MAKÜ | Journal of Mehmet Akif Ersoy University
Economics and Administrative Sciences Faculty

e-ISSN: 2149-1658
Volume: 12 / Issue: 2
June, 2025
pp.: 542-558

Deep Neural Networks in Predicting Financial Volatility: An Application on Cryptocurrencies

Cemile ÖZGÜR[1]

Abstract

The main aim of this research is to obtain daily volatility predictions of cryptocurrencies by using the Deep Belief Network – Deep Neural Network (DBN-DNN) and Backpropagation Neural Network (BPNN) algorithms as well as the traditional Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and one of its asymmetric variants, the Exponential GARCH models. The research data consists of daily price series of the most well-known two cryptocurrencies (Bitcoin and Ethereum) for the period of 4 July 2022 and 3 February 2024. Performance of the out-of-sample volatility forecasts of the models are compared by using three different metrics, that are Root Mean Squared Error (RMSE), R squared (R^2) and Mean Absolute Error (MAE). According to the obtained performance measures estimated from the out-of-sample period of 11 October 2023 and 3 February 2024, DBN-DNN and BPNN yielded the best RMSE and MAE values of Bitcoin, respectively. On the other hand, DBN-DNN obtained the best RMSE and R^2 among the Ethereum volatility predictions. Overall, the findings of this research reveal a promising cryptocurrency volatility forecasting ability in favor of the applied deep neural networks, since they managed to obtain most of the best as well as the second-best performance metrics of both cryptocurrencies.

Keywords: Cryptocurrency, Volatility, Deep Belief Network, Deep Neural Networks.

1. Dr., Boğaziçi University,
cemile.ozgur@pt.bogazici.edu.tr,
https://orcid.org/0000-0001-8366-6745

https://doi.org/10.30798/makuiibf.1530528

Article Type	Application Date	Admission Date
Research Article	August 8, 2024	June 3, 2025

To cite this article: Özgür, C. (2025). Deep neural networks in predicting financial volatility: An application on cryptocurrencies. *Journal of Mehmet Akif Ersoy University Economics and Administrative Sciences Faculty*, *12*(2), 542-558. https://doi.org/10.30798/makuiibf.1530528

1. INTRODUCTION

Modelling and forecasting volatility of financial time series is a long-standing and challenging research subject. Additional to the well-known financial assets, cryptocurrencies, that are digital currencies supported by the blockchain technology, gained a prominent attention by the researchers. The first cryptocurrency proposed by Nakamoto (2008) is Bitcoin. Following the invention of Bitcoin (BTC), numerous cryptocurrencies are developed and traded at markets reaching billion dollars of market capitalizations. However, modelling volatility of cryptocurrencies is a trivial task, especially when the high volatility and chaotic patters of the price series are considered (Lahmiri & Bekiros, 2018; Liu, 2019; Pratas et al., 2023).

In economics and finance literature, Autoregressive Conditional Heteroskedasticity (ARCH) (Engle, 1982) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (Bollerslev, 1986) processes are the most popular volatility modelling tools. After Engle (1982) and Bollerslev (1986), an increasing number of asymmetric variants of GARCH models are proposed (e.g. Exponential GARCH (Nelson, 1991), Threshold GARCH (Zakoian, 1994) and so on). Additional to the econometric methods, currently academic researchers and/or practicians are looking forward to use the technological development and utilize various machine and/or deep learning algorithms to predict the volatility of financial time series. Deep learning (DL) is a subclass of Machine Learning (ML) defining deeper Artificial Neural Network (ANN) structures that have many layers. As a tool capable of modelling complex and nonlinear dependencies, employing DL to predict cryptocurrency volatility is an important and a challenging research subject. Nevertheless, there are still a few papers employing DL algorithms to model and predict volatility of cryptocurrencies (see for example discussion of Wang et al. (2023) and García-Medina and Aguayo-Moreno (2024) about the research gap on the subject). Moreover, the ones focusing on volatility mainly employ only BTC or concentrated on the applications of Long Short-Term Memory (LSTM) and/or Recurrent Neural Networks (RNN).

In this paper, daily volatility of two cryptocurrencies (BTC and Ethereum (ETH)) is predicted by Backpropagation Neural Network (BPNN) as well as Deep Neural Networks (DNN) using weights trained by Deep Belief Networks (DBN) that are called DBN-DNN (Deng, 2014). Out-of-sample volatility predictions of BPNN and DBN-DNN with two different activation functions (either the sigmoid or the hyperbolic tangent activation function) are compared with the traditional volatility forecasting tools, GARCH and Exponential GARCH (EGARCH), using the commonly employed performance measures of Root Mean Squared Error (RMSE), R^2 and Mean Absolute Error (MAE).

While there are various applications of Deep Belief Networks for example in biomedical engineering (Abdel-Zaher & Eldeib, 2016), meteorology (Du et al., 2018) as well as in economics and finance (Chen et al., 2019; Karathanasopoulos, 2017; Li & Sun, 2023), any published research focusing on cryptocurrency volatility is not found. It is also important to note that some papers do not differentiate DBN from DBN-DNN and call the whole process of weight initialization of DNN using DBN as a DBN

network, see also the discussion of Deng (2014) on the subject. Moreover, research papers that use DBN or DBN-DNN, calibrate optimal parameters of the algorithms without a consistent parameter selection method, such as the time series cross-validation (ts-cv) approach (Hyndman & Athanasopoulos, 2018). As a result, the main contribution of this research paper relies on the uniqueness of its methodology to predict volatility of the two cryptocurrencies.

The following sections of this paper give a brief review of the literature and explain the research methodology. The experimental design section consists of the description of the data, data pre-processing, volatility predictions as well as the calibration of the algorithms. The last two sections report the empirical findings and conclude the paper with the final remarks.

2. LITERATURE REVIEW

Utilizing ML in various fields/tasks of finance is a fast-developing and a prominent research topic, see for example Nazareth and Ramana Reddy (2023)'s paper in which the authors give an extensive review on the financial applications of ML algorithms. However, applications of ML and/or DL to especially model and predict financial volatility of cryptocurrencies are still scarce. One of the early works on this subject is Shah and Zhang (2014)'s work in which the authors predicted the future Bitcoin price variation. On the other hand, Pichl and Kaizoji (2017) employed the Heterogeneous Autoregressive model for Realized Volatility (HARRVJ) RV with jumps and the Artificial Neural Networks (ANN) to predict realized volatility and price of bitcoin, respectively. The research results indicate the well ability of the applied models to capture the dynamics of Bitcoin price and realized volatility series.

