0.7. The XRP, however, is less correlated with the other three variables, with values of approximately 0.4, which implies that its volatility is more independent than those of the BTC, ETH, and LTC. Correlation tests for the coefficients in Fig. 3 show that all reported correlations are statistically significant at the 5% level or better, with the highest correlations (≥ 0.70) being significant at the 0.1% level. Notably, all diagonal elements are 1.00 by definition and are not tested.

Figure 4 shows the graphs of the Hurst exponent for the realized variance (RV) series of BTC, ETH, LTC, and XRP. All of the values are above 0.5, which indicates persistent long-memory behavior for the four selected coins. Specifically, BTC and LTC have an estimated Hurst exponent of 0.71, ETH is 0.69, and XRP is 0.78. These values imply that the realized volatility of these cryptocurrencies is highly persistent, meaning that

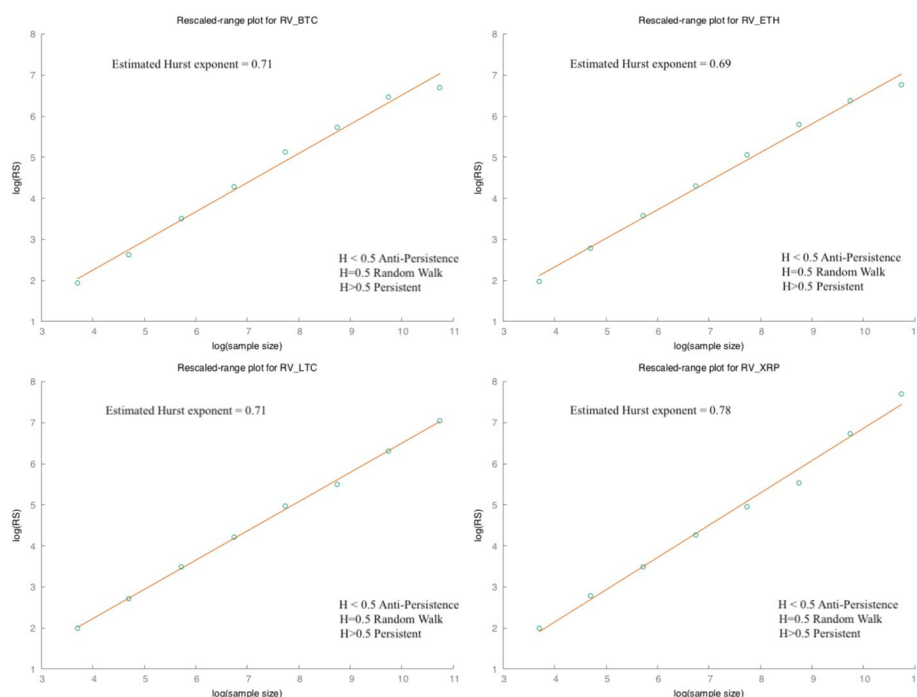


Fig. 4 Graphs of the estimated Hurst exponent for the realized volatility of the selected coins

past movements influence future volatility. This persistence is crucial for modeling, as it suggests that volatility patterns do not follow a random walk but rather exhibit trends that can be exploited for forecasting and risk management. This enhances the understanding of volatility dynamics and supports the use of models that account for long-memory characteristics in cryptocurrency markets, such as ML models and some traditional TS models.

Empirical results

This study incorporates two distinct methods, volatility forecasting and volatility connectedness, which employ high-frequency cryptocurrency data. The first part of the empirical results thoroughly assesses the predictive performance of traditional TS models against ML models for realized volatility forecasting across four major cryptocurrencies.⁴ In the second part, an extensive connectedness analysis for cryptocurrency realized volatility is conducted utilizing a dynamic VAR model to understand volatility spillovers within the cryptocurrency market. Our analysis is conducted at the coin-specific level; therefore, the findings cannot be directly generalized across all cryptocurrencies. Therefore, this scope may not capture the volatility dynamics of smaller coins, which could behave differently. Moreover, relying solely on realized volatility excludes option-implied measures that reflect forward-looking expectations.

Model comparison for realized volatility forecasting

Tables 4, 5, 6, and 7 for Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and Ripple (XRP) realized volatility forecasting, respectively, provide a comprehensive comparison of various machine learning (ML) and traditional time series (TS) models. Relying on a single loss function, such as the RMSE and MAE, can lead to misleading conclusions. Therefore, the models used in this work are evaluated on the basis of a wide range of different error metrics, such as the mean error (ME), root mean squared error (RMSE), mean absolute error (MAE), root mean square error (MDE), median absolute error (MDAE), mean absolute error relative to the mean of realizations (MAM), median absolute error relative to the median of realizations (MDAMD), mean squared logarithmic error (MSLE), Theil's U (U), bias proportion (UM), regression proportion (UR), and disturbance proportion (UD).

Notably, the average of all the metrics is given at the bottom row of each table. By averaging across all loss functions, the goal is to avoid possible misleading conclusions. For example, LSTM for the BTC yields competitive results for the MDAE, MDAMD and UD loss functions but exhibits relatively poor performance in terms of several other metrics, such as the RMSE, ME and MDE. However, the averaging method could provide a robust, consistent and comprehensive evaluation of the forecasting models. This holistic approach approximates the true predictive power of the models and provides a more balanced evaluation.

⁴ To address potential concerns regarding sample selection bias and enhance the robustness of our findings, we compare model performance under a 60%–40% training–test split with an alternative 80%–20% split as a subperiod robustness check. The results remain consistent, with only minor variations. Notably, the 80%–20% split, which excludes much of the COVID-19 period from the test sample, yields slightly lower loss function values, which likely reflecting the reduced impact of pandemic-induced volatility. Full results are available upon request.

Table 4 Forecast results of the TS and ML models for the realized volatility of BTCs

Bitcoin	GFN	LSTM	MFN	MLP	RBFN	RNN	ML_FC	AR	ARFIMA	ARMA	TS_FC	HAR
ME	-3.371	0.999	1.325	-8.065	-0.104	0.992	-1.371	-0.154	-0.159	-0.129	-0.139	-0.115
RMSE	5.841	4.570	4.775	14.245	5.537	5.268	5.270	4.008	3.933	3.898	3.926	3.934
MAE	4.532	1.985	2.259	10.236	2.961	2.836	3.292	1.882	1.879	1.808	1.828	1.784
MdE	-3.938	0.094	0.533	-6.575	-0.006	0.295	-1.702	-0.893	-1.017	-0.868	-0.904	-0.805
MdAE	4.182	0.805	1.152	7.239	1.524	1.654	2.177	1.277	1.351	1.188	1.239	1.135
MAM	1.668	0.731	0.831	3.767	1.090	1.044	1.212	0.693	0.691	0.665	0.673	0.656
MdAMd	2.551	0.491	0.703	4.416	0.930	1.009	1.328	0.779	0.824	0.725	0.756	0.692
MSLE	1.171	0.448	0.815	2.514	0.885	0.953	0.765	0.330	0.346	0.306	0.317	0.298
U	10.290	2.717	2.430	15.240	4.687	4.404	6.139	2.531	2.960	2.156	2.387	1.928
UM	0.333	0.048	0.077	0.321	0.000	0.035	0.068	0.001	0.002	0.001	0.001	0.001
UR	0.124	0.063	0.109	0.588	0.395	0.298	0.265	0.008	0.016	0.001	0.000	0.004
UD	0.543	0.889	0.814	0.091	0.605	0.667	0.667	0.991	0.982	0.997	0.999	0.995
Average	1.994	1.153	1.318	3.668	1.542	1.621	1.509	0.954	0.984	0.896	0.924	0.876
MCS	0.000	0.102	0.003	0.000	0.000	0.000	0.000	0.412	0.813	0.805	0.539	0.921

