Physica A 596 (2022) 127158

Contents lists available at ScienceDirect

Physica A

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

Deep learning in predicting cryptocurrency volatility

Valeria D'Amato [a], Susanna Levantesi [b], Gabriella Piscopo [c,*]

[a] Department of Pharmacy, University of Salerno, Via Giovanni Paolo, II 132, I-84084 Fisciano, Salerno, Italy
[b] Department of Statistics, Sapienza University of Rome, Viale Regina Elena, 295, 00161 Roma, Italy
[c] Department of Economic and Statistical Sciences, University of Naples Federico II, Italy

ARTICLE INFO

Article history:
Received 23 December 2020
Received in revised form 10 August 2021
Available online 2 March 2022

Keywords:
Deep learning
Neural networks
Cryptocurrency
Volatility

ABSTRACT

This paper focuses on the prediction of cryptocurrency volatility. The stock market volatility represents a very influential aspect that affects a wide range of decisions in business and finance. Recently, the volatility spillovers between the cryptocurrency market and other financial markets are detecting. Nevertheless, the cryptocurrency volatility forecasts underperform the market dynamics. This paper develops a suitable model to capture the cryptocurrency volatility dynamics. We base on deep learning techniques, which produce more reliable results than standard methods in finance by capturing complex data interactions. Specifically, we refer to a Jordan Neural Network, which is a parsimonious recurrent neural network showing more predictability power compared to other models designed for time series, the Self Exciting Threshold Autoregressive model models and the Non-Linear Autoregressive Neural Networks. Empirical evidence is provided using data from three different cryptocurrencies, Bitcoin, Ripple, and Ethereum.

1. Introduction

The superexponential market capitalization growth of cryptocurrencies [1] attracts interest from investors, regulators, public institutions, and the scientific community. The volume of daily exchanges has been enforced by the emergence of over 170 hedge funds specialized in cryptocurrencies and bitcoin futures. The increasing popularity of cryptocurrencies has implications for various fields. The main dimensions over which the debate on the virtual market has unfolded are related to the cybersecurity treated by cryptographic problems, the vulnerability to attacks [2], to legal issues especially linked to anti-money-laundering regulations. From political, sociological, and ethical points of view, the digital platforms, the decentralized peer-to-peer network, the anonymity of transactions, the absence of supervision from public authority increase the concerns about illegal activities for instance in [3,4].

Cryptocurrency transactions also involve a huge and complex discussion on economic issues. Central banks manage the money supply to reach currency stability along with macroeconomic objectives, by adjusting the money supply to counteract destabilizing periods of inflation or deflation. Nevertheless, countercyclical monetary policies are not allowed when the money supply is predetermined. Sticky prices and wages could involve recession and high levels of unemployment caused by the impossibility of conducting counter deflationary pressures. In particular, the public lexicon has focused on one cryptocurrency regarded as the father of virtual currencies, the Bitcoin. The number of existing Bitcoins will grow geometrically until it will reach a finite limit of 21 million in around 2040 according to European Central Bank

* Corresponding author.
E-mail addresses: vdamato@unisa.it (V. D'Amato), susanna.levantesi@uniroma1.it (S. Levantesi), gabriella.piscopo@unina.it (G. Piscopo).

https://doi.org/10.1016/j.physa.2022.127158
0378-4371/

[5]. Broadly speaking, the increase of Bitcoin users involves a long-term appreciation of currency which can discourage from consuming goods and services quoted in Bitcoins, by triggering a deflationary spiral [5,6].

There is currently much debate in the literature about the existence of speculative bubbles in the fluctuations of digital currencies. It is still unclear the status of an alternative asset or speculative asset [5]. The sensitiveness of the digital currencies to the market sentiments [7–9] enforces the idea of a speculative commodity, some authors finding a substantial speculative bubble component in the cryptocurrency prices [10–13].

Due to speculative behavior in the market, the dynamics of the price movements of the cryptocurrencies show high volatility. Plenty of studies analyze the volatility properties, especially in the case of Bitcoin (as stressed in [14–19]). Other authors, as ElBahrawy et al. [1], find the stability of the market prices, on average, for the whole market from April 2013 to May 2017. In respect of volatility, Baur and Dimpfl [20] analyze the asymmetric feature of volatility which increases more in response to positive shocks than in response to negative shocks, implying an asymmetric effect that is different and depending on traders' investment in the market. Glosten and Milgrom [21] explain the distinction between informed traders (or insiders) and uninformed noise traders, according to microstructure models. In particular, Baur and Dimpfl [20] find that noise trading dominates after positive shocks while informed investors trade more after negative shocks.

As the cryptocurrency market is growing more and more mature, different stakeholders require to accurately forecast the price of cryptocurrencies, their return, and volatility [22] by considering spillover effects that carry direct implications for transactions in the market [23]. Motivated by these considerations, the key contribution of this article is to develop a suitable model to capture the cryptocurrency volatility dynamics.

In the financial time series, such as stocks and currencies returns, the GARCH model is one of the most commonly employed tools for modeling the volatility trend (for example in [24–26]). Likewise, relevant literature based on GARCH models applied to cryptocurrency volatility flourished. Nevertheless, different and sometimes discordant bunches of studies implement various GARCH-type models. In particular, the strands of literature are very heterogeneous as regards the in-depth objectives, for instance modeling conditional volatility (as in [16,27]) as well as analyzing the volatility spillover between cryptocurrencies [28–31] and relating to competing model specifications. To that purpose, so far, the ongoing debate on GARCH-type models pointed out various specifications providing the best in-sample performance. Katsiampa [16] selects an AR-CGARCH model as the preferred specification. Charles and Darne [32] replicate the study of Katsiampa [16] considering the presence of extreme observations and using jump-filtered returns and the AR(1)-GARCH (1,1) model is selected as the optimal model. Chu et al. [33] evaluate IGARCH (1,1) as the most suitable for modeling volatility. Naimy and Hayek [34] conclude that the EGARCH models present the best performances. In [35] Markov-switching GARCH models show that cryptocurrency daily log-returns exhibit regime changes in their volatility dynamics. Cheikh et al. [36] propose Smooth Transition GARCH (ST-GARCH) allowing greater flexibility as the transition from a low- to a high-volatility regime could be gradual. Despite the abundance of contributions on GARCH methodology applied to cryptocurrency volatility, the literature does not provide consolidated indications. In fact, the dispute about the most appropriate GARCH model seems to be overcome by the hybrid schemes that researchers have recently begun to explore (see, for instance, Aras [37] and Kristjanpoller and Minutolo [38]).

In recent years, along with the diffusion of artificial intelligence in various fields, including financial markets, many authors have applied machine learning techniques to predict price fluctuations. The application of these techniques to the cryptocurrency market is developing fast, mostly focused on Bitcoin prices. To predict Bitcoins prices, for instance, Madan et al. [39], applied generalized linear models (GLM), random forest, and support vector machine on Blockchain data; Jang and Lee [40] used a Bayesian neural network and McNally et al. [41] applied Autoregressive Integrated Moving Average (ARIMA) models, simple recurrent and long short-term memory (LSTM) neural networks. Rebane et al. [42] considered seq2seq (turning one sequence into another sequence) recurrent deep multi-layer neural network and compared the results to those obtained by the ARIMA model. Finally, Alessandretti et al. [43] tested the performance of three machine learning models (two based on gradient boosting decision trees and one on LSTM neural networks) to predict daily cryptocurrency prices for 1681 currencies. All these studies assert the supremacy of machine learning in predicting Bitcoin price, compared to traditional methods, such as ARIMA and GLM.