Peng et al. (2018) predicted daily and hourly volatility of three cryptocurrencies (Bitcoin, Ethereum, Dash) by employing the GARCH processes (EGARCH, GARCH and gjrGARCH) as well as Support Vector Regression (SVR) and combinations of GARCH and SVR models, which they called SVR-GARCH. According to the estimated performance metrics, the researchers argue that the applied SVR-GARCH models were able to outperform volatility forecasting performance of all the rest GARCH processes (EGARCH, GARCH and gjrGARCH using either Normal, Student's t or Skewed Student's t distributions). Moreover, Kristjanpoller and Minutolo (2018) predicted Bitcoin price volatility by using a combination of GARCH and ANN consisting of Principal Component Analysis (PCA) as a pre-processing step. The researchers evaluated and compared performance of twelve hybrid models with the predictions of the best GARCH process (the one with the lowest MSE) for three different volatility windows (10-day, 22-day and 44-day volatility). According to the obtained empirical results, the authors argue in favor of the applied hybrid ANN models consisting of technical indicators as model inputs and a PCA pre-processing step.

Jang and Lee (2018) applied Bayesian Neural Networks (BNN) and Support Vector Regression (SVR) to predict Bitcoin price and volatility by employing blockchain data (e.g. hush rate, trading

volume, average block size), exchange rates (GBP/USD, JPY/USD, EUR/USD, CNY/USD, CHF/USD) and macroeconomic variables (such as, Crude oil, gold, and VIX) as model features. Two performance measures that are RMSE and MAPE are used to compare volatility predictions of BNN, SVR and Linear Regression. According to the results of the research, the authors stress that BNN can be considered as a more reliable model for defining the log volatility of Bitcoin compared to the applied benchmark models. Miura et al. (2019) employed several machine learning algorithms (Ridge Regression, LSTM, ANN, GRU and SVM) to predict realized volatility (RV) of Bitcoin. RV prediction performance of the applied algorithms is evaluated by using the mean squared error (MSE) and root mean squared error (RMSE) metrics. Overall, the researchers argue that Ridge regression can be classified as the best and SVM as the worst performer in predicting RV of Bitcoin.

Seo and Kim (2020) also employed a hybrid of ML (ANN, Higher Order Neural Network (HONN)) and GARCH models to predict realized volatility of Bitcoin. Prediction performance of the applied models is compared with the RMSE, MAE and MAPE measures as well as the Model Confidence Set test. The results of the research point out that volatility prediction accuracy of the HONN based hybrid models is better than the rest applied in the paper. Furthermore, Shen et al. (2021) compared out-of-sample Bitcoin volatility and Value at Risk (VaR) predictions of GARCH as well as EWMA models to the predictions of Recurrent Neural Networks (RNN). The researchers used RMSE and MAE performance metrics to compare the applied models' volatility forecasts. On the other hand, they employed unconditional coverage (Kupiec, 1995) and conditional coverage (Christoffersen, 1998) tests to back test model specific VaR predictions. According to the estimated out-of-sample performance metrics, RNN is found to outperform GARCH and EWMA in terms of the MAE metric. However, GARCH and EWMA outperformed RNN in terms of the RMSE metric as well as the applied VaR back tests.

Zahid et al. (2022) is also among the researchers applying a hybrid of GARCH models (EGARCH, GARCH and gjrGARCH) and deep learning (DL) algorithms (Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM) and Gated Recurrent Unit (GRU)) to forecast realized volatility of Bitcoin. The results of this research are also in favor of the constructed GARCH-DL hybrid models by reporting a better forecasting accuracy than the single GARCH models. D'Amato et al. (2022) employed a Jordan Neural Network (JNN) to predict daily as well as intraday volatility of three cryptocurrencies (Bitcoin, Ethereum and Ripple). The researchers compared volatility predictions of JNN to the Non-Linear Autoregressive Neural Network (NLANN) and SETAR models by using the MSE and MAPE performance measures. Overall, results of the research classify JNN as a better volatility predictor than the SETAR and NLANN both for daily and intraday observations.

Wang et al. (2023) applied Random Forest (RF) and LSTM to predict volatility of four cryptocurrencies (Bitcoin, Ripple, Ethereum and Litecoin) for three different time horizons (daily, weekly, and monthly). The researchers used directional accuracy, RMSE, MAPE and NMSE metrics to

compare volatility prediction performance of the applied models. Research results of Wang et al. (2023) indicate an outperformance of the applied ML algorithms with especially additional determinants (internal and/or determinants from several cryptocurrencies) compared to the traditional GARCH. Pratas et al. (2023) employed several DL algorithms (LSTM, RNN and MLP) to forecast and compare return volatility of Bitcoin to the forecasts of the traditional ARCH and GARCH processes. As in many of the reviewed papers, the researchers used MAE and MAPE metrics to compare volatility prediction performance of the models. Moreover, the Diebold–Mariano test is also applied for the statistical significance of the predictions. According to the estimated performance metrics, the authors argue that the applied DL algorithms managed to produce superior volatility forecasts, especially in the short-term. Therefore, DL should be considered in predicting complex volatility dynamics of not only cryptocurrencies but also the other financial time series.

On the other hand, Khan et al. (2023) applied three ML algorithms (NNETAR, GMDH-NN and CSS) to forecast volatility of Bitcoin, Tether, XRP and Ethereum considering the period of 14 April 2017 to 30 October 2020. According to the estimated RMSE and MAE values, the authors were not able to rank a specific algorithm as the best volatility forecaster for all the four cryptocurrencies uniformly. While cubic smoothing spline (CSS) obtained the best error metrics of Bitcoin and XRP, NNETAR was the best performer of Ethereum's volatility predictions and GMDH-NN was the winner in case of Tether. García-Medina and Aguayo-Moreno (2024) also employed deep learning algorithms (MLP, LSTM), the traditional GARCH models (EGARCH, gjrGARCH) and their hybrids (LSTM-EGARCH and LSTM-gjrGARCH) to forecast volatility of ten cryptocurrencies. According to the empirical findings, the authors argue that since MLP was able to outperform the rest, simple learning models can be mostly sufficient to model and predict cryptocurrency volatility.