The average represents the arithmetic mean of all loss functions

Table 5 Forecast results of the TS and ML models for the realized volatility of the ETH

Ethereum	GfN	LSTM	MFN	MLP	RBFN	RNN	ML_FC	AR	ARFIMA	ARMA	TS_FC	HAR
ME	0.599	1.643	− 1.114	− 2.983	− 1.936	− 3.380	− 1.195	0.040	0.279	0.034	0.094	0.021
RMSE	8.099	8.193	8.084	9.883	8.438	9.689	8.045	6.130	6.833	6.008	6.125	6.004
MAE	4.149	3.862	4.620	6.548	5.181	6.568	4.624	2.772	3.165	2.705	2.741	2.647
MdE	− 0.357	0.352	− 2.515	− 3.475	− 2.933	− 4.527	− 2.464	1.180	− 1.504	− 1.095	− 1.250	− 1.031
MdAE	2.144	1.712	3.312	4.872	3.684	5.165	3.304	1.742	2.170	1.659	1.770	1.593
MAM	0.895	0.833	0.996	1.412	1.117	1.416	0.997	0.598	0.683	0.583	0.591	0.571
MdAMd	0.780	0.623	1.205	1.772	1.340	1.879	1.202	0.633	0.789	0.603	0.644	0.579
MSLE	0.841	0.794	0.829	1.237	0.927	1.247	0.814	0.356	0.485	0.338	0.361	0.321
U	1.415	2.461	3.747	8.094	4.677	6.854	4.356	1.430	2.292	1.312	1.515	1.146
UM	0.005	0.040	0.019	0.091	0.053	0.122	0.022	0.000	0.002	0.000	0.000	0.000
UR	0.188	0.174	0.176	0.368	0.205	0.320	0.162	0.001	0.077	0.002	0.020	0.001
UD	0.807	0.786	0.805	0.541	0.743	0.558	0.816	0.999	0.921	0.998	0.980	0.999
Average	1.630	1.789	1.680	2.363	1.791	2.159	1.724	1.127	1.349	1.096	1.132	1.071
MCS	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.743	0.000	0.845	0.722	0.867

The average represents the arithmetic mean of all loss functions

Table 6 Forecast results of the TS and ML models for realized volatility of the LTC

Litecoin	GFN	LSTM	MFN	MLP	RBFN	RNN	ML_FC	AR	ARFIMA	ARMA	TS_FC	HAR
ME	1.660	0.471	0.851	5.398	-0.466	1.083	1.499	0.330	0.684	0.186	0.342	0.170
RMSE	15.920	14.700	14.942	28.832	15.270	18.946	15.772	14.013	14.349	13.642	13.747	13.723
MAE	6.578	5.190	5.641	15.960	6.237	8.640	6.464	4.398	4.984	4.055	4.219	4.109
MdE	-0.098	-1.842	-1.666	1.921	-1.983	-1.659	-0.817	-1.899	-2.220	-1.691	-1.969	-1.724
MdAE	3.261	2.890	3.110	7.612	3.721	4.692	3.341	2.828	3.252	2.343	2.715	2.367
MAM	1.033	0.815	0.886	2.507	0.980	1.357	1.015	0.691	0.783	0.637	0.663	0.645
MdAMd	0.928	0.822	0.885	2.166	1.059	1.335	0.950	0.805	0.925	0.667	0.772	0.673
MSLE	1.284	0.656	0.804	1.986	0.978	1.105	1.025	0.499	0.674	0.417	0.471	0.425
U	1.387	1.194	1.220	3.660	1.337	1.860	1.322	1.088	1.276	0.981	1.060	0.978
UM	0.011	0.001	0.003	0.035	0.001	0.003	0.009	0.001	0.002	0.001	0.001	0.001
UR	0.185	0.056	0.084	0.720	0.125	0.434	0.172	0.004	0.011	0.001	0.006	0.001
UD	0.804	0.943	0.913	0.245	0.874	0.563	0.819	0.996	0.987	0.999	0.993	0.999
Average	2.746	2.158	2.306	5.920	2.344	3.197	2.631	1.979	2.142	1.853	1.918	1.864
MCS	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.598	0.001	0.576

The average represents the arithmetic mean of all loss functions

Table 7 Forecast results of the TS and ML models for the realized volatility of the XRP

XRP	GFN	LSTM	MFN	MLP	RBFN	RNN	ML_FC	AR	ARFIMA	ARMA	TS_FC	HAR
ME	-0.417	1.767	0.859	-0.266	4.237	3.812	1.665	0.014	0.481	0.004	0.092	-0.133
RMSE	13.316	12.481	15.325	65.589	13.038	17.171	17.847	11.196	11.922	11.091	11.075	10.897
MAE	6.953	5.775	7.493	36.872	7.049	10.039	9.607	4.927	5.473	4.722	4.799	4.714
MdE	-2.399	-0.834	0.127	3.255	3.092	3.555	1.354	-2.402	-2.785	-2.059	-2.368	-1.845
MdAE	3.826	2.349	3.289	16.812	4.071	5.726	4.555	2.984	3.694	2.632	2.923	2.342
MAM	1.017	0.845	1.096	5.393	1.031	1.468	1.405	0.721	0.800	0.691	0.702	0.689
MdAMd	1.174	0.721	1.009	5.157	1.249	1.756	1.397	0.915	1.133	0.807	0.897	0.718
MSLE	0.966	0.751	1.230	3.717	2.892	2.131	1.979	0.582	0.728	0.520	0.561	0.492
U	1.788	1.296	1.745	9.946	1.469	2.134	2.295	1.176	1.554	1.062	1.161	0.984
UM	0.001	0.020	0.003	0.000	0.106	0.049	0.009	0.001	0.002	0.001	0.001	0.001
UR	0.130	0.020	0.337	0.963	0.040	0.423	0.501	0.001	0.127	0.001	0.013	0.001
UD	0.869	0.960	0.660	0.037	0.855	0.528	0.490	1.000	0.871	1.000	0.987	1.000
Average	2.622	2.517	3.218	14.647	3.823	4.784	4.211	2.011	2.300	1.947	1.984	1.886
MCS	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.378	0.254	0.502

The average represents the arithmetic mean of all loss functions

Table 4 shows the performance of different forecasting models for Bitcoin in terms of the ML and TS approaches. Looking at the average of various loss functions (i.e., bottom row), it becomes evident that the time series models (TS), particularly HAR and ARMA, tend to perform better overall than the ML models do. The average loss values at the bottom indicate that the HAR model (average=0.876) performs the best among all the models, followed closely by the ARMA model (average=0.896). Generally, these traditional TS models outperform the ML models, with the best ML model being LSTM, which has an average loss of 1.153.

