Applied Soft Computing Journal 151 (2024) 111132

Contents lists available at ScienceDirect

Applied Soft Computing

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

Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study

Grzegorz Dudek [a,*], Piotr Fiszeder [b,c], Paweł Kobus [d], Witold Orzeszko [b]

[a] Department of Electrical Engineering, Czestochowa University of Technology, Czestochowa, Poland
[b] Faculty of Economic Sciences and Management, Nicolaus Copernicus University in Torun, Torun, Poland
[c] Faculty of Finance and Accounting, Prague University of Economics and Business, Prague, Czech Republic
[d] Warsaw University of Life Sciences, Warsaw, Poland

HIGHLIGHTS

- Comprehensive study of the forecasting methods for cryptocurrency volatility.
- 12 popular methods compared including HAR, GARCH, LASSO, SVR, MLP, RF, LSTM.
- No single best method for each cryptocurrency.
- Different models perform better depending on error metric and forecast horizon.
- Simple linear models can perform as well as more complex ML models.

ARTICLE INFO

Keywords:
Machine learning
Cryptocurrency
Bitcoin
Volatility
Neural network
GARCH
HAR
LASSO
SVR
LSTM

ABSTRACT

Forecasting cryptocurrency volatility can help investors make better-informed investment decisions in order to minimize risks and maximize potential profits. Accurate forecasting of cryptocurrency price fluctuations is crucial for effective portfolio management and contributes to the stability of the financial system by identifying potential threats and developing risk management strategies. The objective of this paper is to provide a comprehensive study of statistical and machine learning methods for predicting daily and weekly volatility of the following four cryptocurrencies: Bitcoin, Ethereum, Litecoin, and Monero. Several models and forecasting methods are compared in terms of their forecasting accuracy, *i.e.*, HAR (heterogeneous autoregressive), ARFIMA (autoregressive fractionally integrated moving average), GARCH (generalized autoregressive conditional heteroscedasticity), LASSO (least absolute shrinkage and selection operator), RR (ridge regression), SVR (support vector regression), MLP (multilayer perceptron), FNM (fuzzy neighbourhood model), RF (random forest), and LSTM (long short-term memory). The realized variance calculated from intraday returns is used as the input variable for the models. In order to assess the predictive power of the models considered, the model confidence set (MCS) procedure is applied. Our experimental results demonstrate that there is no single best method for forecasting volatility of each cryptocurrency, and different models may perform better depending on the specific cryptocurrency, choice of the error metric and forecast horizon. For daily forecasts, the method that is always found in a set of best models is linear SVR, while for weekly forecasts, there are two such methods, namely FNM and RR. Furthermore, we show that simple linear models such as HAR and ridge regression, perform not worse than more complex models like LSTM and RF. The research provides a useful reference point for the development of more sophisticated models.

1. Introduction

Forecasting volatility of financial time series has garnered significant attention both from researchers and practitioners, including investors, risk managers, and policymakers. The relevance of this issue is due to the wide range of its practical applications in investment process, including risk management, portfolio optimization, and option pricing. However, forecasting volatility of financial series, due to its specific properties, is

* Corresponding author.
E-mail address: grzegorz.dudek@pcz.pl (G. Dudek).

https://doi.org/10.1016/j.asoc.2023.111132
Received 4 March 2023; Received in revised form 2 November 2023; Accepted 3 December 2023
Available online 7 December 2023

not a trivial task. Such features as a very large share of noise in price fluctuations, the conditional heteroscedasticity of returns, the significant influence of market microstructure, a large number of outliers, the presence of different time scales of investors cause that even more advanced methods have difficulties in achieving a high forecasting accuracy. An even more difficult task is to forecast cryptocurrency volatility. Cryptocurrencies came into existence as a digital payment medium but their volatility is completely different than the volatility of fiat money.

Cryptocurrencies are highly volatile in comparison to traditional currencies (*e.g.*, [11]), experience explosive and bubble behaviours in multiple periods [15,24,29,37], contain periods of very huge volatility (see [22]) and exhibit speculative behaviour [10,12]. These properties are caused by various reasons. Unlike traditional currencies, cryptocurrencies are not issued by central banks and, therefore, they are not attached to a specific country's economy. Their value is not based on any tangible asset. Periods of huge volatility are often connected with cybercrimes, hacks, unsuccessful fork attempts and regulatory disorientation [18,22]. The intrinsic speculative characteristics of the investments, the velocity of transactions, and the unregulated environment are also important causes of higher volatility of cryptocurrencies. Urquhart, Lucey [79] mention nine great challenges which need to be addressed before electronic currencies can become mainstream. One of them is how to manage the huge volatility of cryptocurrencies. In order to efficiently manage volatility, we must have tools that predict this volatility well. In this paper, we address this problem applying several methods which belong to the superior forecasting methods for other financial assets.

Since volatility is an unobservable variable, it has to be estimated. Before high frequency data becomes widely available, daily squared return, calculated from daily closing prices (C_t) as:

$$r_t^2 = (\ln(C_t/C_{t-1}))^2, \tag{1}$$

has been commonly used as a proxy of daily variance (see *e.g.*, [68]). Andersen, Bollerslev [4] showed that although the squared daily return is an unbiased estimator of the variance of return, it is also extremely noisy. A significantly more accurate measure of volatility is the realized variance (RV) calculated from intraday prices:

$$RV_{d,t} = \sum\nolimits_{k=1}^{K} r_{k,t}^2, \tag{2}$$

where $r_{k,t}$ is the intraday return, K is the number of intraday observations during a day.

The most popular class of models which is used to forecast volatility are GARCH-type (generalized autoregressive conditional heteroscedasticity) models. These models use daily squared return as the volatility estimator. However, intraday data contain more information about volatility than daily data, therefore the application of the realized variance provides much better forecasting ability than the use of daily squared return (see *e.g.*, [5,6,53]). In particular, many studies have shown that HAR (heterogeneous autoregressive) and ARFIMA (autoregressive fractionally integrated moving average) models, which use the realized variance to estimate volatility offer better forecasting ability than GARCH-type models (see *e.g.*, [13,17,53,67]). For these reasons, we analyse the realized variance in our research and this approach distinguishes our study from many others in the literature which are based on daily closing prices.

Standard volatility models like GARCH do not fit well to the cryptocurrency time series. For example, Charles, Darné [19] analysed Bitcoin prices and found that GARCH-type models are rejected because either the parameters are not statistically significant or the stationarity condition is not satisfied. Moreover, the results on the filtered returns showed that such models are not able to cope with jumps in volatility. Therefore, more advanced methods like machine learning (ML) methods have been proposed as an alternative. The importance of such algorithms in financial time series forecasting has increased considerably in the last several years [30,33,64,75]. ML models offer an advantage over statistical parametric models as they do not rely on prior assumptions about the data's underlying structure and have the ability to capture nonlinear patterns in time series. These models are flexible and adaptable, improving their performance through a training process. To further enhance their effectiveness, ML models can be complemented with time series decomposition methods and sophisticated optimization methods. Additionally, ensembling techniques are employed to aggregate model predictions, resulting in improved forecasting accuracy (*e.g.*, [26,35]). ML methods have been applied for cryptocurrencies in many applications such as, for instance, price prediction, volatility prediction, automated trading, mining, anonymity and privacy, fraud detection, and security (see [73]). Given the scope of our research, our focus is limited to volatility studies that specifically compare the forecasting accuracy of different methods.