In this paper, we outline new cues of reflections arising from the new frontiers of the research about the cryptocurrency volatility forecasting obtained through a neural network and adopting a fully data-driven approach. We aim to investigate the potential of the Recurrent Neural Network (RNN), suitable to deal with time series, in predicting cryptocurrency volatility. For this purpose, we consider three different volatility measures: the first one is daily, while the others are intraday.

We augment the predictability power of RNNs by developing an approach including a feedback connection from output to inputs, i.e. by estimating a type of parsimonious RNNs called the Jordan Neural Network. We find that a compelling Jordan Neural Network shows satisfactory predictability power in comparison with other models designed for time series, like the Self Exciting Threshold Autoregressive model (SETAR) models and the Autoregressive Neural Networks.

We discuss below the reasons for choosing these methods. Aiming to the objective of the paper, we select RNNs, which are able to capture long-term dependencies. This capability is connected to the architecture of the networks where the output of each layer is stored in a context layer to be looped back in along with the input from the next layer so that RNNs gain memory of sorts. This feature allows to save the temporal nature of cryptocurrency data and to favor the RNNs over traditional feed-forward neural networks. Furthermore, we grasp interesting cues of reflection coming from [44] and [45]. In particular in [44] more sophisticated methods, such as recurrent neural networks from deep learning are suggested for improving the prediction accuracy in the cryptocurrency market. In [46], the authors show that the RNN framework outperforms the mainstream methods in financial time series. Based on these considerations, we model a parsimonious algorithm from the RNNs, i.e. the Jordan technique. We compare the obtained results with the ANN that generally are characterized by excellent performances on the training set, and are specialized to solve non-linear problems. Finally, the comparison is extended to SETAR models, which represent other major nonlinear time series models. The regime changes in the volatility dynamics have to be taken into account, to avoiding bias in the estimation results and to improve the precision of the volatility forecast (see, e.g., [47,48]). In particular, Bariviera [49] pointed out some form of regime change of Bitcoins returns, so that the regime-switching models are proposed to accurately capture the volatility dynamics (for instance, see [50]). Siu and Elliott [51] present a combination of both the SETAR model and the GARCH model, say a SETAR–GARCH model for Bitcoin return dynamics.

To the best of our knowledge, this is the first academic paper that investigates on the prediction of the cryptocurrency volatility dynamics by using a RNN.

The layout of this paper is as follows. Section 2 presents the implemented models used to describe the cryptocurrency volatility, from deep learning techniques to classical approaches suitable for time series. Section 3 is devoted to the investigation of the volatility in the cryptocurrency market with a comparison of the performance of the considered models. Final remarks are offered in Section 4.

2. Deep learning techniques versus classical approach for time series

The neural networks are adaptive statistical models developed in analogy with the brain structure. Basically, the algorithm architecture is built on simple units, called neurons by the aforementioned analogy, organized according to layers. The neurons are interlinked by a set of connection strengths or weights based on an adaption or learning process from a set of training patterns, which defines the processing ability of the network. In light of these basic structural elements, the main operation that underlies the neural networks is learning from experience. Most statistical learning problems characteristically fall into one of two learning types: supervised or unsupervised learning. In the training, a paradigm called supervised learning, from the input predictor is associated with an output response measurement by inferring a model (learn a function) that maps the response to the predictors based on the labeled training data (input–output pairs). The response is compared with a target output. The difference between the patterns determines how the weights are altered, by allowing the creation of learning rules.

In the logic of time series forecasting, the algorithm has to model the predictors of future values of time series given their past. In the forecasting problem relating to time series, the relationship between past and future can be expressed by the conditional probability distribution as a function of the past observations:

$$p\left(X_{t+d}|X_t, X_{t-1}, \ldots\right) = f\left(X_t, X_{t-1}, \ldots\right)$$

In time-series modeling, neural networks become attractive hence they can approximate any continuous function, not requiring restrictive assumptions on the underlying process from which data are generated. Neural networks allow for learning very complex functions based on multiple levels of representation, obtained by combining non-linear modules that transform the representation at one level into a representation at a higher, slightly more abstract level [52].

Deep learning applied to financial prediction problems often involving large data sets are in their infancy (for instance see [53–55]), despite they already outperform the models of the traditional financial economics or methods of statistical arbitrage and other quantitative asset management techniques exploiting the interactions in the data.

2.1. The Neural Network architecture

The most common form of machine learning, deep or not, consists in finding a predictor of an output Y given an input X. In the high-dimensional input space, a learning machine is an input–output mapping where

$$Y = F\left(X\right), \quad X = \left(X_1, \ldots, X_p\right) \tag{1}$$

where the predictor is denoted by $\hat{Y} := F\left(X\right)$.

Deep learning, which is a form of machine learning, is characterized by different layers of abstraction throughout learned features pass from the data. The architecture of the deep approach is founded on L layers, i.e. non-linear transformations applied to X, the original input. For each of the L layers, let $f_1, \ldots, f_L$ be given univariate activation functions. The high-dimensional mapping, F, can be modeled via the concatenation of univariate semi-affine functions. Let $Z^{(l)}$ denote the lth layer and so $X = Z^{(0)}$. The explicit structure of the deep prediction rule is then expressed as follows:

$$\begin{aligned}
&Z^{(1)} = f^{(1)}\left(W^{(0)}X + b^{(0)}\right), \\
&Z^{(2)} = f^{(2)}\left(W^{(1)}Z^{(1)} + b^{(1)}\right), \\
&\quad \ldots \quad \ldots \quad \ldots \quad \ldots. \\
&Z^{(L)} = f^{(L)}\left(W^{(L-1)}Z^{(L-1)} + b^{(L-1)}\right), \\
&\hat{Y}\left(X\right) = W^{(L)}Z^{(L)} + b^{(L)}
\end{aligned} \tag{2}$$

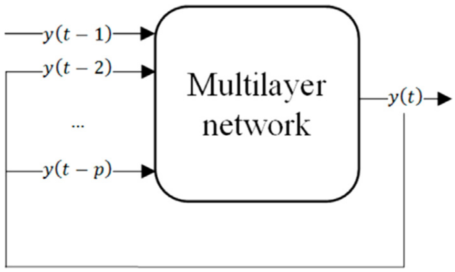

Fig. 1. Architecture of the Autoregressive Neural Network [59].

being $W^{(l)}$ the weight matrices and $b^{(l)}$ the threshold or activation levels for the *lth* layer. To solve the training problem, we have to find $\left(\widehat{W}, \hat{b}\right)$, where:

$$\widehat{W} = \left(\widehat{W}_0, \ldots, \widehat{W}_L\right) \text{ and } \hat{b} = \left(\hat{b}_0, \ldots, \hat{b}_L\right) \tag{3}$$

are learning parameters obtained by the training. On the basis of the training data $D = \left\{Y^{(i)}, X^{(i)}\right\}_{i=1}^{T}$ of input–output pairs and given a certain loss function $\Lambda\left(Y, \hat{Y}\right)$ at the level of the output signal, we have to solve the following expression:

$$\arg min_{W,b} \frac{1}{T} \sum_{i=1}^{T} \Lambda\left(Y_i, \hat{Y}^{W,b}\left(X_i\right)\right) + \lambda\varphi\left(W, b\right) \tag{4}$$

where $\varphi\left(W, b\right)$ is a regularization penalty for avoiding the overfitting and to stabilize the predictive rule according to Heaton et al. [53]. The common approach for solving Eq. (4) is a form of stochastic gradient descent which adapted to a deep learning setting is usually called back-propagation.