The forecast averaging methods, both for the ML models (RNN_FC) and TS models (TS_FC), also demonstrate strong performance, although the TS forecast combination (TS_FC) shows a more consistent outcome (average=0.924) than the ML forecast combination (RNN_FC, average=1.509). This finding indicates that while ML models may achieve strong forecasting power in specific cases (e.g., LSTM and MFN), traditional TS models, especially HAR and ARMA, exhibit more robust, reliable and consistent forecasting performance for Bitcoin.

On the basis of Table 5, several noteworthy points can be inferred regarding the performance of ML models versus traditional TS models for forecasting Ethereum volatility on the basis of various loss functions. Considering the average of the loss functions, traditional TS models again tend to perform better than ML models do. Specifically, the HAR and ARMA models have the lowest values of the average row (1.071 and 1.096, respectively), followed by simple AR (1.127). These results indicate that, similar to the former Bitcoin analysis, traditional TS models also seem to be more robust for Ethereum realized volatility forecasting. Unlike the former Bitcoin analysis, the GFN model of the ML group has a competitive average of 1.630, making it the best-performing ML model, followed by the MFN and LSTM models (1.680 and 1.789, respectively). However, even if both ML models for Ethereum achieve the lowest forecast errors in the ML group of models, they still fall short compared with the worst-performing TS models, such as ARFIMA (1.349) and AR (1.127).

Like Bitcoin and Ethereum, in the context of Litecoin (LTC) realized volatility forecasting, traditional TS models consistently outperform the ML models in Table 6. The ARMA model emerges as the top performer, showing the lowest error rates, with an average of 1.853, indicating high forecasting accuracy. The HAR and AR models also perform well, achieving relatively low average error values. Conversely, some ML models, such as LSTM and MFN, exhibit promising performance for LTC, with the lowest average performance values of 2.158 and 2.306, which are almost comparable to those of the worst-performing TS models, such as the ARFIMA model. However, the MLP model shows the highest average, indicating that for Litecoin, the gap between the best and worst models' average performance ratings is more explicit than it is for Bitcoin and Ethereum.

Similar to the findings for the other three cryptocurrencies, Table 7 shows that traditional TS models, especially HAR and ARMA, once again demonstrate their reliability and consistency in forecasting XRP realized volatility. Among the ML models, the LSTM performs well, whereas the MLP seems to be the worst-performing model because of its high average criterion value. Interestingly, when average performance across large-cap and medium-cap coins is compared, the gap between the best and worst

models becomes more pronounced. This suggests that large-cap coins such as BTC and ETH exhibit more predictable volatility patterns than medium-cap coins do (e.g., LTC and XRP).

Following the approach of Hansen et al. (2011), Wang et al. (2016), and Liu et al. (2017), we implement the Model Confidence Set (MCS) procedure at a 75% confidence level, using the average across multiple loss functions (penultimate rows in the tables) to assess the relative predictive performance of competing models. Models with p values above 0.25 are retained in the confidence set, and the best-performing models are highlighted in bold. The results reveal that traditional time series models, particularly HAR and ARMA models, are consistently retained in the MCS and dominate in terms of forecast performance. Among machine learning models, only LSTM and MFN occasionally appear in the confidence set, although with relatively weaker support. The tables reveal that the MCS procedure penalizes machine learning models more stringently because of their generally higher loss function values, leading to the exclusion of many ML models from the superior set. Overall, the MCS test based on average loss confirms the superiority of traditional time series models over most machine learning counterparts in terms of forecasting the realized volatility of the four selected coins.

To enhance the interpretability of the detailed forecasting results presented in Tables 4, 5, 6 and 7, we incorporate Fig. 5, which provides a clearer cross-model and cross-asset comparison. The figure presents a bar chart of the average forecasting error for each model across the four cryptocurrencies, calculated as the arithmetic mean of all evaluation metrics. This confirms that traditional time series models, particularly HAR and ARMA, consistently outperform their machine learning counterparts, especially for large-cap coins such as Bitcoin and Ethereum. In contrast, the performance of the ML models is more variable, with some models (e.g., MLP and RBFN) exhibiting substantial error magnitudes, particularly for mid-cap assets such as Litecoin and Ripple.

The abovementioned results show that traditional TS models, specifically the HAR and ARMA models, have robust predictive capabilities, likely because of their ability to capture the multiscale dynamics inherent in cryptocurrency volatility data. On the other hand, ML models exhibit poor forecasting performance, probably because they struggle with overfitting and capture short-term volatility. However, the HAR model

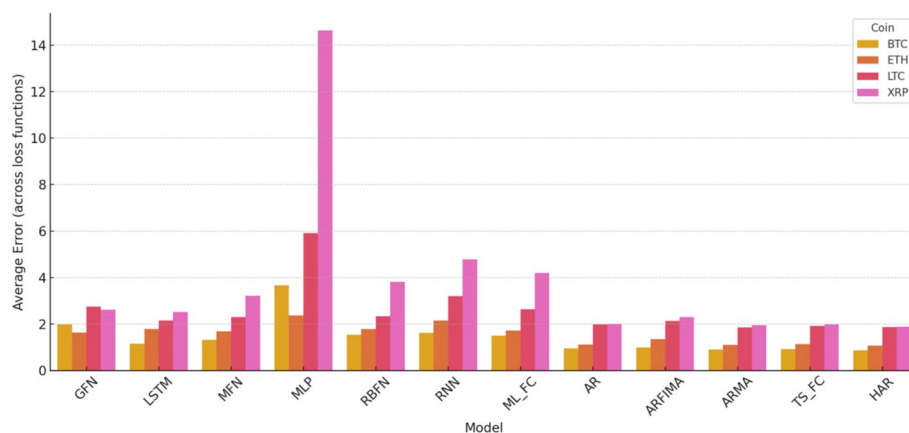


Fig. 5 Average forecast error by model and cryptocurrency

consists of daily, weekly, and monthly components to capture short-, middle- and long-term volatility. Consequently, the HAR model seems to perform the best because of its consistent outperformance in forecasting realized volatility, particularly compared with ML models across different cryptocurrencies such as BTC, ETH, LTC, and XRP. Therefore, this strong performance across multiple coins demonstrates the HAR model's suitability for financial volatility forecasting, making it an ideal choice for enhancing predictive accuracy through the incorporation of additional exogenous variables such as cross-cryptocurrency data.