Dudek et al. (2024) conducted extensive comparative research among twelve statistical and ML models for the task of predicting daily and weekly volatility of four cryptocurrencies, that are Bitcoin, Monero, Ethereum and Litecoin. The researchers used MSE, MAE as well as the model confidence set test to compare performance of the cryptocurrency specific volatility predictions of the models. Referring to the obtained empirical results, the researchers point out that performance of the models are not uniform across cryptocurrencies as well as the applied performance measures. Different models ranked as the best depending on the error metric and cryptocurrency. Moreover, the researchers argue that simple and linear ML models can produce volatility forecasts as well as their more complex counterparts. Table 1 presents a methodological review of the cryptocurrency volatility prediction literature employing various ML and/or DL algorithms. Additionally, one can also consult to Ozbayoglu et al. (2020) and Zhang et al. (2024)'s studies in which the researchers supplied a comprehensive literature review on various applications of ML and/or DL in cryptocurrencies.

Table 1. Methodological Review of the Literature

Authors	Predicted Variable(s)	Methods	Performance Metrics
Pichl and Kaizoji (2017)	Bitcoin (BTC) price and volatility	Heterogeneous Autoregressive model for Realized Volatility, Artificial Neural Network (ANN)	Comparison of the actual and predicted vales, metrics are not specified
Vo and Xu (2017)	BTC returns and volatility	ARMA(1,2)-fGARCH(2,2)/TGARCH using the GHYP distribution, Support Vector Machine (SVM), Neural Networks (NN)	Residual Sum of Squares, Normalized Residual Mean Square Error
Guo et al. (2018)	BTC volatility	Temporal Mixture models, GARCH, Random Forests (RF), Extreme Gradient Boosting, Elastic-Net, LSTM	Root Mean Squared Error (RMSE), Mean Absolute Error
Jang and Lee (2018)	BTC price and volatility	Linear Regression, Bayesian Neural Networks (BNN), Support Vector Regression (SVR)	RMSE, Mean Absolute Percentage Error (MAPE)
Kristjanpoller and Minutolo (2018)	BTC price volatility	Hybrid of ANN-GARCH with PCA pre processing	Mean Squared Error (MSE), the Model Confidence Set (MCS)
Peng et al. (2018)	Volatility of BTC, ETH, Dash	EGARCH, GARCH, gjrGARCH, SVR and combinations of GARCH and SVR models	RMSE, Mean Absolute Error (MAE), Diebold-Mariano test
Miura et al. (2019)	BTC volatility	Ridge Regression, LSTM, ANN, Gated Recurrent Unit (GRU), SVM	MSE, RMSE
Seo and Kim (2020)	BTC volatility	Hybrid of ANN, Higher Order Neural Network and GARCH models	RMSE, MAE, MAPE, MCS
Shen et al. (2021)	BTC volatility	EWMA, GARCH, Recurrent Neural Networks (RNN)	RMSE, MAE
D'Amato et al. (2022)	Volatility of BTC, ETH, XRP	Jordan Neural Network (JNN), Non-Linear Autoregressive Neural Network (NLANN)	MSE, MAPE
Zahid et al. (2022)	BTC volatility	Hybrid of EGARCH, GARCH, gjrGARCH and LSTM, BiLSTM, GRU	Heteroscedasticity adjusted MSE and Heteroscedasticity adjusted MAE
Azizi et al. (2023)	BTC volatility	Differential equation + Artificial Neural Network (ANN), ARCH, GARCH, Deep Learning (DL)	Relative error
Khan et al. (2023)	Volatility of BTC, Tether, XRP, ETH	Neural Network Autoregressive (NNETAR), Group Method of Data Handling Neural Network, Cubic Smoothing Spline (CSS)	RMSE, MAE
Pratas et. al. (2023)	BTC volatility	ARCH, GARCH, Multilayer Perceptron (MLP), RNN, LSTM	MAE, MAPE, Diebold-Mariano test
Wang et al. (2023)	Volatility of BTC, Ripple, ETH, Litecoin	RF, Long Short-Term Memory (LSTM)	Directional accuracy, RMSE, MAPE, Normalized Mean Squared Error
Abarghouie et al. (2024)	Volatility of eight cryptocurrencies	Long Short-Term Memory (LSTM)	MSE, RMSE
Dudek et al. (2024)	Volatility of BTC, Monero, ETH, Litecoin	Autoregressive Fractionally Integrated Moving Average (ARFIMA), GARCH, Ridge Regression, RF, MLP, SVR, LSTM	MSE, MAE, MCS
García-Medina and Aguayo-Moreno (2024)	Volatility of ten cryptocurrencies	EGARCH, gjrGARCH, MLP, LSTM, LSTM-EGARCH, and LSTM-gjrGARCH	HAE, HSE, Diebold-Mariano test

3. RESEARCH METHODOLOGY

3.1. ARMA(*P,Q*)

The ARMA(*P,Q*) process modelling the conditional mean (μ_t) in Equation 1 can be defined as follows:

$$y_t = \mu_t + \epsilon_t \tag{1}$$

$$\mu_t = \varphi_0 + \sum_{k=1}^{P} \varphi_k y_{t-k} + \sum_{s=1}^{Q} \theta_s \epsilon_{t-s} \tag{2}$$

Where y_t is the return of a cryptocurrency observed at time *t*. φ_k and θ_s are the cryptocurrency specific parameters of the conditional mean equation.

3.2. GARCH(*p,q*)

Following the Autoregressive Conditional Heteroskedasticity (ARCH) model of Engle (1982), the Generalized Autoregressive Conditional Heteroskedasticity (GARCH(*p,q*)) definition of Bollerslev (1986) can be written as:

$$\epsilon_t = \sigma_t z_t \tag{3}$$

$$\sigma_t^2 = \omega + \sum_{l=1}^{p} \beta_l \sigma_{t-l}^2 + \sum_{j=1}^{q} \alpha_j \epsilon_{t-j}^2 \tag{4}$$

In Equations 1 to 4, ϵ_t defines the error term distributed according to distribution $\mathcal{D}$ $\left(\epsilon_t \sim \mathcal{D}(0, \sigma_t^2)\right)$. σ_t^2 is the conditional volatility satisfying $\omega > 0$ and $\alpha, \beta \geq 0$. The Exponential GARCH (EGARCH) model of Nelson (1991) can be defined as:

$$ln(\sigma_t^2) = \omega + \sum_{l=1}^{p} \beta_l \, ln(\sigma_{t-l}^2) + \sum_{j=1}^{q} g_j(z_{t-j}) \tag{5}$$

where $g_j(\mathrm{z}_{t-j}) = \alpha_j \mathrm{z}_{t-\mathrm{j}} + \gamma_j(|\mathrm{z}_{t-j}| - E|\mathrm{z}_{t-j}|)$.