The research with the application of ML methods to analyse volatility of cryptocurrencies can be divided into two main parts. The first one use daily squared return as the volatility estimator, and the second one – the realized variance. Papers from this first group, which is more numerous, are: Kristjanpoller, Minutolo [54] – hybrid GARCH models with multilayer perceptron (MLP) and separate GARCH-type models for Bitcoin (BTC), Peng, Albuquerque, de Sá, Padula, Montenegro [66] – hybrid GARCH models with Support Vector Regression (SVR), and separate GARCH-type models for BTC, Dash, Ethereum (ETH), Khaldi, El Afia, Chiheb [51] – MLP, Elman neural network (ENN)[1], GARCH-type models for BTC, Alqaralleh, Abuhommous, Alsaraireh [2] – MLP, GARCH-type models for BTC, Dash, Litecoin (LTC), Monero (XMR), Ripple (XRP), Seo, Kim [74] – hybrid GARCH models with MLP[2] and separate GARCH-type models for BTC, Aras [7] – meta-learning method based on GARCH-type models with SVR, MLP, random forest (RF), and separate GARCH-type models for BTC, Aras [8] – hybrid GARCH models with MLP, SVR, RF, K-nearest neighbours algorithm, and separate GARCH-type models for BTC, Shen, Wan, Leatham [76] – gated recurrent unit (GRU) with MLP, GARCH-type models, exponential weighted moving average (EWMA) for BTC, D'Amato, Levantesi, Piscopo [25] – nonlinear autoregressive neural network, Jordan neural network, self-exciting threshold autoregressive model (SETAR) for BTC, ETH, XRP, Lu, Liu, Lai, Cui [60] – long short-term memory (LSTM), GARCH for BTC, Zahid, Iqbal, Koutmos [84] – hybrid GARCH models with LSTM, GRU and Bidirectional LSTM and separate GARCH-type models for BTC, Amirshahi, Lahmiri [3] – hybrid GARCH models with feed forward neural networks (DFFNNs) and LSTM and separate GARCH, DFFNNs, LSTM models for selected 27 cryptocurrencies, Khan, Khan, Shaikh [52] - neural network autoregressive, cubic smoothing spline, group method of data handling neural network for BTC, ETH, XRP and Tether, Pratas, Ramos, Rubio [69] - MLP, recurrent neural networks, LSTM, GARCH models for BTC.

The second group of applications, which is less numerous, is based on realized variances. It includes among others: Guo, Bifet, Antulov-Fantulin [42] – RF, extreme gradient boosting (XGT), elastic-net, Gaussian process, LSTM, EWMA, GARCH-type models, structural time series model, autoregressive integrated moving average (ARIMA), temporal mixture models for BTC, Miura, Pichl, Kaizoji [61] – MLP, GRU, LSTM, convolutional neural network (CNN), SVR, ridge regression (RR), HAR for BTC, Gkillas, Tantoula, Tzagarakis [38] – hybrid HAR model with RF for BTC, Lehrer, Xie, Yi [58] – LASSO, regression tree, boosting tree, bagging tree, RF, SVR, least squares SVR, HAR for BTC, Li [59] – RF, HAR for BTC), Qiu [70] – complete subset regression (CSR), Mallows model averaging estimator (MMA), least absolute shrinkage and selection operator (LASSO), least squares support vector regressions (LSSVR), complete subset least squares support vector regressions (LSSVR^{CS}),

[1] MLP and ENN were based on realized variance but this measure was calculated from daily returns.

[2] MLP methods were based on realized variance but this measure was calculated from daily returns.

HAR for BTC, Rodikov, Antulov-Fantulin [72] – LSTM, HAR, GARCH-type models, EWMA, ARIMA for BTC, ETH, García-Medina, Aguayo-Moreno [34] – MLP, LSTM, hybrid GARCH models with LSTM, separate GARCH models for BTC, ETH, LTC, Bitcoin Cash, Tether, Eos, Binance Coin, Bitcoin SV, Stellar, Tron.

In the above studies, volatility forecasts based on ML methods were usually more accurate than forecasts derived from traditional parametric models like GARCH-type models, EWMA or ARIMA. Moreover, the selected ML methods often performed better than the HAR model. It is also worth emphasizing that hybrid models were superior to individual methods. Due to the limited number of forecasting methods used in each study and the differences between conducted research in terms of investigated time series, prediction horizons and the structure of regressors in the models, the formulated results are largely incomparable and can only provide a limited basis for making general conclusions about the usefulness of available methods for predicting the volatility of cryptocurrencies. The lack of clear guidance in the literature on the usefulness of existing forecasting methods raises the need for extensive comparative research in this area. The present work is part of such research, as its objective is to provide a comprehensive study of statistical and machine learning methods for predicting daily and weekly volatility of the most popular cryptocurrencies.

This study has four main contributions. Firstly, we compare a large group of statistical and ML methods for prediction of cryptocurrencies volatility. We use the following methods: HAR, HAR with robust estimation (HAR-R), ARFIMA, GARCH, LASSO, ridge regression, linear SVR, radial basis function SVR, MLP, the fuzzy neighbourhood model (FNM), RF and LSTM. Most of the previous studies were performed only for Bitcoin. On the other hand, in analyses completed for higher number of cryptocurrencies [2,3,25,34,66]) only a small number of ML methods was used. According to our knowledge, our study is the most comprehensive in terms of the number of methods applied to several cryptocurrencies. Christensen, Siggaard, Veliyev [21] used a broad selection of the most widely applied ML methods for the stocks of the Dow Jones Industrial Average index. We apply even broader group of methods, including twelve popular statistical and ML methods, to forecast daily and weekly volatility of Bitcoin, Ethereum, Litecoin and Monero.

Secondly, we concentrate on the models where volatility is estimated using realized variance, calculated from intraday prices. We compare this approach with the most popular GARCH model which uses daily squared return as the volatility estimator. We demonstrate that forecasts obtained from the GARCH model are poor for cryptocurrencies and this model is inferior to models based on the realized variance.

Thirdly, we show that there is no single best method for forecasting volatility of each cryptocurrency as our statistical analysis always identified multiple methods within the sets of best models. For daily forecasts the method that is always found in such a set of models based on the MSE and MAE measures is SVR, while for weekly forecasts there are two such methods, namely FNM and RR.

Fourthly, we demonstrate that forecasts based on simple methods, such as HAR or RR, are not significantly worse than forecasts obtained from some more advanced and complex methods like LSTM or RF.

The rest of the paper is organized in the following way. In Section 2 we describe the models and methods considered. Section 3 presents the analysed data and shows its summary statistics. In Section 4 we depict the forecasting procedure, evaluate the performance of the methods adopted for daily and weekly forecasts and discuss the results. The last section concludes.

2. Applied methods

The aim of this paper is to compare popular statistical and ML methods in forecasting volatility of cryptocurrency returns. We concentrate on the realized variance as the volatility estimator, for the reason that it is a significantly more accurate measure of volatility than daily squared returns [4]. We apply the realized variance as a proxy of ex-post volatility in the process of forecasts evaluation. Moreover, in all the considered models, except GARCH – which is based on daily returns – the realized variance is also the input variable for the calculation of volatility forecasts.

We include the weekly and monthly average realized variances as additional input variables in the models that allow for supplementary inputs, alongside the daily realized variance. Previous research, such as Corsi [23], has demonstrated the relevance of these variables in modelling and forecasting financial market volatility. The weekly and monthly average realized variances can be described, respectively, as:

$$RV_{w,t} = \frac{RV_{d,t-6} + RV_{d,t-5} + \ldots + RV_{d,t}}{7}, \tag{3}$$

$$RV_{m,t} = \frac{RV_{d,t-29} + RV_{d,t-28} + \ldots + RV_{d,t}}{30}. \tag{4}$$

We use 7 and 30 days averages because the cryptocurrency market is open every day of the week and month.