In the real world, non-linear systems and time series are characterized by dynamic behavior which depends on their current and past states. The Non-Linear Autoregressive Neural Network (NLANN) and the RNNs are useful to deal with dynamic inputs represented by time series sets. The Neural Network (NN) time series forecasting is a non-parametric method because the knowledge of the process that generates the time series is not required. The difference between the NLANN and RNN is that the former processes in the NN architecture the past values of the time series to predict future values; while the latter does not need past time series values as inputs nor delays, since it has recurrent connections within its own structure. In other words, RNNs are networks where connections don't only go "forward" but can connect back to previous layers.

Unlike the feed-forward neural networks, that straightforward allow signals to travel one way only from input to output, as in the case of an NLANN, the back-propagation neural networks have signals traveling in both directions due to the recursive nature of the weights and their effect on the loss which spans over time. The basics of RNNs or Multi-Layer Perceptrons (MLPs) are based on a back-propagation procedure, where the network output is fed back to its input. The RNN architecture works on sequential information and performs the same operations on every element of the input sequence. Its output, at each time step, depends on previous inputs and past computations, allowing to catch a memory of the previous events in the hidden state variables. This is the main difference with the feedforward neural networks, where all inputs are independent of each other. To detect a temporal pattern, as in financial problems, such models are suited as recently shown in the literature (for instance, Giles et al. [56] and Rout et al. [57]).

In this paper, we use the Jordan Neural Network that is a type of parsimonious RNNs, and compare its accuracy in describing cryptocurrency volatility to the NLNN and the Self Exciting Threshold Autoregressive model (SETAR), which is a general case of the classical ARIMA model. A brief description of the implemented deep learning models (NLNN and JNR respectively) is presented in Sections 2.2 and 2.3; while Section 2.4 is devoted to a description of the SETAR model.

2.2. The Non-Linear Autoregressive Neural Network

The NLANN is a type of multilayer feed forward network that designs a discrete non-linear autoregressive model, where the input variables in the Eq. (1) are *n* past observations of the output variable. As in a general feed forward structure, the number of hidden layers and neurons per layer are flexible and can be optimized through a trial and error procedure. The most common learning rule is the Levenberg–Marquardt backpropagation procedure [58], based on the approximation of the second order derivatives without the computation of the Hessian matrix in order to increase the training speed. The performance function used is the mean of the sum of the squared errors, and the training minimize this measure. The architecture of the Autoregressive Neural Network is shown in Fig. 1.

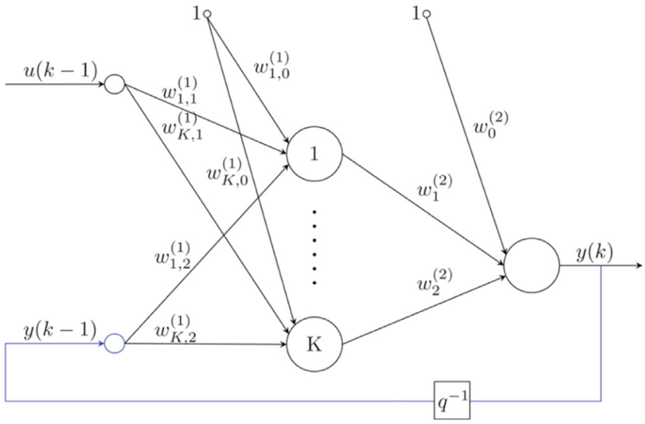

Fig. 2. Architecture of the Jordan Neural Network.

2.3. The Jordan Neural Network

The Jordan Neural Network (JNN) is an RNN proposed by Jordan [60]. It involves feedback from the output to the input layer. In the JNN, the inputs of the neurons are taken from the outputs of the network, and connected back to the inputs of the same layer, providing a memory of the previous state of this layer. Hence, the context units save the current state of the network, and this memory feature allows to deal with temporal characteristics of sequential data. Therefore, the JNN is suitable for modeling time series.

The JNN has K hidden neurons with a nonlinear transfer function and one linear output neuron (adder). Suppose that the network has K hidden neurons with a nonlinear transfer function $\varphi : R \to R$ and one linear output neuron. Let $u(k-1)$ be the input of the network at the previous instant and $y(k)$ the current output. The network is modeled as follows:

$$y\,(k) = f\,(u\,(k-1)\,, y(k-1))$$

Let define as $\omega_{i,j}^{(1)}$ the weights of layer 1, for $i = 0, \ldots, K$ and $j = 0, \ldots, 2$, while $\omega_i^{(2)}$ are the weights of layer 2 for $i = 0, \ldots, K$. The network functioning can be expressed as:

$$y\,(k) = \omega_0^{(2)} + \sum_{i=0}^{K} \omega_i^{(2)} \varphi\,(z_i(k))$$

where $z_i\,(k) = \omega_{i,0}^{(1)} + \omega_{i,1}^{(1)} u\,(k-1) + \omega_{i,2}^{(1)} y(k-1)$ is the sum of the inputs of the hidden node i.

The JNN architecture is illustrated in Fig. 2.

2.4. The SETAR models

Unlike the deep learning techniques described previously, the SETAR model belongs to the classical statistical approaches to time series.

The SETAR models, introduced by Tong in 1977 and fully developed in [61], are an extension of the autoregressive model generally applied to time series. They are a special case of the Markov switching model; for a detailed analysis see [62]. These models assume that the behavior of the series changes once the series enters a different regime.

Let m be the number of regimes, usually equal to 2 (in this case mL stays for Low regime and mH for High regime) or 3 (mL, mH, mM that stays for Medium regime) and z be the threshold variable, at which the series changes regime. Let d be the order of the AR part, the SETAR($m = 2$, d) model has the following expression:

$$\begin{aligned} x\,(t) = {} & 1_{(z(t-1) \le th)} \left(\alpha_0^{mL} + \alpha_1^{mL} x\,(t-1) + \cdots + \alpha_d^{mL} x\,(t-d)\right) \\ & + 1_{(z(t-1) > th)} \left(\alpha_0^{mH} + \alpha_1^{mH} x\,(t-1) + \cdots + \alpha_d^{mH} x\,(t-d)\right) \end{aligned} \tag{5}$$

For fixed threshold variable, the model is linear, so the parameters estimation can be done directly by Conditional Least Squares.

3. Numerical application

In this section we firstly consider the Bitcoin daily prices for the period April 28th, 2013 to December 15th, 2019, being the most actively traded and popular cryptocurrency. Secondly, we refer to Ethereum and XRP (Ripple), which are among the most common digital coins as shown in Fig. 3.