3.3. Backpropagation Neural Networks

Dating back to the 1940s, the development of Artificial Neural Networks (ANN) arises from the ideas of being able to model biological neural networks with computation and logic (Carbonell et al., 1983). Since Rosenblatt (1962), who introduced the simplest neural networks (perceptron), various ANN network structures with several layers and different learning algorithms are developed.

Back Propagation Neural Network (BPNN) is a feed-forward ANN trained by the error back-propagation algorithm. In a typical ANN, network weights are randomly initialized and a feed-forward computation is followed:

$$y_h = f\left(\sum_{j=1}^{m}\left(x_j * \omega_{jh}\right) - \vartheta_h\right) \tag{6}$$

Where y_h and x_j are the outputs of the hidden and input layers, respectively. ϑ_h is the bias term and $f(\cdot)$ is the activation function. In this research sigmoid as well as the hyperbolic tangent activation

functions are employed. The following Equation 7 defines the feed-forward output value (o_o) of the output layer.

$$o_o = f(\sum_{h=1}^{n}(y_h * \omega_{ho}) - \vartheta_o) \tag{7}$$

Once the output of feed-forward ANN is computed, the backpropagation (BP) algorithm propagates the network error backwards from the output to the hidden layers modifying the layer weights to reduce the error. The algorithm employs the gradient descent method to minimize the network error (Li et al., 2012). Following BP, the updated network weights can be expressed as:

$$\omega'_{jh} = \omega_{jh} + \Delta\omega_{jh} \quad \wedge \quad \Delta\omega_{jh} = -\eta * \partial\mathcal{L}/\partial\omega_{jh} \tag{8}$$

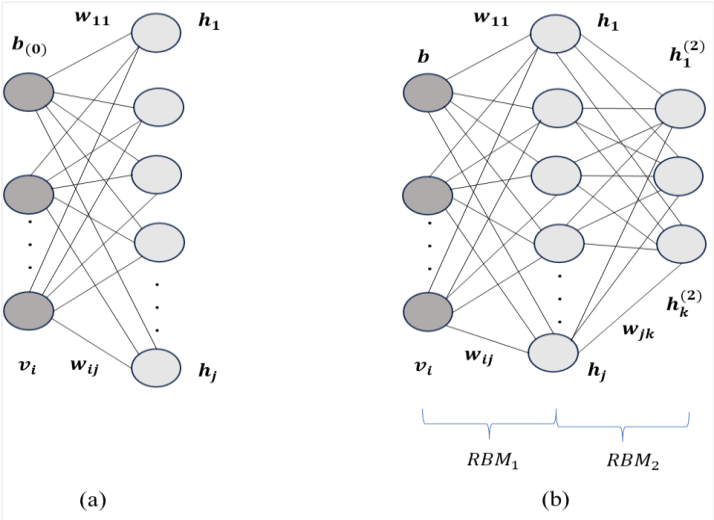
$$\omega'_{ho} = \omega_{ho} + \Delta\omega_{ho} \quad \wedge \quad \Delta\omega_{ho} = -\eta * \partial\mathcal{L}/\partial\omega_{ho} \tag{9}$$

where η is the learning rate and $\mathcal{L}$ is the difference between the actual network output and the expected (desired) level of output. The training process of error back propagation is carried until the network error converges to an acceptable level.

3.4. Deep Belief Network

Deep Belief Network (DBN), proposed by Hinton et al. (2006), is a composite multi-layer neural network with directed and undirected layer connections. DBN can be defined as a "Bayesian probabilistic generative model" constructed by stacking several layers of simple models that are called Restricted Boltzmann Machines (RBM) (Deng, 2014). On the other hand, RBM is a Boltzmann Machine with a restriction of not having connections between the hidden units as well as within the same layer. Figure 1 shows an illustration of a typical RBM that has a visual (v) and a hidden layer (h). While visual layer consists of input data, the hidden layer represents the set of features that can define the data specific dependencies.

Figure 1. An illustration of RBM (a) and DBN (b)



Source: Author's own illustration

A RBM can be trained using the Gibbs sampling and the Contrastive Divergence (CD) algorithm. Once a layer of RBM is trained, the output of the layer is used as an input to the next layer. This greedy layer-by-layer training process is called pretraining. The unsupervised pretraining process of each RBM proceeds sequentially until the last one of the DBN network is trained. Figure 1 also illustrates a DBN stacked by two RBM. Following the pretraining, weight updates of DBN can be written as:

$$\Delta\omega_{ij} = \eta\left[\left(v_i h_j\right)_0 - \left(v_i h_j\right)_1\right] \tag{10}$$

where η, v and h are the learning rate, visual and hidden units, respectively. ω_{ij} is the weight between visual unit *i* and hidden unit *j*. Furthermore, as mentioned in the previous section, the initial weights of a neural network are randomly generated. Instead, the updated weights of DBN can be utilized as the initial weights of a Deep Neural Network (DNN). DNN is a multi-layer ANN that has many hidden units as well as layers. If a DNN uses the weights of DBN to initialize its training, then the network is named as DBN-DNN (Deng, 2014).

4. EXPERIMENTAL DESIGN

4.1. Data

This research employs daily closing price series of two cryptocurrencies (Bitcoin (BTC-USD) and Ethereum (ETH-USD)) for the period of 4 July 2022 and 3 February 2024. While BTC is currently (as of 1st of July 2024) trading on 11363 markets with a volume of 21.55 billion dollars, ETH is trading on 9108 markets with a volume of 10.73 billion dollars. The data consisting of 580 observations of both cryptocurrencies is downloaded from the open source of Yahoo Finance (www.finance.yahoo.com) website. Daily returns are estimated from:

$$r_{t,d} = \ln\left(P_{t,d}/P_{t-1,d}\right) \tag{11}$$

Where $P_{t,d}$ is the daily closing price of cryptocurrency $d \in \{1,2\}$. Descriptive statistics of the data including the returns is given in Table 2.