In addition to using $RV_{d,t}$, $RV_{w,t}$ and $RV_{m,t}$ as input variables, we also explored alternative approaches such as using 30 lags of $RV_{d,t}$. However, these alternative approaches did not improve the accuracy of our forecasts, and they increased the computational time. In order to ensure positive forecasts of volatility, we apply a logarithmic transformation to all realized variances. Thus, the models based on daily, weekly and monthly average realized variances can be generally expressed as $\ln RV_{d,t} = f(\ln RV_{d,t-1}, \ln RV_{w,t-1}, \ln RV_{m,t-1})$, as illustrated in Fig. 1. The forecasting procedure, utilizing a 23-month rolling window, is explained in detail in Section 4.1.

We forecast daily and weekly volatility. For this purpose, we utilize ten types of models (in twelve variants) with diverse properties and characteristics. The first three models, *i.e.*, HAR, ARFIMA, and GARCH, are classical statistical autoregressive models commonly employed in finance and economics. They primarily capture linear relationships in the data. HAR is specifically designed for forecasting realized volatility, while ARFIMA has a broader range of applications and can model both short-term and long-term dependencies. GARCH models are developed specifically for volatility modelling and forecasting, taking into account the conditional heteroscedasticity in financial time series. LASSO and RR are general-purpose linear models equipped with mechanisms to reduce model variance and enhance generalization. They employ different regularization terms.

The remaining five models are nonlinear ML models. SVR is a powerful technique for handling nonlinear regression problems by finding an optimal hyperplane in a higher-dimensional space using

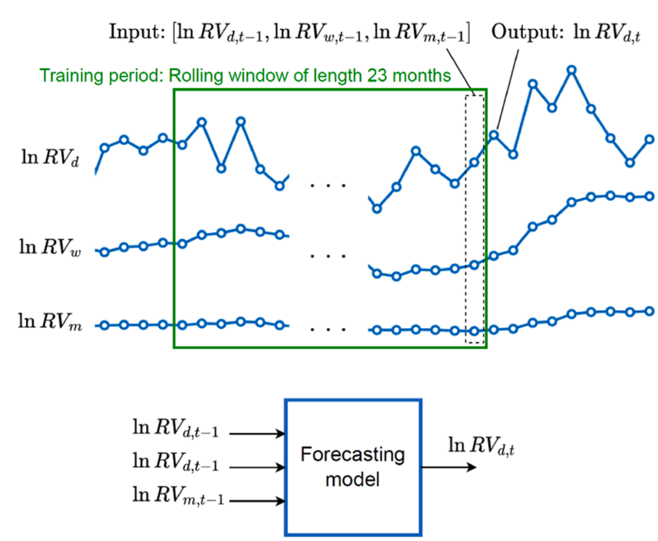

Fig. 1. The input and output data used in the forecasting models.

kernel functions. MLP constructs the regression function by combining activation functions of hidden nodes through linear combination. These functions are learned to achieve the best fit for the model. FNM, on the other hand, is a non-parametric regression model that combines training targets with weights expressing similarity between the current input pattern and the training patterns. RF is an ensemble learning method that aggregates multiple decision trees to make predictions. Each decision tree in the forest produces slightly different regression function based on different splits of the data. The trees are grown in a hierarchical manner and can be linearized into interpretable decision rules. Lastly, LSTM is a sophisticated and advanced model belonging to the family of recurrent neural networks. It excels at capturing both short-term and long-term dependencies in sequential data by utilizing internal states to store information over extended periods and by employing a gating mechanism for processing temporal information.

These ten types of models encompass a range of methodologies, allowing us to leverage their respective strengths and capabilities for volatility forecasting tasks. Below we present a short description of all the models.

2.1. Heterogeneous autoregressive model (HAR)

The HAR model of realized volatility was introduced by Corsi [23]. The main idea behind this model is to use different volatility components each of which is generated by actions of different types of market participants. For this purpose, the model combines volatility measures over different time horizons. Despite of a simple AR-type structure the HAR model is able to capture key properties of volatility of financial assets like heteroscedasticity, long memory and heterogeneity. We use the model with the log transformation of realized variances:

$$\ln RV_{d,t} = \gamma_0 + \gamma_1 \ln RV_{d,t-1} + \gamma_2 \ln RV_{w,t-1} + \gamma_3 \ln RV_{m,t-1} + \varepsilon_t. \tag{5}$$

This specification guarantees positive forecasts of volatility and additionally reduces the impact of outliers on the estimation results. We use two methods for estimation of parameters in the HAR model, namely ordinary least squares and robust estimation. In the latter case, the model is marked as HAR-R. For forecasts of weekly volatility, the daily forecasts are calculated recursively for each day of the week.

2.2. Autoregressive fractionally integrated moving average model (ARFIMA)

The ARFIMA model was proposed by Granger, Joyeux [39] and Hosking [49]. It generalizes the ARIMA model by allowing a non-integer value of the difference parameter. The ARFIMA model is useful in modelling time series with long memory. Such a property possesses the realized volatility that is why this model is applied for volatility modelling of financial time series. The ARFIMA(p, d, q) model for realized variance can be presented as:

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1-B)^d \ln RV_t = \left(1 + \sum_{j=1}^{q} \theta_j B^j\right)\varepsilon_t, \tag{6}$$

where d is the difference parameter, B denotes the lag (or backshift) operator ($B^s x_t = x_{t-s}$), ε_t is the innovation process, the fractional difference operator $(1-B)^d$ is defined in the following way:

$$(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k. \tag{7}$$

The ARFIMA process described by formula (6) is stationary when $d < 0.5$ and all roots of equation $\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)$ lie outside the unit circle. If $d \in (0, 0.5)$, then the ARFIMA process is called a long memory process. For estimation of parameters, we use the maximum likelihood method and apply the model with lags one, *i.e.*, the ARFIMA$(1,d,1)$. To calculate weekly forecasts, we use a similar approach to the HAR model.

2.3. Generalized autoregressive conditional heteroscedasticity model (GARCH)

The GARCH model of Bollerslev [14] is the most popular univariate volatility model. In contrary to other methods based on realized variances, it is formulated on returns. The GARCH(p, q) model can be presented as:

$$\varepsilon_t | \psi_{t-1} \sim N(0, h_t), \tag{8}$$

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}, \tag{9}$$

where ε_t is the innovation process from the conditional mean equation of returns, ψ_{t-1} is the set of all information available at time $t-1$, N is the conditional normal distribution and h_t is the conditional variance.

The standard restrictions for non-negativity of the conditional variance are $\alpha_0 > 0, \alpha_i \geq 0, \beta_j \geq 0$ (for $i = 1,2,\ldots,q$; $j = 1,2,\ldots,p$), however, weaker conditions can also be assumed (see [62]). For covariance stationarity, the following condition has to be satisfied $\alpha_1 + \ldots + \alpha_q + \beta_1 + \ldots + \beta_p < 1$. We use the maximum likelihood method for estimation of parameters. We apply the GARCH$(1,1)$ model with lags one which is the most frequently used in empirical studies. The weekly forecasts are determined in the same way as in the HAR and AFRIMA models.

2.4. Ridge regression (RR) and least absolute shrinkage and selection operator (LASSO)

In order to estimate the parameters of the HAR model for the dependent variable $y_t = \ln RV_{d,t}$ (see Eq. (5)), we additionally applied the following two shrinking methods: RR and LASSO. In general, when the relationship between variables is close to linear, the least squares estimates will have low bias but may have high variance. This means that a small change in the training data can cause a large change in the least squares coefficient estimates [50]. As has been pointed out in the literature, shrinking the coefficient estimates towards zero can considerably reduce their variance [45,55]. The two best-known shrinking methods are RR [48] and LASSO [78]. Both of these methods control the magnitude of the coefficient estimates by adding a penalty to the sum of the residual sum of squares $RSS = \sum_{t=1}^{N} (y_t - \hat{y}_t)^2$, if the estimates become large. This penalty introduces a trade-off between the model variance and bias. By sacrificing some bias, one can often reduce the variance in order to increase the prediction accuracy of unbiased models [55].