Coin impact on cross-crypto volatility

While finding the best-suited volatility model for the cryptocurrency market is important, the cryptocurrency market is generally characterized by extreme price volatility, making it one of the most volatile assets. Given this unique feature of the coin market, its cross-asset drivers are not well established and are thus mostly ambiguous.⁵ However, the cross-crypto relationship among cryptocurrency prices or volatilities is a well-known fact. For this reason, instead of examining the cross-asset volatility relationship of the cryptocurrency market because it does not have a solid foundation, this study considers only cross-crypto volatility information to gain further improvements in the forecasts of the best-performing HAR model.

Evaluating the coin impact on cross-crypto volatility, on the basis of the average of all the metrics, reveals that certain cryptocurrencies have a more significant influence on others when used as exogenous variables. Table 8 indicates that BTC's realized volatility forecasting is largely unaffected by other coins, with the exception of ETH, which improves BTC's forecast accuracy by 5.25%. The average of all loss functions suggests that BTC, compared with other coins, maintains more consistent volatility predictions without relying on external crypto data. However, BTC improves the forecasting accuracy of ETH by 1.21% and significantly increases XRP accuracy by 12.26%.

In the right panel of Table 8, ETH exhibits similar characteristics to those of BTC in terms of the impact of cross-crypto volatility. Its forecasting performance remains largely unchanged when other coins' data are included, except for BTC, which provides a modest improvement of 1.21%. However, ETH significantly improves the accuracy of XRP's realized volatility forecasting by 12.20%. Like BTC, ETH shows stability in its own volatility forecasts, suggesting that it does not rely on external crypto information for more accurate forecasting. This finding indicates that the realized volatility of ETH behaves similarly to that of BTC, remaining consistent without the need for additional coin data. The robustness of these results is further supported by the MCS procedure, which confirms the stability of the ranking across different models.

According to Table 9, Litecoin's cross-crypto forecasting performance shows that the impacts of cross-crypto information are mixed, with the HAR-XRP model showing a modest improvement (0.86%), whereas both HAR-BTC and HAR-ETH lead to slight increases in error.

⁵ Although some studies (Shahzad et al. 2019; Corbet et al. 2020; Liang et al. 2022; Wang et al. 2023c) suggest a number of external factors that have a significant impact on forecasting cryptocurrency volatility, what drives the cryptocurrency volatility is a quite new ongoing topic without solid foundation.

Table 8 Forecast results of the HAR model with exogenous cryptocurrencies for BTC and ETH

	HAR Bitcoin (BTC)	HAR-eth	HAR-ks	HAR-ltc	HAR-xrp	HAR Ethereum (ETH)	HAR-btc	HAR-ks	HAR-ltc	HAR-xrp
ME	− 0.115	− 0.234	− 0.172	− 0.065	− 0.119	0.021	− 0.090	− 0.046	0.059	0.021
RMSE	3.934	4.068	4.132	3.934	3.929	6.004	6.028	6.060	6.061	6.000
MAE	1.784	1.829	1.929	1.800	1.783	2.647	2.689	2.717	2.697	2.649
MdE	− 0.805	− 0.663	− 0.631	− 0.806	− 0.782	− 1.031	− 0.988	− 0.894	− 0.990	− 1.034
MdAE	1.135	1.021	1.119	1.174	1.112	1.593	1.522	1.534	1.581	1.591
MdM	0.656	0.674	0.710	0.663	0.657	0.571	0.580	0.586	0.582	0.571
MdAMd	0.692	0.623	0.683	0.716	0.679	0.579	0.554	0.559	0.576	0.579
MSLE	0.298	0.289	0.344	0.309	0.296	0.321	0.320	0.324	0.328	0.321
U	1.928	1.355	1.444	2.119	1.888	1.146	1.087	1.094	1.166	1.137
UM	0.001	0.003	0.002	0.000	0.001	0.000	0.000	0.000	0.000	0.000
UR	0.004	0.075	0.082	0.000	0.005	0.001	0.006	0.004	0.001	0.001
UD	0.995	0.922	0.917	1.000	0.994	0.999	0.994	0.996	0.999	0.999
Average	0.876	0.830	0.880	0.904	0.870	1.071	1.058	1.078	1.088	1.070
%	−	5.25%	− 0.46%	− 3.19%	0.68%	−	1.21%	− 0.65%	− 1.58%	0.09%
MCS	0.000	1.000	0.000	0.000	0.078	0.000	0.875	0.000	0.000	0.057

Bold indicate the best-performing model

The average represents the arithmetic mean of all loss functions

The percentage changes shown in the penultimate row are indicated with percentage symbols

A negative percentage change means that the respective model distorts the forecasting accuracy compared with the benchmark HAR model

Table 9 Forecast results of the HAR model with exogenous cryptocurrencies for the LTC and XRP

	HAR Litecoin (LTC)	HAR-btc	HAR-eth	HAR-ks	HAR-xrp	HAR Ripple (XRP)	HAR-btc	HAR-eth	HAR-ks	HAR-ltc
ME	0.170	0.340	0.306	0.275	0.210	− 0.133	− 0.125	− 0.122	− 0.069	− 0.103
RMSE	13.723	13.695	13.688	13.651	13.640	10.897	10.900	10.895	10.893	10.879
MAE	4.109	4.109	4.219	4.201	4.057	4.714	4.711	4.711	4.724	4.707
MdE	− 1.724	− 1.754	− 1.646	− 1.525	− 1.720	− 1.845	− 1.849	− 1.845	− 1.794	− 1.812
MdAE	2.367	2.359	2.475	2.393	2.314	2.342	2.339	2.342	2.393	2.385
MAM	0.645	0.645	0.662	0.659	0.637	0.689	0.689	0.689	0.691	0.689
MdAMd	0.673	0.671	0.704	0.681	0.658	0.718	0.719	0.720	0.736	0.733
MSLE	0.425	0.432	0.447	0.440	0.414	0.492	0.492	0.494	0.496	0.497
U	0.978	0.978	0.963	0.957	0.971	0.984	0.985	0.989	0.994	1.000
UM	0.001	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000
UR	0.001	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.000	0.000
UD	0.999	0.999	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000
Average	1.864	1.873	1.902	1.894	1.848	1.886	1.655	1.656	1.672	1.665
%	−	− 0.48%	− 2.04%	− 1.60%	0.86%	−	12.26%	12.20%	11.35%	11.71%
MCS	0.845	0.652	0.391	0.569	1.000	0.236	1.000	1.000	0.798	0.854

Bold indicate the best-performing model

The average represents the arithmetic mean of all loss functions

The percentage changes shown in the penultimate row are indicated with percentage symbols

A negative percentage change means that the respective model distorts the forecasting accuracy compared with the benchmark HAR model

Ripple, however, shows the most significant interdependence with other coins. This is to say that XRP's forecasting is slightly more sensitive than other coins are, experiencing the largest percentage improvement from the cross-crypto information. The inclusion of BTC, ETH, and LTC on an individual basis improves the forecasting power of the baseline HAR model of the XRP by 12.26%, 12.20% and 11.71%, respectively. These figures drawn from average loss measures are broadly supported by the MCS analysis. Therefore, this might be evidence that XRP seems to be significantly influenced by the cross-crypto data of large-cap coins, suggesting that Ripple may have a greater degree of interconnectedness with larger cryptocurrencies in its realized volatility movements. This interconnectedness could imply the potential for more predictive power when large-cap cross-crypto data are utilized, especially for smaller coins such as XRP.