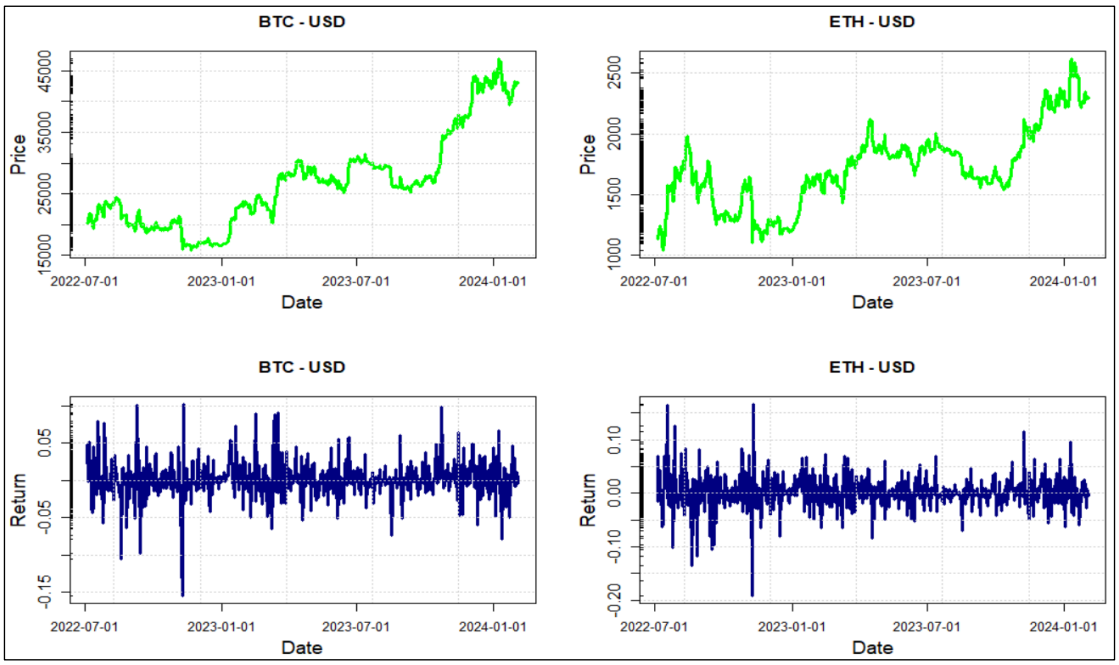
Table 2. Descriptive Statistics – Prices and Returns

Data	Mean	Stand. Dev.	Jarque-Bera (p val.)	ADF	LB(Q10)	LM(Q10)
BTC	26,816	7,589	62.83 (0.000) ***	0.655	0.000 ***	0.000 ***
ETH	1,709	320	10.06 (0.007) ***	0.357	0.000 ***	0.000 ***
BTCR	0.0014	0.0253	706.38 (0.000) ***	0.01 **	0.904	0.000 ***
ETHR	0.0013	0.0326	985.65 (0.000) ***	0.01 **	0.134	0.000 ***

Notes: *** and ** indicate significance at 1% and 5% levels, respectively. BTCR and ETHR refer to the return series of BTC and ETH, respectively. ADF is the Augmented Dickey-Fuller Test (Said & Dickey, 1984). Estimated p – values of ADF, LB (Ljung-Box (Ljung & Box, 1978)) and LM (Lagrange Multiplier test for autoregressive conditional heteroscedasticity (ARCH) of Engle (1982)) tests are reported.

As can be seen from Table 2, both price series violate the normal distribution assumption. Moreover, prices are non-stationary with the properties of serial auto-correlation and heteroskedasticity. Similarly, both return series are heteroskedastic as well as non-normally distributed. On the other hand, cryptocurrency returns do not have unit root and serial auto-correlation. Additionally, time series plots of the daily closing prices and the returns are given in Figure 2.

Figure 2. Time Series Plots of the Cryptocurrencies



Source: Author's own illustration

4.2. Data preparation

The estimated cryptocurrency returns are divided into train and test samples. 80% of the data (464 observations) of each cryptocurrency is employed to train the algorithms, the rest unseen 20% (116 out-of-sample observations) is left for the performance evaluation.

The train data of each cryptocurrency is corrected for outliers by using the Inter Quartile Range (IQR) estimate as follows:

$$IQR_d = Q_{3,d} - Q_{1,d} \tag{12}$$

$$r'_{t,d} = \begin{cases} Q_{1,d} - 1.5 * IQR_d & if \quad r_{t,d} < Q_{1,d} - 1.5 * IQR_d \\ Q_{3,d} + 1.5 * IQR_d & if \quad r_{t,d} > Q_{3,d} + 1.5 * IQR_d \\ r_{t,d} & else \end{cases} \tag{13}$$

where $Q_{1,d}$ and $Q_{3,d}$ are the first and third quartiles of the returns of cryptocurrency $d \in \{1,2\}$, respectively.