The RR coefficients minimize a penalized residual sum of squares:

$$RSS_{L_2} = \sum_{t=1}^{N} (y_t - \hat{y}_t)^2 + \lambda \sum_{j=1}^{P} \beta_j^2, \tag{10}$$

where β_j ($j = 1, 2, \ldots, P$) are the coefficients of the model. The RR solutions can be calculated directly from the formula:

$$\hat{\beta}^{ridge} = \left(X^T X + \lambda I\right)^{-1} X^T y. \tag{11}$$

A popular alternative to RR is LASSO. The LASSO estimate is defined by:

$$RSS_{L_1} = \sum_{t=1}^{N} (y_t - \hat{y}_t)^2 + \lambda \sum_{j=1}^{P} |\beta_j|. \tag{12}$$

It means that the L_2 penalty $\sum_{j=1}^{P} \beta_j^2$ in RR is replaced in LASSO by the L_1 penalty $\sum_{j=1}^{P} |\beta_j|$. This latter constraint makes the solutions nonlinear in the y_t and there is no closed form expression as in RR [45]. Moreover, the L_1 penalty leads to some parameters equal to zero, which means that LASSO yields models that simultaneously use regularization to improve the model and to conduct feature selection (*e.g.*, [50,55]).

In both methods, the tuning parameter $\lambda \geq 0$ controls the amount of shrinkage and is determined separately. We select this hyperparameter on a validation set. In order to calculate the weekly forecasts, we recursively predict volatility for each day of the week.

2.5. Support Vector Regression (SVR)

Support vector regression model [81] is based on the support vector machine method [80], which was originally introduced to solve classification problems. It combines the training efficiency and simplicity of linear algorithms with the prediction accuracy of the best nonlinear techniques [9,33]. It has been shown that SVR can be applied to high-dimensional or incomplete data and is robust to outliers [36,83]. Moreover, the computational complexity of SVR does not depend on the dimensionality of the input space [9].

The idea of SVR is to map the vectors of regressors $\mathbf{x}$ onto a high-dimensional feature space using some fixed (nonlinear) transformation and then to estimate the linear model

$$f(\mathbf{x}) = \sum_{j=1}^{m} \omega_j \varphi_j(\mathbf{x}) + b, \tag{13}$$

where m is the dimension of the space, $\varphi_j(\mathbf{x})$ denote transformations, ω_j are the coefficients and b is the bias term [20,30,57]. In order to estimate the coefficients of the SVR model the ε-insensitive loss function

$$L_\varepsilon(y, f(\mathbf{x})) = \begin{cases} 0, & |y - f(\mathbf{x})| \leq \varepsilon, \\ |y - f(\mathbf{x})| - \varepsilon, & \text{otherwise}, \end{cases} \tag{14}$$

has been proposed [80]. It means that training points $\{(\mathbf{x}_t, y_t)\}$ within the ε-margin have no loss, hence only points located outside the ε-margin are used as the support vectors to estimate the model. However, the accuracy of approximation (measured by the function L_ε) is not the only postulate taken into account in SVR. Besides it, SVR tries to reduce the model complexity by minimizing the formula $\|\boldsymbol{\omega}\|^2 = \boldsymbol{\omega}^T\boldsymbol{\omega}$, where $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_d)^T$. This optimization problem has the following solution:

$$f(\mathbf{x}) = \sum_{t=1}^{N_{SV}} (\alpha_t - \alpha_t^*) K(\mathbf{x}_t, \mathbf{x}), \text{ s.t. } 0 \leq \alpha_t \leq C,\ 0 \leq \alpha_t^* \leq C, \tag{15}$$

where α_t and α_t^* are the Lagrange multipliers, N_{SV} is the number of support vectors and K is the kernel function of the form (*cf.* [63]):

$$K(\mathbf{x}_t, \mathbf{x}) = \sum_{j=1}^{m} \varphi_j(\mathbf{x}) \varphi_j(\mathbf{x}_t). \tag{16}$$

Any function satisfying the Mercer's condition [80] can be used as the kernel. In our study we applied two kernel functions:

- Linear (dot product)

$$K(\mathbf{x}_t, \mathbf{x}) = \mathbf{x}_t{}^T \mathbf{x}, \tag{17}$$

- Radial basis function – RBF (Gaussian)

$$K(\mathbf{x}_t, \mathbf{x}) = \exp\left(-\gamma \|\mathbf{x}_t - \mathbf{x}\|^2\right). \tag{18}$$

In further sections, we denote the SVR model with the linear kernel by SVR-L and the model with the RBF kernel by SVR-G.

For each predicted value, we retrained SVR models with new values of hyperparameters ε, C (and γ in case of the SVR-G model) using the grid-search procedure. In order to calculate the weekly forecasts we apply the same procedure as in the case of the models described above.

2.6. Multilayer Perceptron (MLP)

MLP is a type of neural network, which is widely used for regression problems due to its valuable properties. They include universal approximation property, learning capability, nonlinear modelling, massive parallelism, robustness in the presence of noise, and fault tolerance.

Unlike the models described above, MLP produces forecasts for seven days ahead at once. Thus, it generates output vector $\widehat{\mathbf{y}}_t = \left[\ln\widehat{RV}_{d,t}, \ldots, \ln\widehat{RV}_{d,t+6}\right]$ based on input vector $\mathbf{x}_t = \left[\ln RV_{d,t-1},\ \ln RV_{w,t-1},\ \ln RV_{m,t-1}\right]$.

MLP architecture has three inputs, seven outputs and one nonlinear hidden layer. MLP is trained on historical data and fits a set of seven functions, $f_1(\mathbf{x})$, …, $f_7(\mathbf{x})$, which model forecasting relationships of different horizons. The function for the horizon $k = 1, \ldots, 7$ is expressed as follows:

$$f_k(\mathbf{x}) = \sum_{j=1}^{m} v_{j,k} \varphi_j(\mathbf{x}) + v_{0,k}, \tag{19}$$

where m represents the number of hidden nodes, $\varphi_j(\mathbf{x})$ is a j-th hidden node activation function:

$$\varphi_j(\mathbf{x}) = \tanh(\mathbf{x}) = \frac{2}{1 + \exp\left(-\left(\sum_{i=1}^{n} w_{i,j} x_i + w_{0,j}\right)\right)} - 1, \tag{20}$$

$n = 3$ is the number of inputs, w and v are weights of the hidden nodes and output nodes, respectively.

Approximation properties of MLP are dependent on the number of hidden nodes, m. We select this hyperparameter on a validation set. To prevent overfitting, we train MLP using the Levenberg–Marquardt algorithm with Bayesian regularization, which minimizes a combination of squared errors and weights.

2.7. Fuzzy Neighbourhood Model (FNM)

FNM is a nonparametric regression model, which takes into account all training patterns when constructing a regression surface [27]. The underlying assumption of FNM is as follows: similarity in space X implies similarity in space Y. It is assumed in this approach that the training pattern belongs to the neighbourhood of query pattern $\mathbf{x}$ with some degree of membership. The degree of membership is defined by a membership function which monotonically decreases with distance between patterns. The degree of memberships are treated as weights for training patterns in the regression function defined as follows:

$$f(\mathbf{x}) = \frac{\sum_{t=1}^{N} \mu(\mathbf{x}, \mathbf{x}_t) \mathbf{y}_t}{\sum_{t=1}^{N} \mu(\mathbf{x}, \mathbf{x}_t)}, \tag{21}$$

where $\mu(\mathbf{x}, \mathbf{x}_t)$ is a Gaussian-type function of membership of training pattern $\mathbf{x}_t$ to the neighbourhood of query pattern $\mathbf{x}$ of the form:

$$\mu(\mathbf{x}, \mathbf{x}_t) = \exp\left(-\left(\frac{\|\mathbf{x} - \mathbf{x}_t\|}{\sigma}\right)^2\right), \tag{22}$$

where $\|\|$ is a Euclidean norm and σ is a bandwidth parameter.