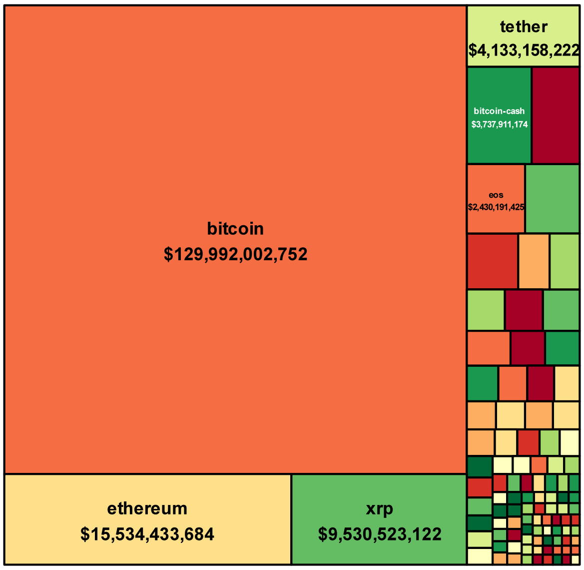

Fig. 3. Cryptocurrency Market Cap, December 2019.

We focus our analysis on the forecasting of price fluctuations based on volatility empirical evidence. Our goal is to model and predict the daily volatility measured by the difference between high and low prices (in log scale) as described in the next subsection. The analysis is then extended to two intraday volatility measures, one based on the standard deviation and the other on the interquartile of the closed prices (in log scale).

3.1. An overview of Bitcoin, Ethereum and XRP

Before using Bitcoin, Ethereum, and XRP prices as input of the predictive models, we analyze the main features of their time series. In Fig. 4 the temporal evolution of the market capitalization ("Market Cap", blue line), closing prices ("Price", green line), Bitcoin closing price ("Price (BTC)", orange line), trading volumes in 24 h ("24 h vol", gray histograms) are illustrated for the different cryptocurrencies: Bitcoin (top panel), Ethereum (middle panel), and XRP (bottom panel). The trading volumes in 24 h are represented by the histogram at the bottom of each panel of Fig. 4. The panels in Fig. 4 have been downloaded on December 16th, 2019 with all data available at the date from the website https://www.coinmarketcap.com, which gathers detailed data on the main cryptocurrencies traded in the financial market. The blue line represents the market cap, which is an essential part of understanding the cryptocurrency market, and the gray line represents the trading volume, which is an indicator of assets' importance to the market. The XRP charts show a marked volatile market.

3.2. Daily volatility estimation of Bitcoin

To deepen the movements of the cryptocurrency market, we focus on the volatility measurement as an important predictor of future dynamics. The theoretical and empirical literature on the financial volatility market and its prediction are based on well-known time series models such as GARCH as in [63] and [64], and stochastic volatility models (SV) in [65]. The financial market observes a different behavior of the traders at different periods, in the long-term perspective and the short-term or during the trading day. For example, Admati and Pfleiderer [66] showed that the market is very active immediately after the opening based on the information arriving while the market is closed. Consequently, there are different measures of historical volatility which can involve open (O), high (H), low (L), and close (C) prices.[1]

[1] The standard calculation of the historical volatility is the close-to-close, which is preferable for large sample sizes. Several volatility measures have been proposed in literature. For instance, Parkinson [67] extends the standard calculation of the volatility of returns by including the low and high price during the day. By assuming an underlying geometric Brownian motion with no drift for the prices, the Parkinson volatility is measured as: $V_{P,t} = \frac{1}{4\,ln(2)}\left[ln\left(H_t/L_t\right)\right]^2$. Otherwise, volatility can be described on the basis of open and close prices, respectively O_t and C_t, according to Garman and Klass [68]: $V_{GK,t} = \frac{1}{2}\left[ln\left(H_t\right) - ln\left(L_t\right)\right]^2 - \left[2ln(2) - 1\right]\left[ln\left(C_t\right) - ln\left(O_t\right)\right]^2$. When the sample data are continuously observed both these latter measures are unbiased [69], nevertheless the Garman–Klass volatility measure being more efficient than one of Parkinson. Chan and Lien [69] show that neither the Parkinson nor the Garman–Klass measures are efficient, in case of the drift term is not zero. While, Rogers et al. [70] proposed a volatility measure which is more suitable to measure the volatility for stocks with non-zero mean: $V_{RS,t} = \left[ln\left(H_t\right) - ln\left(O_t\right)\right]\left[ln\left(H_t\right) - ln\left(C_t\right)\right] + \left[ln\left(L_t\right) - ln\left(O_t\right)\right]\left[ln\left(L_t\right) - ln\left(C_t\right)\right]$. More recently, by assuming a geometric Brownian motion with zero drift, Yang and Zhang [71] built a historical volatility measure that allows for opening jumps and is independent of the drift. The Yang–Zhang volatility measure can be interpreted as a weighted average of the Rogers–Satchell volatility, $V_{RS,t}$, the open-close, $V_{O,t}$, and the close-open, $V_{C,t}$, volatility: $V_{YZ,t} = V_{O,t} + k \cdot V_{C,t} + (1-k)\,V_{RS,t}$.

Fig. 4. Market fluctuations of Bitcoin (top panel), Ethereum (middle panel) and XRP (bottom panel).

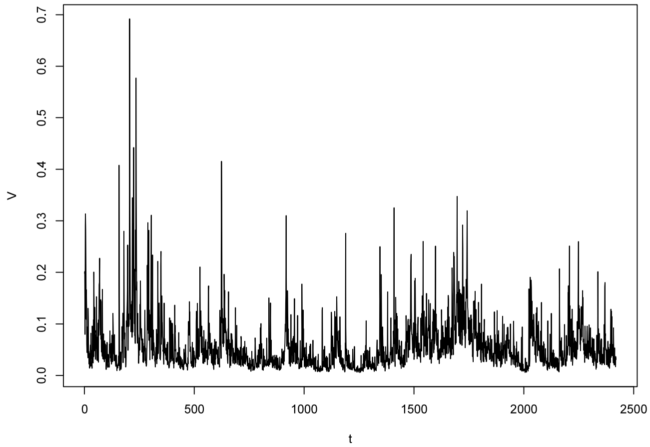

Fig. 5. Bitcoin Daily Volatility ranging from April 28th, 2013 to December 15th, 2019.

Firstly, we use a simple measure of daily volatility (at time t) defined as the first logarithmic difference between the high and low prices [72,73]:

$$V_t = ln\left(H_t\right) - ln\left(L_t\right) \tag{6}$$

where H_t and L_t are respectively high and low prices at time t. It allows to measure the daily volatility-based intraday ranges, satisfying the efficiency and consistency properties as demonstrated by the specialized literature (e.g. [71,72]).

The Bitcoin daily fluctuations of the prices typically exhibit high volatile and speculative behavior, even though the digital currency has shown remarkable signs of stability of late, as shown in Fig. 5.

The dataset under consideration is divided into training and test set (respectively 85% and 15%). Therefore, the test set ranges from December 18th, 2018 to December 15th, 2019.

We compare the predictability of the NLANN, SETAR, and JNN. To perform our analysis, the models have been run in the R CRAN environment, using the following contributed R packages: *tsDyn* for NLANN and SETAR model, *RSNNS* for JNN. While the real data has been downloaded through the *crypto* package. In the following, we briefly describe the implementation of the three models in R.

The SETAR model is implemented through the routine for the automatic selection of hyper-parameters available in the *tsDyn* package, based on an exhaustive search over all possible combinations of values of specified hyper-parameters. In this way, the threshold delay, the number of lags in each regime, and the threshold value are computed. The number of thresholds is set equal to one; the maximum autoregressive order for both low and high regime is set equal to 2; the parameters are chosen to maximize the AIC criterion. An automatic routine for the selection of the hyperparameters of the NNLAN is also available.