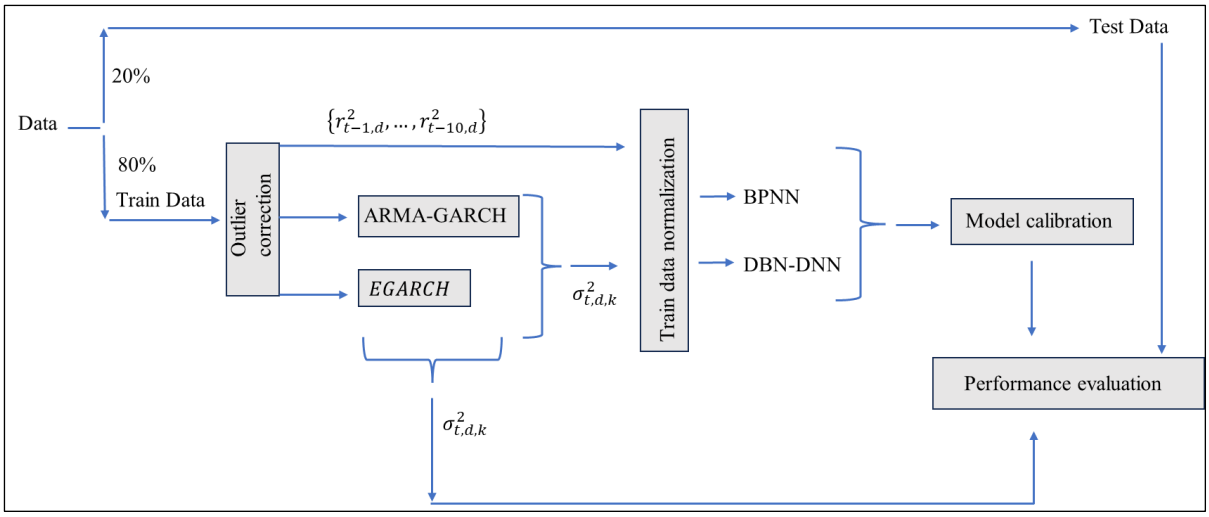
4.3. Volatility predictions

Conditional volatility forecasts of the ARMA(P,Q)-GARCH(p,q) and EGARCH(1,1) processes are obtained by using one-day ahead rolling windows approach as follows:

- Each estimation window consisting of 464 observations is corrected for outliers as explained in Equation 13 for a total of 116 windows.
- EGARCH(1,1) with student-t innovations is fitted to each window.
- The window specific (P,Q) and (p,q) orders of the ARMA-GARCH processes with either normal or student-t innovations are determined according to AIC criteria $[p, q \in \{0,1,2,3\}\ and\ P, Q \in \{0,1,2\}]$.
- Residuals of the fitted ARMA(P,Q)-GARCH(p,q) and EGARCH(1,1) processes are controlled for the remaining auto-correlation and heteroskedasticity. The orders of conditional mean and/or volatility are increased by one in case of remaining auto-correlation and/or heteroskedasticity.
- One-day ahead out-of-sample volatility forecasts are obtained.

The remaining algorithm's (BPNN and DBN-DNN) volatility predictions are obtained by using one of the commonly employed volatility proxies, daily squared returns. Figure 3 gives a clear overview of the methodology of this paper.

Figure 3. Overview of the Prediction Framework



Source: Author's own illustration

BPNN and DBN-DNN algorithms' one-day ahead volatility (squared return) predictions are obtained by employing squared cryptocurrency returns' own lagged values as well as volatility forecasts of the ARMA-GARCH and EGARCH as input variables (features). The steps are as follows:

- Train data is corrected for outliers as explained in Equation 13. Test data is left untouched.

- Train and test data are normalized by using only the train data normalization parameters to prevent leakage from the test data.
- Period specific volatility predictions of the algorithms are obtained by:

$$\hat{r}^2_{t,d} = f\left(r^2_{t-1,d}\,,\ldots,\ r^2_{t-10,d}\,,\ \sigma^2_{t,d,k}\right) + \varepsilon_t \tag{14}$$

where $r^2_{t-1,d}$ is the one-day lagged squared returns of cryptocurrency $d \in \{1,2\}$. Moreover, $\sigma^2_{t,d,k}$ is the day t volatility forecast of GARCH process k [$k \in \{ARMA-GARCH, EGARCH\}$].

4.4. Model calibration and performance

Parameters of the employed algorithms that are fine-tuned are summarized in Table 3. The optimal parameters are selected with a time series cross-validation (ts-cv) approach (Hyndman & Athanasopoulos, 2018) which keeps the order of the training data unchanged. Both BPNN and DBN-DNN algorithms are trained with a fixed rolling-window and validation samples of 116 days (h = 116). Parameters that minimize mean of the RMSE of validation samples are selected for out-of-sample volatility forecasts. The sigmoid as well as hyperbolic tangent activation functions are used both by BPNN and DBN-DNN networks. R software (R Core Team, 2019) is employed to obtain volatility predictions of the algorithms.

Table 3. Hyper-parameters of the Algorithms

Crypto	Model	Act. Func.	Learning Rate	Hidden Layers	Units of HL	Epoch
BTC	BPNN	Sigmoid	0.95	2	11/20	3
	BPNN	H. tangent	0.95	2	12/20	37
	DBN-DNN	Sigmoid	0.90	1	6	35
	DBN-DNN	H. tangent	0.95	2	9/16	48
ETH	BPNN	Sigmoid	0.95	2	8/19	3
	BPNN	H. tangent	0.95	2	8/19	27
	DBN-DNN	Sigmoid	0.95	2	17/20	3
	DBN-DNN	H. tangent	0.90	2	13/19	29

Notes: Hidden Layers refers to the number of hidden layers of the network. Units of HL indicates the number of units of each hidden layer. For example, the network design of BPNN (BTC), with the sigmoid activation function, has two hidden layers with 11 (Layer 1) and 20 (Layer 2) units. Moreover, DBN-DNN (BTC) with the sigmoid activation function, has one hidden layer consisting of 6 units. Epoch defines the iteration number of samples.

Period specific performance of the out-of-sample volatility forecasts of the models are compared with the well-known metrics of:

$$RMSE = \left(\sum_{t=1}^{N}\left(r^2_{t,d} - \hat{r}^2_{t,d}\right)^2 / N\right)^{1/2} \tag{15}$$

$$R^2 = 1 - \left(\sum_{t=1}^{N}\left(r^2_{t,d} - \hat{r}^2_{t,d}\right)^2 / \sum_{t=1}^{N}\left(r^2_{t,d} - \bar{r}^2_{d}\right)^2\right) \tag{16}$$

$$MAE = \sum_{t=1}^{N}\left|r^2_{t,d} - \hat{r}^2_{t,d}\right| / N \tag{17}$$

where $r_{t,d}^2$ and $\bar{r}_d^2$ refer to the observed and mean of the observed out-of-sample squared returns of cryptocurrency *d*, respectively. $\hat{r}_{t,d}^2$ is the volatility prediction of a model. *N* is the number of observations.

5. EMPIRICAL RESULTS

Once the algorithms are calibrated, cryptocurrency specific out-of-sample volatility predictions of the models are obtained. Table 4 and Table 5 report the estimated performance measures for BTC and ETH, respectively.