Regression function (21) combines all training output patterns $\mathbf{y}$ with weights dependent on the distance between their corresponding x-patterns to the query pattern. Note that it is a vector-valued function producing vector $\mathbf{y}$ as a result. This vector includes forecasted values for seven days ahead.

The only hyperparameter in FNM is bandwidth σ, deciding about the smoothness of the regression function. It is selected on the validation set. Too small values of σ result in undersmoothing, whereas too large values result in oversmoothing. We make this hyperparameter dependent on data, expressing it as a function of the distance between the query pattern and the training patterns as follows: $\sigma = ad$, where a is a coefficient to adjust, and d is the median of Euclidean distance between the training input patterns and the query pattern.

2.8. Random Forest (RF)

RF is an ensemble learning method based on decision trees as the base models [16]. For regression problem, we focus on the regression RF, which grows and aggregates multiple regression trees. The method combines bagging with a random subspace method to construct a collection of noisy but approximately unbiased base models and thus reduce the prediction variance. The main issue of the ensemble learning is to generate appropriate diversity in the base models. A random subspace method helps to increase diversity between trees by restricting them to work on different random subsets of the predictor space, while bagging generates additional diversity by building each tree in the forest from a bootstrap sample of the original dataset.

RF produces an scalar output based on input pattern $\mathbf{x}$, so for each forecast horizon $k = 1, \ldots, 7$, an individual RF should be built. RF for horizon k is expressed as [45]:

$$f_k(\mathbf{x}) = \frac{1}{p} \sum_{j=1}^{p} T_{k,j}(\mathbf{x}), \tag{23}$$

where p is the number of trees in a forest and $T_{k,j}(\mathbf{x})$ is a response of the j-th tree of the k-th forest on the query pattern $\mathbf{x}$.

A tree is characterized by several parameters: split predictors and cutpoints at each node, and terminal-node (leaf) values. These parameters are selected in the learning procedure based on the split criterion which is a mean square error for regression. The main RF hyperparameters are the number of trees in the forest, p, the minimum number of observations in a leaf (or equivalent), q, and the number of predictors to select at random for each split, r. All hyperparameters control the bias-variance tradeoff of the model. We select p and q on the validation set, while r we assume as $n/3$, as the RF inventors recommend.

2.9. Long Short-Term Memory (LSTM)

LSTM is a modern recurrent neural network with gating mechanism [47]. Unlike the other ML models used in this study, it was designed for sequential data and it is capable of learning both short and long-term relationships in time series [46]. LSTM is equipped with recurrent cells that can maintain their states over time. Nonlinear 'regulators' called gates can control the flow of information inside the cell and so adapt the cell state, $\mathbf{c}$, and the hidden state, $\mathbf{h}$, to the current dynamics of the process.

The LSTM network is composed of two layers: the LSTM layer and the linear layer. The LSTM layer approximates temporal dependencies in time series and produces state vectors. The linear layer transforms the hidden state vector $\mathbf{h}$ into the output vector $\mathbf{y}$. The model fits seven functions at once, $f_1(\mathbf{x}),\ldots,f_7(\mathbf{x})$, for each forecast horizon. The function for the horizon k is expressed as:

$$f_k(\mathbf{x}) = \mathbf{v}_k^T \mathbf{h}(\mathbf{x}) + v_{0,k}, \tag{24}$$

where $\mathbf{v}_k$ is an m-component weight vector of the k-th output node and $v_{0,k}$ is a node bias, m is the number of nodes in the LSTM gates, $\mathbf{h}(\mathbf{x})$ is a hidden state of LSTM:

$$\mathbf{h}(\mathbf{x}) = \mathrm{LSTM}(\mathbf{x}; \mathbf{W}) \tag{25}$$

and $\mathbf{W}$ are LSTM weights and biases.

To improve learning ability of LSTM, the input and output variables are defined differently than for other models: $\mathbf{x}_t = \left[\ln(1+100RV_{d,t-1}), \ln(1+100RV_{w,t-1}), \ln(1+100RV_{m,t-1})\right]$, $\mathbf{y}_t = \left[\ln(1+100RV_{d,t}), \ldots, \ln(1+100RV_{d,t+6})\right]$.

All weights and biases are learnable parameters. The most important hyperparameter, which we select on the validation set, is the number of nodes in each gate, m. It decides about the amount of information contained in the states. For more complex temporal relationships, more nodes are required.

3. Description of data

We apply the competing methods and models to four cryptocurrency rates: BTC/USD (Bitcoin), ETH/USD (Ethereum), LTC/USD (Litecoin), XMR/USD (Monero). When selecting assets, we are guided by two criteria. First, we choose the most heavily traded cryptocurrencies in order to limit the influence of market microstructure effects on volatility estimates. The second criterion is the length of the time series. This factor is important because the statistical evaluation requires an appropriate number of forecasts.

We analyse data received directly from the crypto exchange Kraken. Data from coin-ranking sites can be questionable due to non-traded prices, mistakes in time stamps, use of non-fiat cross-rates and wash trading (see [1]). The data start from January 1, 2017 and end on December 31, 2021. For each day we estimate the daily realized variance calculated as the sum of squares of intraday returns (Eq. (2)). Since cryptocurrencies are quoted 24 h a day and we take 5-min returns, the number of intraday observations used to calculate the realized variance is equal to 288. Fig. 2 presents the series of daily prices, returns, realized variances and logarithm of realized variances of all analysed cryptocurrencies. We calculate daily returns as $r_t = \ln(C_t/O_t)$, where C_t and O_t are daily closing and opening prices, respectively.

Huge price increases are visible for BTC/USD and ETH/USD. Moreover, two speculative bubbles are noticeable for prices of LTC/USD and XMR/USD. Large outliers are present both in returns and realized variances. The biggest daily losses for all cryptocurrencies took place on March 12, 2020 as fears of the economic damage from the coronavirus pandemic took hold.

Descriptive statistics of daily returns and realized variances are given in Table 1.

All mean returns are positive and relatively high. The highest gains could be obtained for ETH/USD. Volatility of BTC/USD is clearly lower than volatility of other cryptocurrencies. However, the standard deviation of returns of all analysed digital assets is considerably larger in comparison to other assets. It is almost ten times higher than the standard deviation of fiat currencies and about three times higher than the standard deviation of stocks and commodities (*cf.* [30,32]). All distributions are asymmetric, and display high kurtosis. These summary results suggest that forecasting volatility of cryptocurrencies is a pretty difficult task.

4. Experimental study

4.1. Forecasting procedure

In this section, we compare the forecasting performance of all the methods discussed in Section 2. To evaluate their effectiveness, we generate out-of-sample forecasts for both one-day ahead and one-week ahead volatility.