To design the JNN architecture, the initialization is done with the standard initialization function "JE_Weights" with five parameters [74], where the first two parameters define an interval from which the forward connections are randomly chosen. The third parameter gives the self-excitation weights of the context units. The fourth parameter gives the weights of context units between them, and the fifth parameter gives the initial activation of context units. The default architecture is implemented through the Stuttgart Neural Network Simulator: the initialization parameters are respectively set equal to (1, −1, 0.3, 1, 0.5), the learn function parameter is set equal to 0.8. The optimal tuning of the parameters is out of the scope of this research, and the default setting is chosen due to satisfactory results in terms of the goodness of the model to the others described.

For each model, we provide $b = 10{,}000$ bootstrap projections in a time horizon of 369 days according to the test dataset. To contextualize within classic statistical methods, we get the validation of the goodness of projections by comparing the projections and test values. We use mean square error (MSE) and mean absolute percentage error (MAPE) respectively calculated as:

$$MSE = \frac{1}{T}\sum_{t=1}^{T}\left(Y_t - \hat{Y}_t\right)^2; \qquad MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right|$$

where Y_t are the observed and $\hat{Y}_t$ the predicted values at time t ($t = 1, \ldots, T$).

Table 1 shows the results of MSE and MAPE for the JNN, NLANN, and SETAR models.

As we can observe, the comparison among the error measures reveals the better performance of the JNN. The Bitcoin Daily Volatility outcomes, V_t, of the three models are illustrated in Fig. 4, which shows both the test data and the fitted

Table 1
Projection errors for Bitcoin: JNN, NLANN, and SETAR models. Volatility measure: V_t.

Model	MSE	MAPE
JNN	0.000527	0.641597
NLANN	0.008381	0.834193
SETAR	0.009058	0.842966

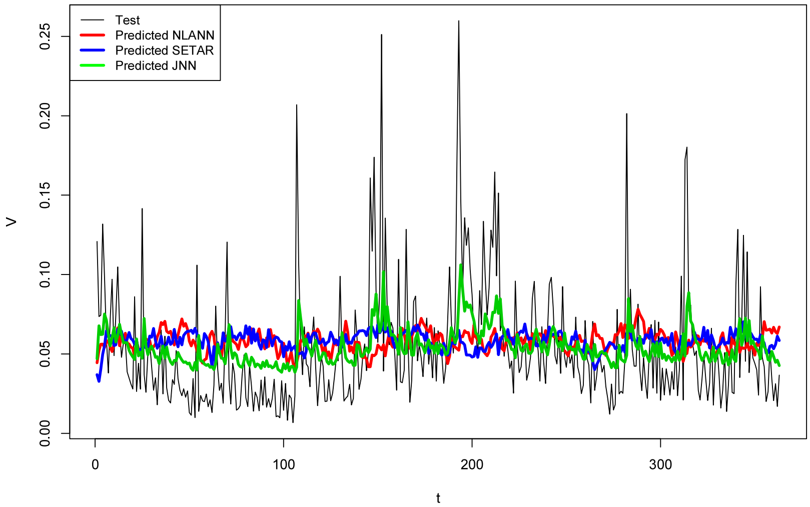

Fig. 6. Bitcoin Daily Volatility (V_t) prediction: JNN, NLANN and SETAR models.

values. We can observe significant evidence that both NNLAN and SETAR do not adequately capture the test dataset volatility, while the JNN appears to better superimpose the test dataset in terms of fluctuations and scale (Fig. 6).

Fig. 6 highlights how the volatility estimated by JNN achieves outperformance over the validation window. In other terms, the JNN deep-learning architecture based on the non-linear input–output mappings seems to capture the persistence feature of the Bitcoin time series by optimizing the predictive performance. Indeed, the JNN allows for a memory effect, where recurrence lets the network remember cues from the recent past, but does not significantly complicate the training process.

3.3. Intraday volatility estimation of Bitcoin

Until now, we have used a daily volatility measure based on daily closing prices [75], since very often are the only ones available. Nevertheless, the range-based volatility estimators can be calculated and notably the difference between high and low prices represents a natural candidate for the volatility estimation [75]. Recently several authors in literature started to use the range-based volatility (as in [72,76–80]). In line with this literature, we extend our analysis introducing a range-based estimator of intraday volatility.

Therefore, we also tested NLANN, SETAR, and JNN on other measures of intraday volatility represented by:

- the standard deviation (SD) of the log close prices:

$$V_t^{SD} = SD(ln\,(C_t)) \tag{7}$$

  where C_t are the close prices at time t, and
- the interquartile range (IQR) of the log close prices:

$$V_t^{IQR} = IQR(ln\,(C_t)) \tag{8}$$

The results in Table 2 show a comparison among the performance of the selected methods applied to the Bitcoin intraday volatility:

The findings confirm the best performance of the JNN model also for these other volatility measures based on intraday observations. Figs. 7 and 8 illustrate the test data and the estimation of the Bitcoin Intraday volatility based on the standard deviations and the interquartile range of the log close prices, respectively. However, high-frequency data are in many cases not available at all or available only over a shorter time horizon for all cryptocurrencies. Due to the limitation of the available data, this analysis has been only feasible for bitcoins.

Table 2
Projection errors for Bitcoin: JNN, NLANN, and SETAR models. Volatility measures: intraday V_t^{SD} and V_t^{IQR}.

Model	Intraday SD		Intraday IQR	
	MSE	MAPE	MSE	MAPE
JNN	7.440755e−08	1.006649	1.250798e−06	1.373698
NLANN	0.000231	1.261502	0.009813	0.941498
SETAR	0.000477	1.323710	0.008473	0.996713

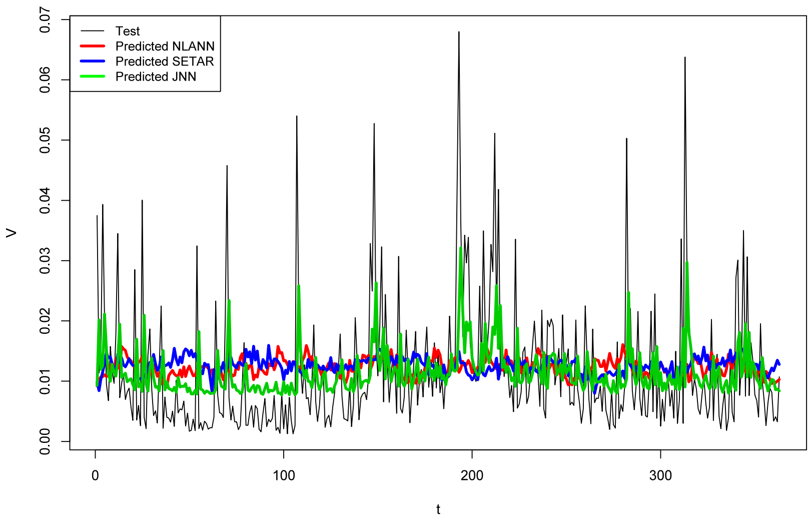

Fig. 7. Bitcoin Intraday Standard Deviation of the log close prices (V_t^{SD}) prediction: JNN, NLANN and SETAR models.