Table 4. Performance Metrics - BTC

Method	RMSE	R^2	MAE
ARMA-GARCH	0.0013604	0.0024790	0.0006900
EGARCH	0.0013638	**0.0134340**	0.0006917
BPNN*	0.0013329	0.0021690	**0.0006493**
BPNN**	*0.0013319*	*0.0118286*	*0.0006560*
DBN-DNN*	0.0013391	0.0053625	0.0006739
DBN-DNN**	**0.0013300**	0.0000352	0.0006637

Notes: The best and second-best metrics are shown in bold and italics, respectively. * and ** indicate the algorithms that use either the sigmoid (*) or the hyperbolic tangent (**) activation functions.

According to Table 4, DBN-DNN with the hyperbolic tangent activation function yielded the best RMSE value of BTC. On the other hand, EGARCH obtained the best R^2 and BPNN algorithm that use the sigmoid activation function yielded the smallest MAE metric of BTC. While different algorithms are ranked as the best depending on the performance metric, BPNN with the hyperbolic tangent activation function yielded the closest as well as the second-best values of all the three metrics of BTC.

Table 5. Performance Metrics - ETH

Method	RMSE	R^2	MAE
ARMA-GARCH	0.0017355	0.0020939	**0.0007846**
EGARCH	0.0017237	0.0035364	*0.0007888*
BPNN*	*0.0017045*	0.0032395	0.0008268
BPNN**	0.0017053	0.0028780	0.0008226
DBN-DNN*	**0.0017039**	*0.0056355*	0.0008258
DBN-DNN**	0.0017346	**0.0092906**	0.0008629

Notes: The best and second-best metrics are shown in bold and italics, respectively. * and ** indicate the algorithms that use either the sigmoid (*) or the hyperbolic tangent (**) activation functions.

When the ETH volatility predictions of the applied algorithms are compared with the same performance metrics, in this case, DBN-DNN that use the sigmoid activation function yielded the best and the second-best RMSE and R^2 values, respectively. However, DBN-DNN with the hyperbolic tangent activation function had the best R^2 of ETH. Moreover, ARMA-GARCH and EGARCH processes obtained the best and the second-best MAE. Compared to the metrics of BTC reported in Table 4, BPNN with the hyperbolic tangent activation function did not rank as the best or the second-best in any of the performance measures of ETH.

6. CONCLUSION

In this research, one-day ahead volatility predictions of BPNN and DBN-DNN deep learning algorithms that use either the sigmoid or the hyperbolic tangent activation functions, are obtained for two different cryptocurrencies, Bitcoin and Ethereum. While it is easier to find research papers employing various deep learning algorithms to forecast price or price direction of a financial asset, the research focusing on predicting cryptocurrency volatility is still scarce. As mentioned by Zahid et al. (2022) only 13.8% of published papers, which conducted research on cryptocurrencies during 2013 and 2019 used deep learning algorithms. Moreover, most of them (most of the 13.8%) that use deep learning algorithms are mainly focused on predicting price or price direction of cryptocurrencies.

In this paper, out-of-sample volatility prediction performance of BPNN and DBN-DNN are measured by using several error metrics (RMSE, R^2 and MAE) and compared to the most frequently applied volatility modelling tools in economics and finance literature, that are the (ARMA-)GARCH and EGARCH processes. From the estimated performance measures, DBN-DNN algorithm yielded the lowest RMSE values both for Bitcoin and Ethereum. While DBN-DNN still yielded the best as well as the second-best R^2 of Ethereum, EGARCH was the winner in case of R^2 of Bitcoin. Moreover, BPNN yielded the best value of MAE of Bitcoin and ARMA-GARCH had the best MAE of Ethereum. These results are in line with the works of Kristjanpoller and Minutolo (2018) as well as Zahid et al. (2022) who reported an enhanced performance when a combination of GARCH and deep learning algorithms are used to predict volatility of cryptocurrencies. Overall, this research indicates a promising volatility forecasting ability in favor of the applied deep neural networks, since they managed to obtain most of the best as well as the second-best performance metrics of both Bitcoin and Ethereum.

Future research could assess volatility prediction performance of DBN-DNN and the other underexplored deep learning algorithms by using various conventional cryptocurrencies as well as the decentralized finance (DeFi) tokens that are emerged as a new asset class (Huang & Hsu, 2024). Moreover, it would also be interesting first to explore various drivers of cryptocurrency volatility including the DeFi tokens, and then to investigate the structure of the dependence between them. Finally, ML and/or DL algorithms yielding superior volatility predictions could be employed in portfolio risk

management and asset allocation context since accurate risk prediction is vital for both investors and regulators.

Ethics Committee approval was not required for this study.

The author declares that the study was conducted in accordance with research and publication ethics.

The author confirms that no part of the study was generated, either wholly or in part, using Artificial Intelligence (AI) tools.

The author declares that there are no financial conflicts of interest involving any institution, organization, or individual associated with this article.

The author affirms that the entire research process was performed by the sole declared author of the study.

REFERENCES

Abarghouie, M. O., Alizadeh, S. H., & Khademzadeh, A. (2024). Cryptocurrency volatility prediction based on price, return and volatility cross-correlation using LSTM. In *2024 11th International Symposium on Telecommunications (IST)* (pp. 309–318). IEEE. https//doi.org/10.1109/IST64061.2024.10843571

Abdel-Zaher, A. M., & Eldeib, A. M. (2016). Breast cancer classification using deep belief networks. *Expert Systems with Applications*, *46*, 139–144. https://doi.org/10.1016/j.eswa.2015.10.015

Azizi, S. P., Huang, C. Y., Chen, T. A., Chen, S. C., & Nafei, A. (2023). Bitcoin volatility forecasting: An artificial differential equation neural network. *AIMS Mathematics*, *8*(6), 13907–13922.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*, 307–327.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine Learning* (pp. 3–23). Morgan Kaufmann.