The HAR-KS model not only includes the standard model parameters but also incorporates all cross-crypto coins as exogenous variables simultaneously. In this regard, compared with the baseline HAR model, the HAR-KS model slightly reduces the performance for BTC and ETH by 0.46% and 0.65%, respectively. This suggests that including all cross-crypto information does not enhance the volatility forecasts of large-cap coins. When non-large-cap coins are assessed, the HAR-KS model causes a 2.04% decline in forecasting accuracy for LTCs compared with the HAR model. This suggests that for LTC, incorporating cross-crypto volatilities slightly worsens the model's performance. However, XRP shows a significant improvement, with the forecasting performance increasing by 11.36%. This infers that while cross-crypto volatilities do not substantially improve (and may even slightly worsen) forecasting performance for large-cap coins such as BTC and ETH, they can enhance volatility forecasts for smaller-cap coins such as XRP.

Dynamic coin connectedness

Volatility forecasting exercises can improve the understanding of how the volatility of one cryptocurrency may affect another, which indirectly measures spillovers. However, this method typically focuses on predicting individual volatility rather than broader, interconnected dynamics. For a more direct and system-wide understanding of volatility spillovers, volatility connectedness analysis is generally more suitable, as it evaluates how volatility propagates across multiple assets within a network, revealing a holistic view of these relationships across the entire market. Therefore, this section conducts a

Table 10 Dynamic coin-connectedness table

	RV_BTC	RV_ETH	RV_LTC	RV_XRP	FROM
RV_BTC	39.05	26.33	21.4	13.23	60.95
RV_ETH	24.91	36.78	22.85	15.46	63.22
RV_LTC	23.06	24.64	37.87	14.42	62.13
RV_XRP	16.42	18.24	15.8	49.54	50.46
TO	64.39	69.21	60.05	43.11	236.76
INC.OWN	103.44	105.99	97.92	92.65	cTCI/TCI
NET	3.44	5.99	− 2.08	− 7.35	78.92/59.19
NPT	2	3	1	0	

NET TO – FROM, NPT net pairwise transmission, (c) TCI (Inc. own volatility) total connectedness index

The results are produced on the basis of a 200-day rolling-window VAR model with a 10-step-ahead forecast horizon

volatility connectedness analysis to identify which coins influence others and which are influenced from a systemic and comprehensive perspective.

The dynamic coin-connectedness table (Table 10) presents the results of a dynamic VAR model estimated via the rolling-window equation-by-equation technique, which is applied to the realized variance series of the BTC, ETH, LTC, and XRP. The "FROM" column represents how much each coin's realized variance is influenced by others, whereas the "TO" row shows how much each coin influences other cryptocurrencies.

In Table 10, the total connectedness index (cTCI) including own influence is 78.92%, indicating a high level of interconnectedness among cryptocurrency assets, whereas the TCI (connectedness excluding own influence) is slightly lower at 59.19%. Figure 6 illustrates the total connectedness index (TCI) of BTCs, ETHs, LTCs, and XRP from 2019 to 2023. The TCI reflects the extent of interdependence among these cryptocurrencies' realized volatilities over time. As shown, the TCI fluctuates between 40 and 80, with several notable peaks in early 2020, mid-2021, and late 2022. These spikes indicate periods of greater cross-crypto connectedness, likely corresponding to global market events or volatility shocks that impact all coins simultaneously. The general trend suggests that cross-crypto connectedness is time-varying but tends to remain relatively high, implying a significant degree of volatility spillover across the major cryptocurrencies throughout this period.

In terms of the numbers in Table 10, BTC has a significant impact on others, with a "TO" total percentage value of 64.39, whereas ETH has the highest overall impact, with a total percentage value of 69.21. LTC and XRP coins exhibit slightly lower influence, particularly XRP, with a "TO" of only 43.11. In terms of their "FROM" values, XRP is relatively less influenced by external coins, with 50.46% of its variance explained by external factors. On the other hand, BTC, ETH and LTC coins are relatively more dependent on one another, with total percentages of "FROM" of 60.95, 63.22 and 62.13 and "TO" of 64.39, 69.21, and 60.05, respectively. Even if these "FROM" and "TO" figures provide some related information regarding the spillover magnitude of coins, they cannot determine whether a coin is a net volatility transmitter or net volatility receiver. For this reason, NET statistics for coins must be analyzed for a more reliable and balanced assessment.

The net total directional connectedness (NET) is determined by the difference between the "TO" and "FROM". This metric reveals the overall influence of a series