For the daily forecasts, we apply the following procedure, based on a rolling window approach. We start with a training sample spanning from February 1, 2017, to December 31, 2018 (data from January 2017 is not included in this range because it is used for RV_m and RV_d calculation). We optimize and train the ML models and estimate the parameters of the statistical models based on this training sample. Using these models, we

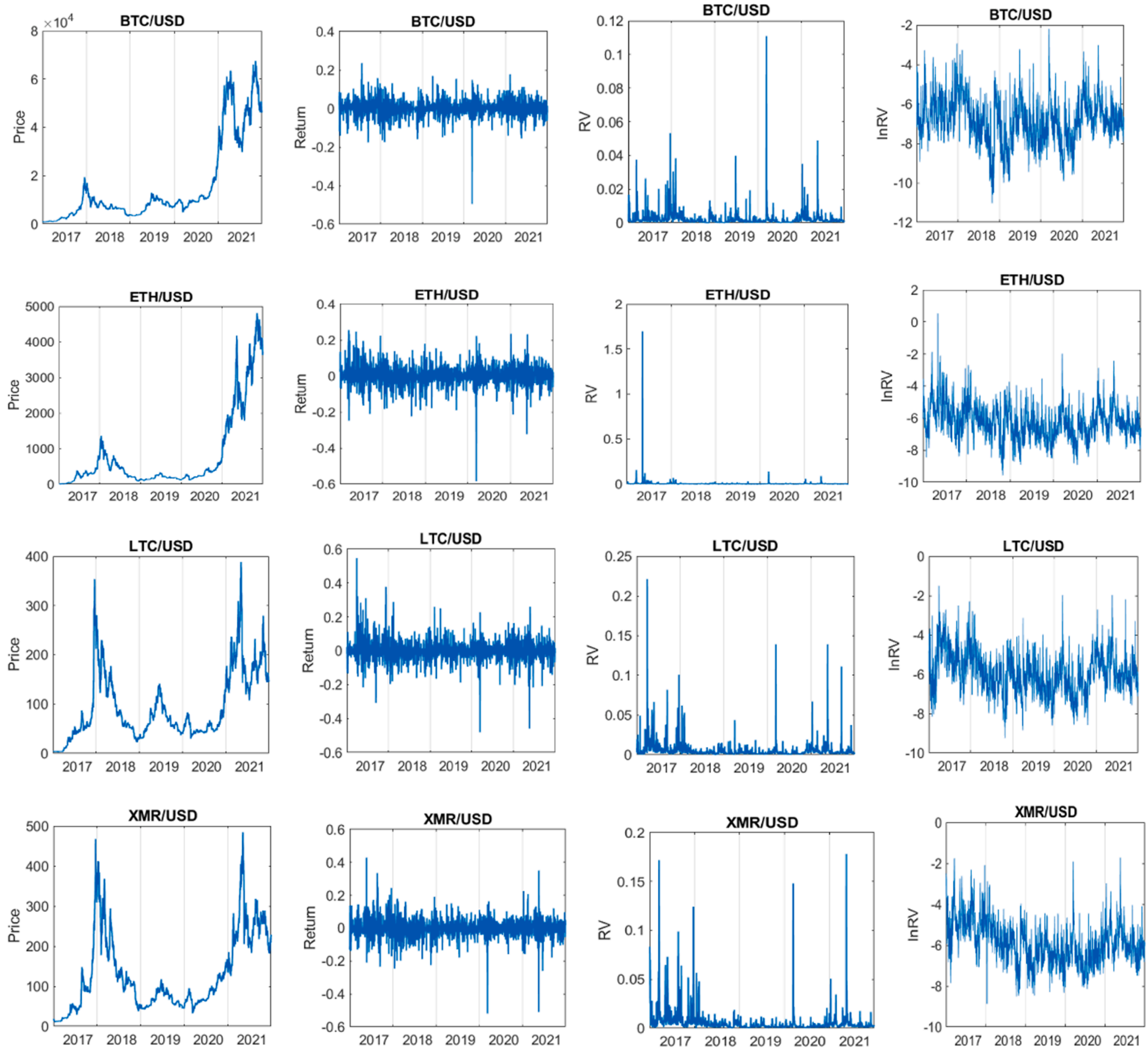

Fig. 2. Price, return, realized variance and logarithm of realized variance of cryptocurrencies.

Table 1
Summary statistics of daily returns and realized variances of analysed cryptocurrencies.

Rates	Mean	Minimum	Maximum	Standard deviation	Skewness	Excess kurtosis
			Daily returns			
BTC/USD	0.00218	-0.496	0.236	0.043	-0.850	14.756
ETH/USD	0.00332	-0.585	0.256	0.057	-0.528	11.764
LTC/USD	0.00181	-0.480	0.547	0.062	0.394	12.957
XMR/USD	0.00133	-0.520	0.428	0.060	-0.376	12.664
			Daily realized variances			
BTC/USD	0.00222	0.000	0.111	0.005	10.527	185.259
ETH/USD	0.00478	0.000	1.697	0.041	39.902	1661.728
LTC/USD	0.00521	0.000	0.221	0.012	9.310	136.481
XMR/USD	0.00549	0.000	0.178	0.011	7.914	91.421

Note: the realized variance is estimated as the sum of squared of 5-minute returns. The analysed period is January 1, 2017 - December 31, 2021.

1. **Input**: daily realized variances $RV_{d,t}$ from January 1, 2017 to December 31, 2021.

2. Calculation of weekly and monthly realized variances $RV_{w,t}$ and $RV_{m,t}$ (see Eqs. 3-4).

3. Logarithmic transformation: $\ln RV_{d,t}$, $\ln RV_{w,t}$, $\ln RV_{m,t}$.

4. Constructing a training sample $\{[\ln RV_{d,t-1}, \ln RV_{w,t-1}, \ln RV_{m,t-1}], \ln RV_{d,t}; t = 2, 3, \ldots, T\}$. In the first iteration the training sample spans February 1, 2017 to December 31, 2018.

5. Building predictive models: $\ln RV_{d,t} = f(\ln RV_{d,t-1}, \ln RV_{w,t-1}, \ln RV_{m,t-1})$. Based on the training sample, ML models are optimized and trained, and the parameters of the statistical models are estimated.

6. Generation of one day ahead forecast $ln\widehat{RV}_{d,T+1}$.

7. Calculation of the forecast $\widehat{RV}_{d,T+1} = \exp(ln\widehat{RV}_{d,T+1})$.

8. Updating a training sample according to a rolling window procedure, i.e. adding one new observation $\{[\ln RV_{d,T}, \ln RV_{w,T}, \ln RV_{m,T}], \ln RV_{d,T+1}\}$ to the training sample while removing the oldest observation.

9. Updating $T = T + 1$ and repeating steps 5-8 until forecasts for the whole period from January 1, 2019 to December 31, 2021 are generated.

10. **Output**: 1,096 forecasts of daily realized variances $\widehat{RV}_{d,t}$ for the period from January 1, 2019 to December 31, 2021.

Diag. 1. Procedure for generating one-day ahead forecasts.

make one-day ahead forecasts for January 1, 2019. Subsequently, to incorporate new data and maintain a rolling window, we add one new observation to the training sample while removing the oldest one. With the updated training sample, we retrain the ML models, re-estimate the parameters of the statistical models, and generate forecasts for January 2, 2019. This process is repeated until we obtain forecasts for the three-year period from January 1, 2019, to December 31, 2021. In total, this yields 1096 daily forecasts for each method. Forecasting procedure for generating one-day ahead forecasts is illustrated in Diagram 1.

For the weekly forecasts, we follow a similar procedure. However, we only make forecasts once a week on Sundays. We forecast the volatility for each day of the following week, from Monday to Sunday, and calculate the weekly forecast as the sum of seven daily forecasts. In total, this results in 155 weekly forecasts for each method.

The ML models were optimized using the training samples. From these samples the validation samples were selected to estimate generalization errors. Based on these errors, optimal values for the hyperparameters were found. The errors reported in the tables of this section are the average errors obtained over a three-year test period. By averaging the errors, the impact of random initialization, which occurs in some ML models, is effectively mitigated.

The evaluation of forecasts is performed on the basis of two basic measures, namely the mean squared error (MSE) and the mean absolute error (MAE). The MSE is the most frequently used criterion in forecasting studies. The MSE is robust to the use of a noisy volatility proxy (it yields the same ranking of competing forecasts using an unbiased volatility proxy, see [43] and [65]). The MAE is less sensitive to outliers, which is very important when evaluating extraordinary events.

In order to assess the predictive power of the models considered, the model confidence set (MCS) procedure of Hansen, Lunde, Nason [44] is applied. The objective of the MCS test is to identify the set of best models. Starting with the full set of models, the MCS procedure sequentially eliminates the models that are found to be significantly inferior until the null hypothesis of equal forecast accuracy is no longer rejected at the assumed significance level. Finally, MCS contains the best forecasting methods with a certain probability. We use this test for the twelve methods jointly.