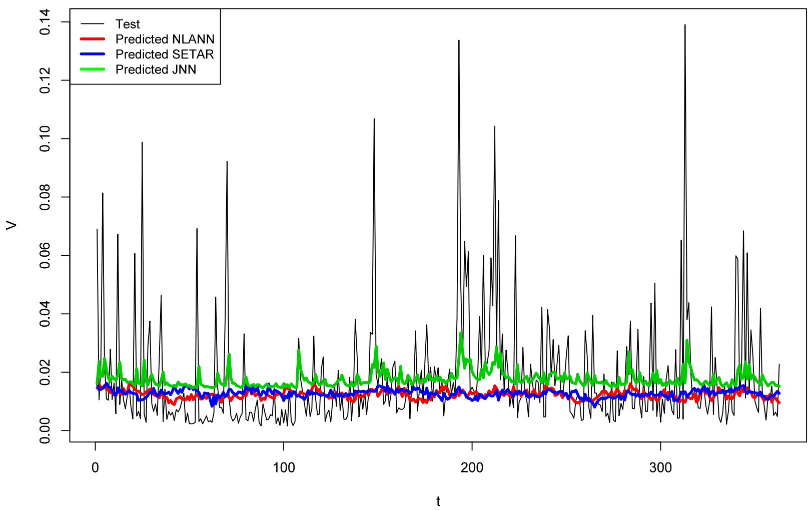

Fig. 8. Bitcoin Intraday Interquartile Range of the log close prices (V_t^{IQR}) prediction: JNN, NLANN and SETAR models.

3.4. Daily volatility estimation of Ethereum and XRP

Based on remarkable improvements that can be obtained with JNN, we develop the JNN deep architecture on other cryptocurrencies. According to cryptocurrency market capitalization rank (see Fig. 3), we consider the two other most common digital coins, Ethereum and XRP (Ripple).

Table 3 shows the projection errors for XRP and Ethereum, providing a comparison among the predicting power of JNN, NLANN and SETAR models.

Table 3
Projection errors for Ethereum and XRP: JNN, NLANN, and SETAR models. Volatility measure: V_t.

Model	XRP		Ethereum	
	MSE	MAPE	MSE	MAPE
JNN	0.086333	0.944954	0.000023	0.518782
NLANN	0.213696	0.801149	0.131523	1.095757
SETAR	0.179813	1.304910	0.214432	1.243817

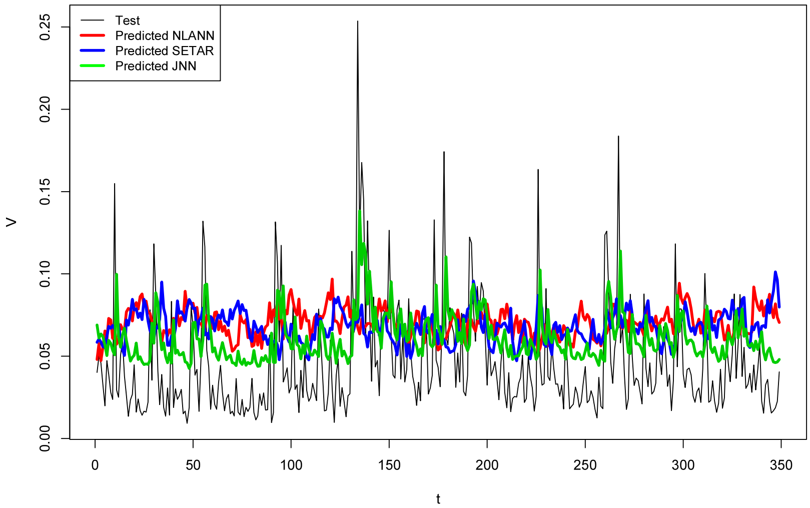

Fig. 9. XRP Daily Volatility (V_t) prediction: JNN, NLANN and SETAR models.

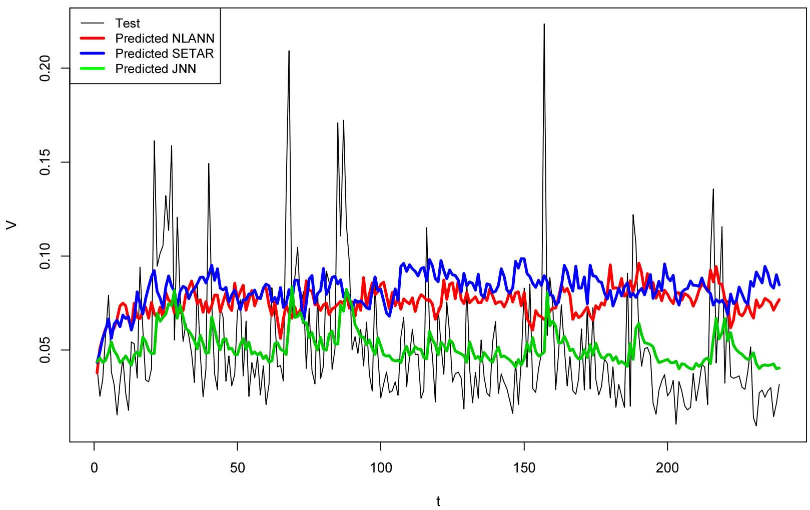

Fig. 10. Ethereum Daily Volatility (V_t) prediction: JNN, NLANN and SETAR models.

We observe that the JNN algorithm outperforms also when consider Ethereum and XRP cryptocurrencies, with the only exception of XRP for MAPE, where NLANN offers the best performance. Thanks to their advanced architecture and ways of training, the JNN is very good at predicting cryptocurrency volatility, obtaining the best performances during the test phase compared to the approaches based on NLANN and SETAR. The test data and the estimation of the XRP and Ethereum volatility V_t provided by the three models are depicted in Figs. 9 and 10, respectively.

4. Conclusions

In this paper, we have dealt with the analysis of the cryptocurrency volatility dynamics. Due to speculative behavior in the market, the dynamics of the price movements of the cryptocurrencies shows high volatility. A model able to accurately describe the daily volatility accounted in terms of variation between the high and low price realized during a day is also suited to predict the intraday return and this information is relevant for the speculators in the market. To this aim, we have investigated the possibility to implement deep learning techniques rather than classical methods. The main difference between the two approaches is that in the former case the forecasting is performed through a non-parametric method because the knowledge of the process that generates the time series is not required, while the latter is a parametric method. We have considered two types of Neural Network suitable for time series: the NLANN and the JNN, where connections do not only go "forward" but can connect to previous layers because has recurrent connections within its own structure. The results are relevant in comparison with those produced by the NLANN and the SETAR model, which is a general case of the classical ARIMA model. Due to the promising results of the JNN, the space for further research has to be identified in the implementation of other RNNs, like the LSTM Networks.