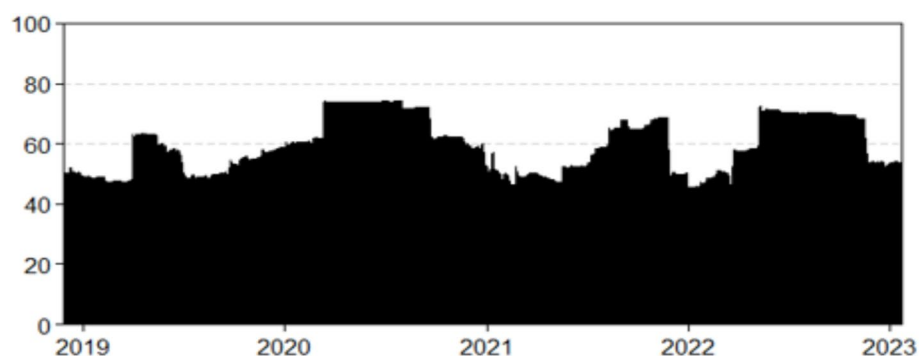


Fig. 6 Total connectedness index of the selected coins

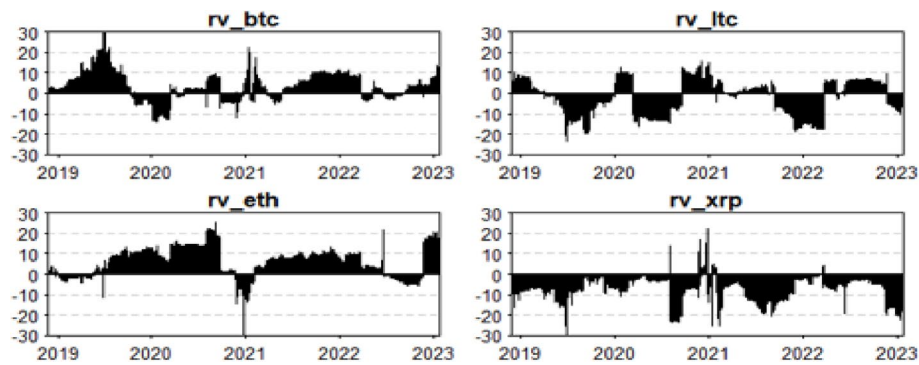


Fig. 7 Net total directional connectedness for each coin

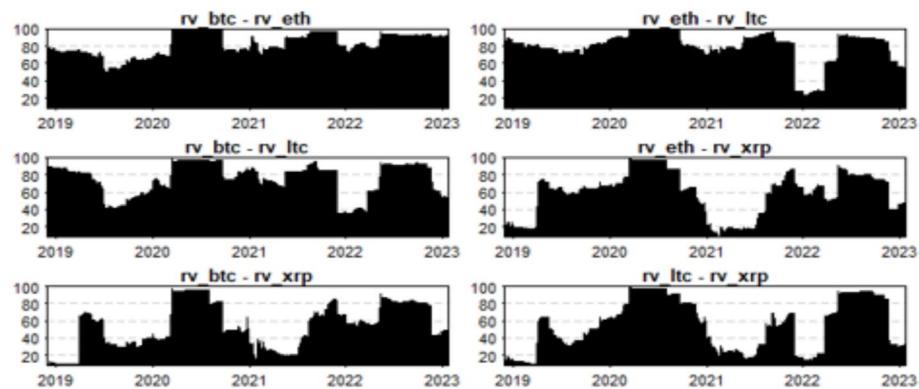


Fig. 8 Pairwise connectedness indices (PCIs)

within a given system. If NET is greater than 0, the influence of series i on other series is greater than the influence others have on series i , identifying series i as a net transmitter of shocks and a driver within the system. Conversely, if NET is less than 0, series i is more influenced by others, making it a net receiver and driven by the system. According to Table 10, BTC and ETH are net transmitters of volatility, with positive values of BTC at 3.44 and ETH at 5.99. On the other hand, the LTC and XRP are net receivers, with negative values of -2.08 and -7.35, respectively. These results are also illustrated in Fig. 7, which underscores the dominant role that BTC and ETH play in transmitting volatility across the crypto market. In contrast, XRP is the most driven coin by the system in comparison to the other three coins.

Figure 8 shows the pairwise connectedness index (PCI) levels between different pairs of cryptocurrencies over the period from 2019 to 2023, with values ranging from 0 to 100. Notably, the connectedness between BTC and ETH in the top-left panel appears relatively high (average 78) throughout, suggesting that these two leading cryptocurrencies exhibit stronger and more consistent interdependencies than those in the other panels. Except for the BTC-ETH panel, a similar pattern observed across all pairwise panels is that while the relationships fluctuate, there are noticeable periods where one cryptocurrency dominates the volatility spillover to others. Notably, the BTC-XRP (57), BTC-LTC (62), ETH-XRP (63), ETH-LTC (69) and LTC-XRP (51) pairs

show significant ups and downs, especially during market turbulence, suggesting that BTC and ETH coins often act as volatility leaders for LTC and XRP coins (inferred from Figs. 7 and 8). Using the “*ConnectednessApproach*” package written by Gabauer and Gabauer (2022) in R Studio, the network system in Fig. 9 is automatically plotted on the basis of the results of the dynamic coin-connectedness table (Table 10). According to the manual of this package, blue nodes represent volatility transmitters, whereas volatility receivers are symbolized by yellow nodes. Consequently, this network graph highlights the interconnected nature of the cryptocurrency market, with large-cap BTC and ETH coins (i.e., blue-colored nodes) spreading stronger spillovers on the realized volatility of mid-cap LTC and XRP coins (i.e., yellow nodes) at different times.

In general, all cryptocurrency pairs, particularly from early 2020 onward, exhibit rising interconnectedness within the market, hinting at periods of greater volatility spillover between these assets. This upward trend in pairwise volatility connectedness, which coincides with the COVID-19 pandemic, diminished toward early 2021 as various vaccines developed and countries began lifting different levels of lockdown restrictions