Chen, J.-H., Hao, Y.-H., Wang, H., Wang, T., & Zheng, D.-W. (2019) Futures price prediction modeling and decision-making based on DBN deep learning. *Intelligent Data Analysis, 23*, 53–65. https://doi.org./10.3233/IDA-192742

Christoffersen, P.F. (1998). Evaluating interval forecasts. *International Economic Review, 39*(4), 841–862. https://doi.org/10.2307/2527341

D'Amato, V., Levantesi, S., & Piscopo, G. (2022). Deep learning in predicting cryptocurrency volatility. *Physica A: Statistical Mechanics and its Applications, 596*, 127158. https://doi.org/10.1016/j.physa.2022.127158

Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing, 3*(1), 1–29. https://doi.org/10.1017/atsip.2013.9

Du, J., Liu, Y., & Zhijun, L. (2018). Study of precipitation forecast based on deep belief networks. *Algorithms*, *11*(9), 132. https://doi.org/10.3390/a11090132

Dudek, G., Fiszeder, P., Kobus, P., & Orzeszko, W. (2024). Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study. *Applied Soft Computing, 151*, 111132. https://doi.org/10.1016/j.asoc.2023.111132

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, *50*(4), 987–1007.

García-Medina, A. & Aguayo-Moreno, E. (2024). LSTM–GARCH hybrid model for the prediction of volatility in cryptocurrency portfolios. *Computational Economics*, *63*, 1511–1542. https://doi.org/10.1007/s10614-023-10373-8

Guo, T., Bifet, A., & Antulov-Fantulin, N. (2018). Bitcoin volatility forecasting with a glimpse into buy and sell orders. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 989–994). IEEE. https://doi.org/10.1109/ICDM.2018.00123

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Huang, C. C., & Hsu, C. C. (2024). Evaluating dependence between DeFi tokens and conventional cryptocurrencies. *Applied Economics Letters*, *32*(7), 1053–1059.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.

Jang, H., & Lee, J. (2018). An empirical study on modeling and prediction of Bitcoin prices with Bayesian neural networks based on blockchain information. *IEEE Access*, *6*, 5427–5437.

Karathanasopoulos, A. (2017). Modelling and trading commodities with a new deep belief network. *Economics and Business Letters*, *6*(2), 28–34.

Khan, F.U., Khan, F., & Shaikh, P.A. (2023). Forecasting returns volatility of cryptocurrency by applying various deep learning algorithms. *Future Business Journal*, *9*(1), 25. https://doi.org/10.1186/s43093-023-00200-9

Kristjanpoller, W., & Minutolo, M. C. (2018). A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. *Expert Systems with Applications*, *109*, 1–11. https://doi.org/10.1016/j.eswa.2018.05.011

Kupiec, P. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, *3*(2), 73–84.

Lahmiri, S., & Bekiros, S. (2018). Chaos, randomness and multi-fractality in bitcoin market. *Chaos Solitons Fractals*, *106*, 28–34.

Li, J., Cheng, J.-h., Shi, J.-y., & Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. In D. Jin & S. Lin (Eds.) *Advances in Computer Science and Information Engineering. Advances in Intelligent and Soft Computing* (169, pp. 553–558). Springer. https://doi.org/10.1007/978-3-642-30223-7_87

Li, J., & Sun, Z. (2023). Application of deep learning in recognition of accrued earnings management. *Heliyon*, *9*, e13664. https://doi.org/10.1016/j.heliyon.2023.e13664

Liu, W. (2019). Portfolio diversification across cryptocurrencies. *Finance Research Letters*, *29*, 200–205. https://doi.org/10.1016/j.frl.2018.07.010

Ljung, G.M., & Box, G.E.P. (1978), On a measure of lack of fit in time series models. *Biometrika*, *65*(2), 297–303. https://doi.org/10.2307/2335207

Miura, R., Pichl, L., & Kaizoji, T. (2019). Artificial neural networks for realized volatility prediction in cryptocurrency time series. In H. Lu, H. Tang, & Z. Wang (Eds) *Advances in Neural Networks – ISNN 2019. Lecture Notes in Computer Science* (11554, pp. 165–172). Springer. https://doi.org/10.1007/978-3-030-22796-8_18

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Nazareth, N., & Ramana Reddy, Y. V. (2023). Financial applications of machine learning: A literature review. *Expert Systems with Applications*, *219*, 119640. https://doi.org/10.1016/j.eswa.2023.119640

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, *59*(2), 347–370.

Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing Journal*, *93*, 106384. https://doi.org/10.1016/j.asoc.2020.106384

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 Systems with Applications*, *97*, 177–192. https://doi.org/10.1016/j.eswa.2017.12.004

Pichl, L., & Kaizoji, T. (2017). Volatility analysis of bitcoin. *Quantitative Finance and Economics*, *1*(4), 474–485. https://doi.org/10.3934/QFE.2017.4.474

Pratas, T. E., Ramos, F. R., & Rubio, L. (2023). Forecasting bitcoin volatility: Exploring the potential of deep learning. *Eurasian Economic Review*, *13*(2), 285–305. https://doi.org/10.1007/s40822-023-00232-0

R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.r-project.org/

Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan.

Said, S. E., & Dickey, D.A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, *71*(3), 599–607.

Seo, M., & Kim, G. (2020). Hybrid forecasting models based on the neural networks for the volatility of bitcoin. *Applied Sciences*, *10*(14), 4768. https://doi.org/10.3390/app10144768

Shah, D., & Zhang, K. (2014). Bayesian regression and bitcoin. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, (pp. 409–414). IEEE. https://ieeexplore.ieee.org/abstract/document/7028484

Shen, Z., Wan, Q., & Leatham, D. J. (2021). Bitcoin return volatility forecasting: A comparative study between GARCH and RNN. *Journal of Risk and Financial Management*, *14*(7), 337. https://doi.org/10.3390/jrfm14070337

Vo, N. N. Y., & Xu, G. (2017). The volatility of Bitcoin returns and its correlation to financial markets. In *2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC)* (pp. 1–6). IEEE. https//doi.org/10.1109/BESC.2017.8256365

Wang, Y., Andreeva, G., & Martin-Barragan, B. (2023). Machine learning approaches to forecasting cryptocurrency volatility: Considering internal and external determinants. *International Review of Financial Analysis*, *90*, 102914. https://doi.org/10.1016/j.irfa.2023.102914

Zahid, M., Iqbal, F., & Koutmos, D. (2022). Forecasting bitcoin volatility using hybrid GARCH models with machine learning. *Risks*, *10*(12), 237. https://doi.org/10.3390/risks10120237

Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, *18*(5), 931–955.

Zhang, J., Cai, K., & Wen, J. (2024). A survey of deep learning applications in cryptocurrency. *iScience*, *27*(1), 108509. https://doi.org/10.1016/j.isci.2023.108509