CRediT authorship contribution statement

Valeria D'Amato: Conceptualization, Validation, Writing – original draft, Writing revision, Supervision. **Susanna Levantesi:** Methodology, Software, Validation, Data curation, Writing revision, Supervision. **Gabriella Piscopo:** Conceptualization, Methodology, Software, Data curation, Writing – original draft, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. ElBahrawy, L. Alessandretti, A. Kandler, R. Pastor-Satorras, A. Baronchelli, Evolutionary dynamics of the cryptocurrency market, R. Soc. Open Sci. 4 (11) (2017) 1–9.
[2] I. Eyal, E.G. Sirer, in: N. Christin, R. Safavi-Naini (Eds.), Majority is Not Enough: Bitcoin Mining is Vulnerable, in: Lecture Notes in Computer Science, vol. 8437, Springer, Heidelberg, 2014, pp. 436–454.
[3] R. Böhme, N.N. Christin, B.B. Edelman, T. Moore, Bitcoin: Economics, technology, and governance, J. Econ. Perspect. 29 (2) (2015) 213–238.
[4] D. Ron, A.A. Shamir, How did dread pirate roberts acquire and protect his bitcoin wealth?, in: R. Böhme, M. Brenner, T. Moore, M. Smith (Eds.), Financial Cryptography and Data Security, in: Lecture Notes in Computer Science Series, vol. 8438, 2014, pp. 3–15.
[5] ECB, European Central Bank (ECB), Virtual Currency Schemes, ISBN: 978-92-899-0862-7, 2012.
[6] D.S. Evans, Economic Aspects of Bitcoin and Other Decentralized Public Ledger Currency Platforms, Coase-Sandor Institute for Law & Economics Working Paper No. 685, 2014.
[7] G.P. Dwyer, The economics of Bitcoin and other similar private digital currencies, J. Financ. Stab. 17 (C) (2015) 81–91.
[8] R.J. Shiller, Speculative asset prices, Amer. Econ. Rev. 104 (6) (2014) 1486–1517.
[9] B. Weber, Bitcoin and the legitimacy crisis of money, Camb. J. Econ. 40 (1) (2016) 17–41.
[10] K. Dowd, New Private Monies. A Bit-Part Player?, The Institute of Economic Affairs, London, ISBN: 978-0-255-36694-6, 2014.
[11] R.S. Dale, J.E. Johnson, L.L.L. Tang, Irrational behaviour during the South Sea bubble, Econ. Hist. Rev. 83 (2) (2005) 233–271.
[12] E.T. Cheah, J.M. Fry, Speculative bubbles in Bitcoinmarkets? An empirical investigation into the fundamental value of Bitcoin, Econ. Lett. 130 (C) (2015) 32–36.
[13] A. Cheung, E. Roca, J.J. Su, Crypto-currency bubbles: an application of the Phillips-Shi-Yu (2013) methodology on Mt. Gox bitcoin prices, Appl. Econ. 47 (23) (2015) 2348–2358.
[14] J. Bouoiyour, R. Selmi, Bitcoin: a beginning of a new phase? Econ. Bull. 36 (3) (2016) 1430–1440.
[15] E. Bouri, G. Azzi, A.H. Dyhrberg, On the return-volatility relationship in the Bitcoin market around the price crash of 2013, Econ.: Open-Access Open-Assess. E-J. 11 (2) (2017) 1–16, http://dx.doi.org/10.5018/economics-ejournal.ja.2017-2.
[16] K. Katsiampa, Volatility estimation for bitcoin: A comparison of GARCH models, Econom. Lett. 158 (C) (2017) 3–6.
[17] C. Conrad, O.O. Kleen, Two are better than one: Volatility forecasting using component GARCH models, Appl. Econom. (2018) 19–45.
[18] L. Catania, S. Grassi, F. Ravazzolo, Predicting the volatility of cryptocurrency time–series, in: Marco Corazza, María Durbán, Aurea Grané, Cira Perna, Marilena Sibillo (Eds.), Mathematical and Statistical Methods for Actuarial Sciences and Finance: MAF 2018, Springer, New York, 2018, pp. 203–207.
[19] J. Chiu, T.V. Koeppl, The Economics of Cryptocurrencies - Bitcoin and beyond, Staff Working Papers 19-40, Bank of Canada, 2019.
[20] D.G. Baur, T. Dimpfl, Asymmetric volatility in cryptocurrencies, Econom. Lett. 173 (C) (2018) 148–151.
[21] L.R. Glosten, P.R. Milgrom, Bid, ask and transaction prices in a specialist market with heterogeneously informed traders, J. Financ. Econ. 14 (1) (1985) 71–100.
[22] K. Katsiampa, An empirical investigation of volatility dynamics in the cryptocurrency market, Res. Int. Bus. Finance 50 (C) (2019) 322–335.
[23] J. Liu, A. Serletis, Volatility in the cryptocurrency market, Open Econ. Rev. 30 (4) (2019) 779–811.
[24] D.Z. Xing, H.F. Li, J.C. Li, C. Long, Forecasting price of financial market crash via a new nonlinear potential GARCH model, Physica A 566 (2021) 1–16.
[25] Y. Wei, Q. Yu, J. Liu, Y. Cao, Hot money and China's stock market volatility: Further evidence using the GARCH–MIDAS model, Physica A 492 (2018) 23–930.
[26] X. Yu, Y. Huang, The impact of economic policy uncertainty on stock volatility: Evidence from GARCH–MIDAS approach, Physica A 570 (2021) 1–8.
[27] A. Urquhart, The volatility of Bitcoin, 2017, https://ssrn.com/abstract=2921082 or http://dx.doi.org/10.2139/ssrn.2921082.