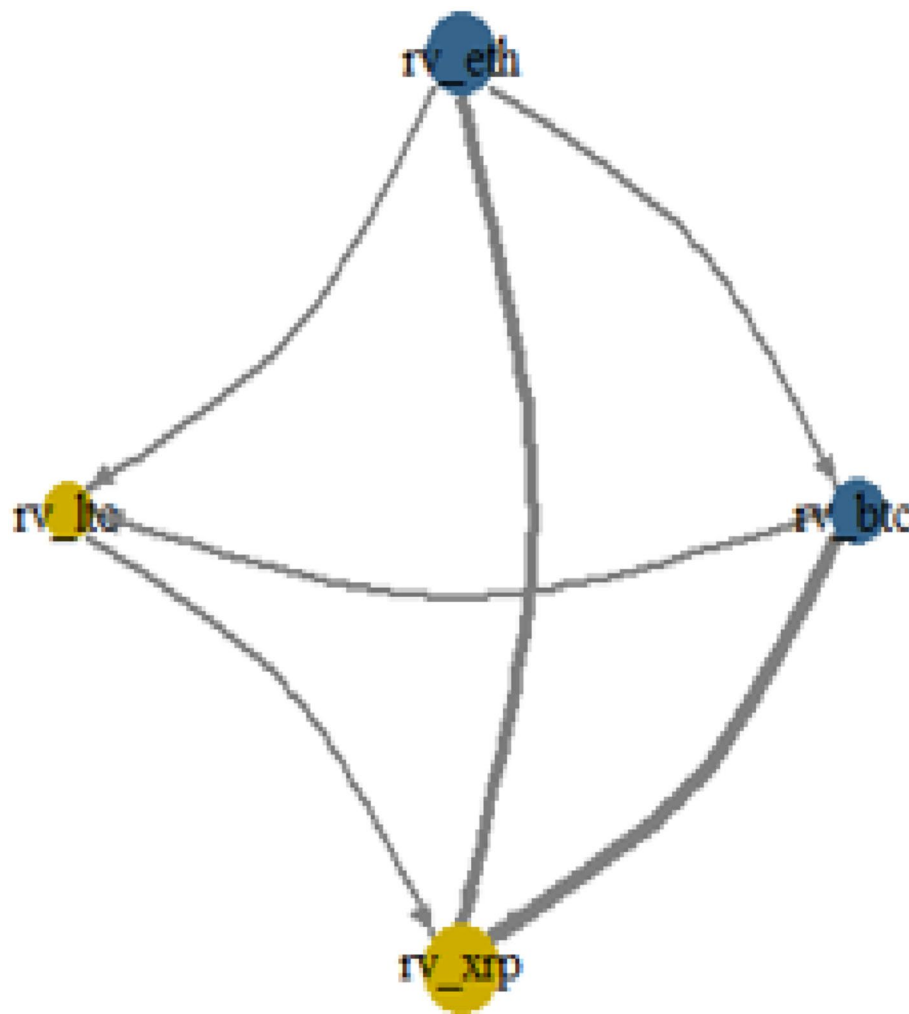


Fig. 9 Network connectedness graph for the four coins

worldwide. A renewed surge in volatility interconnectedness emerges between 2022 and 2023, coinciding with extreme cross-market fluctuations largely driven by macroeconomic factors such as rising interest rates, inflation, and the threat of a global recession, along with significant crypto events such as the Terra (LUNA) collapse, the Ethereum Merge, and the FTX collapse and trial. During this period, Bitcoin's price notably dropped below \$20,000. These factors negatively impact high-risk assets such as cryptocurrencies, leading to a market downturn. Many prominent cryptocurrencies, including Bitcoin, experienced substantial declines throughout 2022. The overall high levels of connectedness across pairs suggest that no single cryptocurrency operates in isolation, with global market trends likely influencing the interconnectedness within the cryptocurrency market.

Conclusion

Employing high-frequency cryptocurrency data, this study incorporates two distinct methods, namely, volatility forecasting and volatility connectedness. The exercise of volatility forecasting thoroughly evaluates the predictive performance of traditional time series (TS) models and machine learning (ML) models for realized volatility forecasting across four major cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and Ripple (XRP). The results show that traditional TS models, particularly the HAR and ARMA models, consistently outperform ML models in terms of predictive accuracy. Owing to its ability to capture short-, medium-, and long-term volatility dynamics, the HAR model has emerged as the best-performing model overall. Notably, ML models such as LSTM and the GFN demonstrate competitive performance in specific instances but often struggle with overfitting and capturing short-term volatility information. This study also examines cross-crypto volatility impacts by adding exogenous variables to the outperformer model. The inclusion of lagged realized volatility of BTC and ETH as a predictor can improve forecasts, particularly for mid-cap cryptocurrencies such as XRP. However, the forecasts of BTC and ETH realized volatility do not rely on external crypto information for more accurate forecasting, which indicates stability in their own volatility forecasts with similar volatility characteristics. This suggests that BTC and ETH exhibit more predictable patterns than LTC and XRP do.

Following the exercise of volatility forecasting, the analysis of realized volatility connectedness is conducted via the dynamic VAR model for a more direct and system-wide understanding of volatility spillovers. The results show that the total cross-crypto volatility connectedness seems to be quite high with time-varying features. In addition, the high levels of pairwise coin-connectedness indicate that no single coin is independent from global market trends, affecting the interconnectedness within the cryptocurrency market. However, while large-cap coins such as BTC and ETH act as volatility transmitters, mid-cap coins (e.g., XRP and LTC) appear as the main volatility receivers, particularly during market turbulence. Notably, XRP is the most driven coin by the system, making it more sensitive to cross-coin information than the other three coins are.

While our asset-specific analysis demonstrates the superiority of traditional TS models in this high-frequency setting, future research could explore the potential of panel-based machine learning models. Recent works by Zhu et al. (2023) and Zhang et al. (2024)

suggest that pooling data across multiple cryptocurrencies may mitigate overfitting and enhance generalizability by capturing shared volatility structures. Such panel frameworks could represent a promising extension to the current analysis, particularly in improving the robustness of the ML model across different market conditions.

It is important to contextualize our results within the broader literature. While our findings emphasize the robustness of HAR and ARMA models in high-frequency environments, other studies (e.g., Koo & Kim 2024) have demonstrated that ML models, especially deep learning architectures, can outperform traditional models under specific conditions, such as low-frequency data, structural breaks, or highly nonlinear dynamics. In high-frequency financial markets, however, ML models face challenges, including noise amplification and model instability, which may counteract their theoretical advantages in terms of flexibility. Thus, the relative performance of forecasting models is highly context dependent and shaped by data granularity, volatility persistence, and the forecast horizon. These findings hold significant importance for financial analysts, cryptocurrency traders, risk managers, and market participants. By identifying that traditional TS models (e.g., HAR and ARMA) outperform ML models in forecasting realized volatility, this research offers valuable insights for enhancing predictive accuracy, particularly in volatile cryptocurrency markets. Traders and investors can take advantage of these outperforming models and the unique characteristics of large-cap and mid-cap coins noted in this work to make informed decisions, improve portfolio management, and mitigate risks associated with high volatility.

Acknowledgements

Not applicable.

Author contributions

Both authors contributed equally to various tasks/parts of the paper. These include idea formation, formulation, analysis, interpretation of results. Authors jointly wrote the manuscript. Authors made detailed discussions, provided critical feedback, and shaped the research, analysis, and manuscript. Both authors read and approved the final manuscript.

Funding

None.

Data availability

Available upon request.

Declarations**Ethics approval and consent to participate**

This article does not contain any studies with human participants or animals performed by any of the authors.

Competing interest

All authors declare no conflict of interest.

Received: 31 January 2025 Accepted: 6 November 2025

Published online: 28 November 2025

References

- Andersen TG, Bollerslev T (1998) Answering the skeptics: yes, standard volatility models do provide accurate forecasts. *Int Econ Rev.* <https://doi.org/10.2307/2527343>
- Andersen TG, Bollerslev T, Diebold FX, Labys P (2003) Modelling and forecasting realized volatility. *Econometrica* 71(2):579–625
- Bouri E, Lucey B, Roubaud D (2020) Cryptocurrencies and the downside risk in equity investments. *Financ Res Lett* 33:101211
- Bouri E, Gkillas K, Gupta R, Pierdzioch C (2021) Forecasting realized volatility of bitcoin: the role of the trade war. *Comput Econ* 57:29–53
- Bucci A (2020) Realized volatility forecasting with neural networks. *J Financ Econometr* 18(3):502–531