4.2. Results for daily forecasts

In this section, we evaluate the daily volatility forecasts for all considered methods. The results are presented in Tables 2 and 3 for the MSE and MAE criteria, respectively.

Both for the MSE and MAE measures, the lowest errors of the forecasts are for BTC/USD, next for ETC/USD then for XMR/USD and LTC/USD. It corresponds with the level of the standard deviation of cryptocurrency returns (see Table 1). The values of RMSE are considerably higher than of MAE because the former measure is more sensitive to outliers.

Generally, according to the rankings of the models considered, the most accurate volatility forecasts are based on the SVR-L model (for the MSE measure) and on the ARFIMA model (for the MAE criterion). To assess formally the relative performance of the models, we apply the MCS procedure (see Tables 2 and 3). For all cryptocurrencies several methods belong to the model confidence set. It means that there is no one method that is significantly better than others. The only model that is included in the set of the best models for all cryptocurrencies and both evaluation measures is SVR-L (eight times). The SVR-G is contained in such a set seven times and the HAR-R and ARFIMA models six times. On the other hand, the methods that are least often included in the set of best models are LSTM (zero times), FNM, RF (once) and GARCH (twice). Most often the size of the model confidence set is bigger for MAE than for the MSE criterion.

Bergsli, Lind, Molnár, Polasik [13] demonstrated for BTC that the forecasts based on the GARCH models are less accurate than the forecasts obtained from the HAR model. We extend this result and show that the HAR model based on the realized variance performs most often better than the GARCH model also for other cryptocurrencies.

We consider two methods of estimation for the HAR model, namely the ordinary least squares and robust methods. The differences in evaluation measures between these two methods are very small and the application of the robust method does not increase significantly the forecasting accuracy in our analysis.

Although the set of best models contains many models, it is worth noting that the lowest values of the MAE measure for all cryptocurrencies are obtained from the ARFIMA model. We can conclude that modelling long memory of the realized variance by the ARFIMA model is beneficiary for volatility forecasts.

Table 2
Evaluation of daily volatility forecasts based on the MSE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	MSE	Rank	*p*-value	MSE	Rank	*p*-value	MSE	Rank	*p*-value	MSE	Rank	*p*-value
HAR	0.191	3	0.342 *	0.330	6	0.022	0.590	7	0.046	0.550	2	0.986 *
HAR-R	0.192	4	0.342 *	0.331	7	0.005	0.590	6	0.261 *	0.552	4	0.007
ARFIMA	0.195	6	0.014	0.329	5	0.736 *	0.587	5	0.363 *	0.555	6	0.007
GARCH	0.215	12	0.001	**0.315**	1	1.000 *	0.620	10	0.021	**0.542**	1	1.000 *
LASSO	0.200	9	0.014	0.350	9	0.005	**0.580**	1	1.000 *	0.556	7	0.000
RR	0.193	5	0.014	0.329	4	0.022	0.581	3	0.647 *	0.556	8	0.000
SVR-G	0.195	7	0.342 *	0.325	2	0.824 *	0.585	4	0.604 *	0.554	5	0.983 *
SVR-L	0.191	2	0.350 *	0.326	3	0.736 *	0.580	2	0.647 *	0.550	3	0.986 *
MLP	**0.186**	1	1.000 *	0.342	8	0.007	0.602	8	0.046	0.559	9	0.000
FNM	0.198	8	0.014	0.363	10	0.005	0.623	11	0.005	0.605	11	0.000
RF	0.206	10	0.014	0.363	11	0.005	0.613	9	0.023	0.602	10	0.000
LSTM	0.207	11	0.000	0.398	12	0.000	0.685	12	0.000	0.631	12	0.000

Note: The values of MSE are multiplied by 10^4, the lowest values of MSE are in bold, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

Table 3
Evaluation of daily volatility forecasts based on the MAE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	MAE	Rank	*p*-value	MAE	Rank	*p*-value	MAE	Rank	*p*-value	MAE	Rank	*p*-value
HAR	**0.102**	1	1.000 *	0.147	3	0.520 *	0.209	6	0.024	0.166	4	0.738 *
HAR-R	0.102	2	0.922 *	0.147	2	0.835 *	0.209	5	0.399 *	0.165	2	0.930 *
ARFIMA	0.102	4	0.922 *	**0.146**	1	1.000 *	**0.206**	1	1.000 *	**0.165**	1	1.000 *
GARCH	0.138	12	0.000	0.205	12	0.000	0.259	11	0.000	0.228	12	0.000
LASSO	0.107	10	0.001	0.157	10	0.000	0.208	2	0.399 *	0.167	7	0.461 *
RR	0.102	3	0.922 *	0.148	6	0.131 *	0.208	3	0.399 *	0.167	8	0.461 *
SVR-G	0.106	9	0.198 *	0.147	4	0.520 *	0.211	8	0.002	0.165	3	0.930 *
SVR-L	0.103	7	0.271 *	0.148	5	0.520 *	0.209	4	0.399 *	0.166	6	0.494 *
MLP	0.103	5	0.922 *	0.149	7	0.131 *	0.211	7	0.002	0.166	5	0.668 *
FNM	0.103	6	0.754 *	0.152	9	0.045	0.215	10	0.000	0.175	10	0.009
RF	0.106	8	0.023	0.152	8	0.131 *	0.213	9	0.000	0.175	9	0.009
LSTM	0.114	11	0.000	0.178	11	0.000	0.263	12	0.000	0.208	11	0.000

Note: The values of MAE are multiplied by 10^2, the lowest values of MAE are in bold, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

For the robustness check of our results, we apply three additional evaluation measures, namely, the coefficient of determination from the Mincer-Zarnowitz regression (R^2), the mean absolute percentage error (MAPE) and the quasi-likelihood loss function (QLIKE). These criteria are often used for an evaluation of volatility forecasts in empirical studies (see *e.g.,* [65,68,82]). The results of these measures are given in Tables A.1-A.3 in Appendix. It is worth paying attention to relatively low values of R^2 in comparison to fiat currencies, stocks and commodities (*cf.* [32,33]). These low values show how difficult it is to forecast cryptocurrency volatility. The conclusions drawn from the additional evaluation measures are more or less similar to the results obtained applying the MSE and MAE criteria. Moreover, the differences between the considered models depend on the adopted measure. They are small for the QLIKE, MSE, MAE values, noticeable for the MAPE values and meaningful for the R^2 values.

Additionally, we assess the effectiveness of different methods in forecasting volatility under both low and high volatility conditions, comparing their performance against competitors. For this reason, we calculate the MAE measure for 5% of the lowest and separately for 5% of the highest values of realized variances. The results of this evaluation are given in Tables 4 and 5. Due to the much smaller number of such observations, we no longer test the significance of these results.

Table 4
Evaluation of daily volatility forecasts based on the MAE measure for 5% of the lowest values of realized variances.

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	MAE	Rank	MAE	Rank	MAE	Rank	MAE	Rank
HAR	0.161	7	0.394	5	0.477	7	0.658	11
HAR-R	0.137	2	**0.340**	1	**0.406**	1	0.573	8
ARFIMA	0.158	5	0.375	4	0.465	6	0.544	5
GARCH	0.848	12	1.684	12	1.821	12	0.215	12
LASSO	0.228	10	0.578	11	0.488	8	0.593	9
RR	0.162	8	0.406	7	0.461	5	0.593	10
SVR-G	**0.136**	1	0.368	2	0.452	4	0.512	3
SVR-L	0.142	4	0.374	3	0.431	3	0.502	2
MLP	0.139	3	0.399	6	0.504	9	0.550	6
FNM	0.167	9	0.445	10	0.528	11	0.572	7
RF	0.159	6	0.415	9	0.509	10	0.527	4
LSTM	0.354	11	0.413	8	0.414	2	**0.283**	1

Note: the values of MAE are multiplied by 10^3, the lowest values of MAE are in bold. The evaluation period is January 1, 2019 - December 31, 2021.