[28] L. Kristoufek, What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis, PLoS One 10 (4) (2015) e0123923, http://dx.doi.org/10.1371/journal.pone.0123923.
[29] A.H. Dyhrberg, Bitcoin, gold and the dollar–A GARCH volatility analysis, Finance Res. Lett. 16 (2016) 85–92.
[30] P. Ciaian, M. Rajcaniova, D. Kancs, The economics of BitCoin price formation, Appl. Econ. 48 (19) (2016) 1799–1815.
[31] A.S. Kumar, S. Anandarao, Volatility spillover in crypto-currency markets: Some evidences from GARCH and wavelet analysis, Physica A 524 (2019) 448–458.
[32] A. Charles, O. Darne, Volatility estimation for Bitcoin: Replication and robustness, Int. Econ. 157 (2018) 23–32.
[33] J. Chu, S.S. Chan, S. Nadarajah, J. Osterrieder, GARCH modelling of cryptocurrencies, J. Risk Financ. Manage. 10 (4) (2017) 1–15.
[34] V.Y. Naimy, M.R. Hayek, Modelling and predicting the bitcoin volatility using Garch models, Int. J. Math. Model. Numer. Optim. 8 (2018) 197–215.
[35] D. Ardia, K. Bluteau, K. Boudt, L. Catania, D.D. Trottier, Markov-switching GARCH models in R: The MSGARCH package, J. Stat. Softw. 91 (4) (2019) 1–38.
[36] N.B. Cheikh, Y.B. Zaied, J. Chevallier, Asymmetric volatility in cryptocurrency markets: New evidence from smooth transition GARCH models, Finance Res. Lett. 35 (2020).
[37] S. Aras, Stacking hybrid GARCH models for forecasting bitcoin volatility, Expert Syst. Appl. 174 (2021) 114747, http://dx.doi.org/10.1016/j.eswa.2021.114747.
[38] W. Kristjanpoller, M.C. Minutolo, A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis, Expert Syst. Appl. 109 (2018) 1–11.
[39] I. Madan, S. Saluja, A. Zhao, Automated Bitcoin Trading Via Machine Learning Algorithms, Vol. 20, Working paper, Stanford University, Stanford, CA, USA, 2014, pp. 1–5, http://cs229.stanford.edu/proj2015/.
[40] H. Jang, J. Lee, An empirical study on modeling and prediction of Bitcoin prices with Bayesian neural networks based on blockchain information, IEEE Access 6 (2018) 5427–5437.
[41] S. McNally, J. Roche, S. Caton, Predicting the price of bitcoin using machine learning, in: 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2018, pp. 339–343.
[42] J. Rebane, I. Karlsson, S. Denic, P. Papapetrou, Seq2Seq RNNs and ARIMA models for cryptocurrency prediction: A comparative study, in: Proceedings of SIGKDD Workshop on Fintech (SIGKDD Fintech'18), 2018, 4, URN: urn:nbn:se:su:diva-161409.
[43] L. Alessandretti, A. ElBahrawy, L.M. Aiello, A. Baronchelli, Anticipating cryptocurrency prices using machine learning, Complexity (2018) 8983590, http://dx.doi.org/10.1155/2018/8983590, Hindawi.
[44] L. Pichl, T. Kaizoji, Volatility analysis of Bitcoin price time series, Quant. Finance Econ. 1 (4) (2017) 474–485.
[45] I.E. Livieris, S. Stavroyiannis, E.G. Pintelas, T. Kotsilieris, A dropout weight-constrained recurrent neural network model for forecasting the price of major cryptocurrencies and CCi30 index, Evol. Syst. (2021) http://dx.doi.org/10.1007/s12530-020-09361-2.
[46] R. Luo, W. Zhang, X. Xu, J. Wang, A neural stochastic volatility model, in: Proceedings of the AAAI Conference on Artificial Intelligence, 32 (1), 2018, pp. 6401–6408.
[47] C.G. Lamoureux, W.D. Lastrapes, Persistence in variance, structural change, and the GARCH model, J. Bus. Econom. Statist. 8 (2) (1990) 225–234.
[48] L. Bauwens, A. Dufays, J.V.K. Rombouts, Marginal likelihood for Markov-switching and change-point GARCH models, J. Econom. 178 (P39) (2014) 508–522.
[49] A.F. Bariviera, The inefficiency of Bitcoin revisited: A dynamic approach, Econ. Lett. 161 (C) (2017) 1–4, http://dx.doi.org/10.1016/j.econlet.2017.09.013.
[50] K. Balcombe, I. Fraser, Do bubbles have an explosive signature in Markov switching models? Econ. Model. 66 (C) (2017) 81–100, http://dx.doi.org/10.1016/j.econmod.2017.06.001.
[51] T.K. Siu, R.J. Elliott, Bitcoin option pricing with a SETAR-GARCH model, Eur. J. Finance 27 (6) (2021) 564–595.
[52] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[53] J.B. Heaton, N.G. Polson, J.H.J.H. Witte, Deep learning for finance: deep portfolios, Appl. Stoch. Models Bus. Ind. 33 (1) (2017) 3–12.
[54] R. Culkin, R.D. Sanjiv, Machine learning in finance: the case of deep learning for option pricing, J. Invest. Manage. 15 (2017) 92–100.
[55] J. Sirignano, R. Cont, Universal features of price formation in financial markets: perspectives from deep learning, Quant. Finance 19 (9) (2019) 1449–1459.
[56] C.L. Giles, S. Lawrence, A.C. Tsoi, Rule inference for financial prediction using recurrent neural networks, domain knowledge, in: Proceedings of the IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering (CIFEr), 1997, pp. 253–259.
[57] A.K. Rout, P.K. Dash, R. Dash, R. Bisoi, Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach, J. King Saud Univ. - Comput. Inf. Sci. 29 (4) (2017) 536–552.
[58] M.T. Hagan, M.B. Menhaj, Training feed forward networks with the Marquardt algorithm, IEEE Trans. Neural Netw. 5 (6) (1994) 989–993.
[59] L.G.B. Ruiz, M.P.M.P. Cuéllar, M.D. Calvo-Flores, M.D.C.P. Jiménez, An application of non-linear autoregressive neural networks to predict energy consumption in public buildings, Energies 9 (2016) 684.
[60] M.I. Jordan, Serial order: A parallel distributed processing approach, Adv. Psychol. 121 (1986) 471–495, http://dx.doi.org/10.1016/S0166-4115(97)80111-2.
[61] H. Tong, K.S. Lim, Threshold autoregression, limit cycles and cyclical data, J. R. Stat. Soc. Ser. B Stat. Methodol. 42 (3) (1980) 245–292.
[62] H. Tong, Threshold models in time series analysis - 30 years on, Stat. Interface 4 (2011) 107–118.
[63] R.F. Engle, Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation, Econometrica 50 (4) (1982) 987–1007.
[64] T.T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, J. Econometrics 31 (1986) 307–327.
[65] S. Hwang, S. Satchell, Market risk and the concept of fundamental volatility: Measuring volatility across asset and derivative markets and testing for the impacts of derivatives markets on financial markets, J. Bank. Financ. 24 (2000) 759–785.
[66] A.R. Admati, P. Pfleiderer, A theory of intraday patterns. Volume and price variability, Rev. Financ. Stud. 1 (1) (1988) 3–40.
[67] M. Parkinson, The extreme value method for estimating the variance of the rate of return, J. Bus. 53 (1) (1980) 61–65, http://dx.doi.org/10.1086/296071.
[68] M.B. Garman, M.J. Klass, On the estimation of security price volatilities from historical data, J. Bus. 53 (1) (1980) 67–78, https://www.jstor.org/stable/2352358.
[69] L. Chan, D. Lien, Using high, low, open, and closing prices to estimate the effects of ash settlement on futures prices, Int. Rev. Financ. Anal. 12 (1) (2003) 35–47.
[70] L.C.G. Rogers, S.E. Satchell, Y. Yoon, Estimating the volatility of stock prices: a comparison of methods that use high and low prices, Appl. Financial Econ. 4 (3) (1994) 241–247.
[71] D. Yang, Q. Zhang, Drift-independent volatility estimation based on high, low, open, and close prices, J. Bus. 73 (3) (2000) 477–491.
[72] S. Alizadeh, M.W. Brandt, F.X. Diebold, Range-based estimator of stochastic volatility models, J. Finance 57 (3) (2002) 1047–1091, http://dx.doi.org/10.1111/1540-6261.00454.
[73] A.R. Gallant, C.T. Hsu, G.G. Tauchen, Using daily range data to calibrate volatility diffusion and extract the forward integrated variance, Rev. Econ. Stat. 81 (4) (1999) 617–631.

[74] C. Bergmeir, J.M. Benitez, Neural networks in R using the stuttgart neural network simulator: RSNNS, J. Stat. Softw. 46 (7) (2012) 1:26.
[75] Peter Molnar, Properties of range-based volatility estimators, Int. Rev. Fin. Anal. 23 (2012) 20–29.
[76] M.W. Brandt, F.X.F.X. Diebold, A no-arbitrage approach to range based estimation of return covariances and correlations, J. Bus. 79 (2006) 61–74.
[77] M.W. Brandt, Ch.S.Ch.S. Jones, Volatility forecasting with range-based EGARCH models, J. Bus. Econom. Statist. 24 (4) (2006) 470–486.
[78] R.Y.R. Chou, Forecasting financial volatilities with extreme values: the conditional autoregressive range (CARR) model, J. Money Credit Bank. 37 (3) (2005) 561–582.
[79] R.Y. Chou, Modeling the asymmetry of stock movements using price ranges, Adv. Econom. 20 (2006) 231–258.
[80] R.Y. Chou, N.N. Liu, The economic value of volatility timing using a range-based volatility model, J. Econ. Dyn. Control 34 (2010) 2288–2301.