- Chaim P, Laurini MP (2018) Volatility and return jumps in bitcoin. *Econ Lett* 173:158–163
- Christensen K, Siggaaard M, Veliyev B (2023) A machine learning approach to volatility forecasting. *J Financ Econometr* 21(5):1680–1727
- Corbet S, Meegan A, Larkin C, Lucey B, Yarovaya L (2018) Exploring the dynamic relationships between cryptocurrencies and other financial assets. *Econ Lett* 165:28–34
- Corbet S, Lucey B, Urquhart A, Yarovaya L (2019) Cryptocurrencies as a financial asset: a systematic analysis. *Int Rev Financ Anal* 62:182–199
- Corbet S, Larkin C, Lucey BM, Meegan A, Yarovaya L (2020) The impact of macroeconomic news on Bitcoin returns. *Eur J Finance* 26(14):1396–1416
- Corsi F (2009) A simple approximate long-memory model of realized volatility. *J Financ Econom* 7(2):174–196
- Degiannakis S, Filis G (2017) Forecasting oil price realized volatility using information channels from other asset classes. *J Int Money Finance* 76:28–49
- Diebold FX, Yilmaz K (2009) Measuring financial asset return and volatility spillovers, with application to global equity markets. *Econ J* 119(534):158–171
- Diebold FX, Yilmaz K (2012) Better to give than to receive: predictive directional measurement of volatility spillovers. *Int J Forecast* 28(1):57–66
- Ding S, Wu X, Cui T, Goodell JW, Du AM (2025) Modelling climate policy uncertainty into cryptocurrency volatilities. *Int Rev Financ Anal* 102:104030
- Farzad A, Mashayekhi H, Hassanpour H (2019) A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Comput Appl* 31:2507–2521
- Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res* 270(2):654–669
- Franses PH, Van Dijk D (2000) Non-linear time series models in empirical finance. Cambridge University Press
- Gabauer D, Gabauer MD (2022) Package ‘ConnectednessApproach’. R Package Version 1.0.
- Granger CWJ, Joyeux R (1980) An introduction to long-memory time series models and fractional differencing. *J Time Ser Anal* 1(1):15–29
- Guo L, Sang B, Tu J, Wang Y (2024) Cross-cryptocurrency return predictability. *J Econ Dyn Control* 163:104863
- Hansen PR, Lunde A, Nason JM (2003) Choosing the best volatility models: the model confidence set approach. *Oxf Bull Econ Stat* 65:839–861
- Hansen PR, Lunde A, Nason JM (2011) The model confidence set. *Econometrica* 79(2):453–497
- Hansen PR, Huang Z, Shek HH (2012) Realized GARCH: a joint model for returns and realized measures of volatility. *J Appl Econom* 27(6):877–906
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Hol E, Koopman SJ (2002) Stock index volatility forecasting with high frequency data (No. 02-068/4). Tinbergen Institute Discussion Paper
- Koo E, Kim G (2024) Centralized decomposition approach in LSTM for Bitcoin price prediction. *Expert Syst Appl* 237:121401
- Köse N, Yildirim H, Ünal E, Lin B (2024) The Bitcoin price and Bitcoin price uncertainty: evidence of Bitcoin price volatility. *J Futures Mark* 44(4):673–695
- Kristjanpoller W (2024) A hybrid econometrics and machine learning based modeling of realized volatility of natural gas. *Financ Innov* 10(1):45
- Kumar A, Ajaz T (2019) Co-movement in crypto-currency markets: evidences from wavelet analysis. *Financ Innov* 5(1):1–17
- Levantesi S, Piscopo G, Roviello A (2025) Cryptocurrency in global dynamics: analyzing the crypto volatility index and financial markets with machine learning. *Phys A: Stat Mech Appl* 130770
- Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. *Q Appl Math* 2(2):164–168
- Liang C, Zhang Y, Li X, Ma F (2022) Which predictor is more predictive for Bitcoin volatility? And why? *Int J Financ Econ* 27(2):1947–1961
- Liu LY, Patton AJ, Sheppard K (2015) Does anything beat 5-minute realized variance? A comparison of realized measures across multiple asset classes. *J Econom* 187(1):293–311
- Liu J, Wei Y, Ma F, Wahab MIM (2017) Forecasting the realized range-based volatility using dynamic model averaging approach. *Econ Model* 61:12–26
- Lowe D, Broomhead D (1988) Multivariable functional interpolation and adaptive networks. *Complex Syst* 2(3):321–355
- Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. *J Soc Ind Appl Math* 11(2):431–441
- McMillan DG (2020) Interrelation and spillover effects between stocks and bonds: cross-market and cross-asset evidence. *Stud Econ Finance* 37(3):561–582
- McNelis PD (2005) Neural networks in finance: gaining the predictive edge in the market. Elsevier Academic Press
- Müller UA, Dacorogna MM, Davé RD, Olsen RB, Pictet OV, Von Weizsäcker JE (1997) Volatilities of different time resolutions, analysing the dynamics of market components. *J Empir Financ* 4(2–3):213–239
- Peng Y, Albuquerque PHM, de Sá JMC, Padula AJA, Montenegro MR (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
- Setiawan H, Bhaduri M (2023) Spotting the stock and crypto markets’ rings of fire: measuring change proximities among spillover dependencies within inter and intra-market asset classes. *Appl Netw Sci* 8(1):61
- Shahzad SJH, Bouri E, Roubaud D, Kristoufek L, Lucey B (2019) Is Bitcoin a better safe-haven investment than gold and commodities? *Int Rev Financ Anal* 63:322–330
- Sims CA (1980) Macroeconomics and reality. *Econometrica*. <https://doi.org/10.2307/1912017>
- Sin CYC (2013) Using CARRX models to study factors affecting the volatilities of Asian equity markets. *N. Am. J Econ Financ* 26:552–564

- Srivastava M, Rao A, Parihar JS, Chavriya S, Singh S (2023) What do the AI methods tell us about predicting price volatility of key natural resources: evidence from hyperparameter tuning. *Resour Policy* 80:103249
- Theil H, Beerens GAC, Tilanus CG, De Leeuw CB (1966) *Applied economic forecasting*, vol 4. North-Holland Publishing Company, Amsterdam
- Thomas AJ, Petridis M, Walters SD, Gheytaasi SM, Morgan RE (2017) Two hidden layers are usually better than one. In *Engineering applications of neural networks: 18th international conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings*. Springer International Publishing, pp. 279–290
- Wang Y, Ma F, Wei Y, Wu C (2016) Forecasting realized volatility in a changing world: a dynamic model averaging approach. *J Bank Financ* 64:136–149
- Wang J, Ma F, Bouri E, Guo Y (2023a) Which factors drive Bitcoin volatility: macroeconomic, technical, or both? *J Forecast* 42(4):970–988
- Wang Y, Andreeva G, Martin-Barragan B (2023b) Machine learning approaches to forecasting cryptocurrency volatility: considering internal and external determinants. *Int Rev Financ Anal* 90:102914
- Wang Y, Chen Z, Ji X (2023c) Cross-market information transmission and stock market volatility prediction. *N Am J Econ Financ* 68:101977
- Wu J, Xu K, Zheng X, Chen J (2021) Fractional cointegration in bitcoin spot and futures markets. *J Futures Mark* 41(9):1478–1494
- Yu H, Wilamowski BM (2018) Levenberg–marquardt training. *Intelligent Systems*. CRC Press, pp 12–21
- Zhang Y, Ma F, Liao Y (2020) Forecasting global equity market volatilities. *Int J Forecast* 36(4):1454–1475
- Zhang C, Zhang Y, Cucuringu M, Qian Z (2024) Volatility forecasting with machine learning and intraday commonality. *J Financ Econom* 22(2):492–530
- Zhu H, Bai L, He L, Liu Z (2023) Forecasting realized volatility with machine learning: panel data perspective. *J Empir Finance* 73:251–271

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.