Table 5
Evaluation of daily volatility forecasts based on the MAE measure for 5% of the highest values of realized variances.

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	MAE	Rank	MAE	Rank	MAE	Rank	MAE	Rank
HAR	1.025	4	1.361	6	1.820	5	1.552	2
HAR-R	1.033	6	1.369	7	1.823	6	1.571	8
ARFIMA	1.026	5	1.355	4	1.832	7	1.563	6
GARCH	1.117	12	1.354	3	1.928	10	1.579	9
LASSO	1.080	10	1.464	10	1.819	4	1.563	4
RR	1.012	3	1.360	5	1.819	3	1.563	5
SVR-G	1.045	7	**1.347**	1	**1.803**	1	**1.492**	1
SVR-L	1.001	2	1.353	2	1.812	2	1.560	3
MLP	**0.993**	1	1.390	8	1.858	8	1.567	7
FNM	1.048	8	1.441	9	1.908	9	1.751	11
RF	1.070	9	1.489	11	1.928	11	1.727	10
LSTM	1.100	11	1.654	12	2.193	12	1.899	12

Note: the values of MAE are multiplied by 10^2, the lowest values of MAE are in bold. The evaluation period is January 1, 2019 - December 31, 2021.

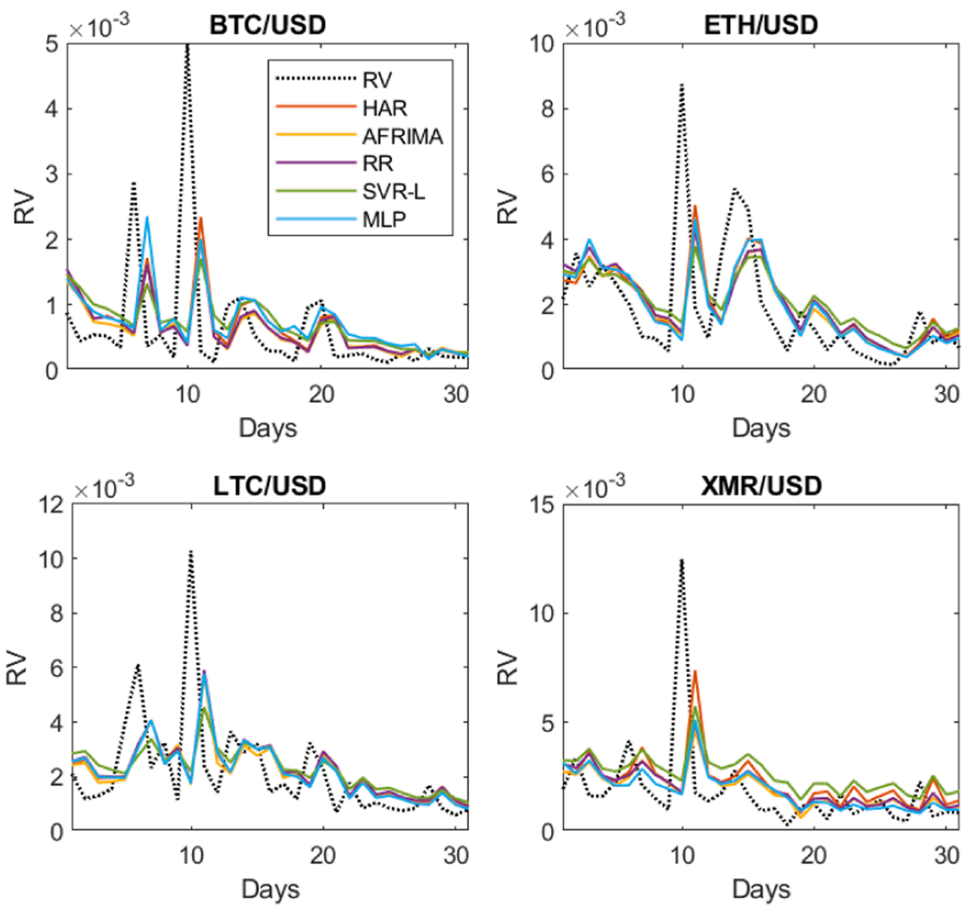

Fig. 3. Examples of forecasts of the daily volatility.

For low volatility, the most accurate forecast are obtained from the SVR-G, SVR-L and HAR-R models, and the least precise forecast are from GARCH, LASSO and FNM. SVR-L and SVR-G are again the best performing models when the realized variance is high. The least accurate forecasts of high volatility are based on the LSTM and RF methods.

Fig. 3 shows examples of the daily forecasts produced by selected models for January 2019. It is clear from the figure that none of the methods is able to forecast big spikes in volatility. Furthermore, a notable delay in the forecast timeline relative to the actual target series is evident.

4.3. Results for weekly forecasts

The weekly volatility forecasts are evaluated in this section. The results are presented in Tables 6 and 7 for the MSE and MAE criteria, respectively.

According to the MSE measure, BTC/USD has the lowest forecasting errors, next is ETC/USD then LTC/USD and XMR/USD. For the MAE criterion LTC/USD and XMR/USD switch places. Similarly like for daily forecasts, the values of RMSE are considerably higher than those of MAE because the former measure is more sensitive to outliers.

Table 6
Evaluation of weekly volatility forecasts based on the MSE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	MSE	Rank	*p*-value	MSE	Rank	*p*-value	MSE	Rank	*p*-value	MSE	Rank	*p*-value
HAR	0.398	9	0.000	0.768	8	0.149 *	1.115	9	0.015	1.185	9	0.298 *
HAR-R	0.405	10	0.000	0.784	9	0.000	1.137	10	0.000	1.195	10	0.048
ARFIMA	0.388	7	0.061	0.746	7	0.149 *	1.068	8	0.094	1.157	6	0.663 *
GARCH	0.391	8	0.573 *	0.819	11	0.013	1.163	11	0.094	1.491	12	0.002
LASSO	0.408	11	0.000	0.804	10	0.013	1.038	3	0.831 *	1.146	4	0.663 *
RR	0.378	2	0.582 *	0.720	2	0.480 *	**1.034**	1	1.000 *	**1.125**	1	1.000 *
SVR-G	0.380	4	0.582 *	**0.713**	1	1.000 *	1.042	4	0.408 *	1.139	3	0.663 *
SVR-L	0.380	3	0.582 *	0.723	3	0.480 *	1.037	2	0.831 *	1.134	2	0.663 *
MLP	**0.358**	1	1.000 *	0.743	6	0.149 *	1.057	7	0.408 *	1.153	5	0.663 *
FNM	0.383	6	0.171 *	0.735	5	0.168 *	1.055	6	0.408 *	1.159	7	0.587 *
RF	0.383	5	0.171 *	0.726	4	0.168 *	1.047	5	0.408 *	1.170	8	0.048
LSTM	0.430	12	0.000	0.922	12	0.000	1.479	12	0.000	1.408	11	0.000

Note: The values of MSE are multiplied by 10^3, the lowest values of MSE are in bold, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

Table 7
Evaluation of weekly volatility forecasts based on the MAE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	MAE	Rank	*p*-value	MAE	Rank	*p*-value	MAE	Rank	*p*-value	MAE	Rank	*p*-value
HAR	0.712	7	0.033	1.044	7	0.033	1.489	9	0.024	1.167	8	0.262 *
HAR-R	0.730	9	0.000	1.062	8	0.008	1.516	10	0.021	1.152	6	0.753 *
ARFIMA	0.703	5	0.625 *	1.025	5	0.585 *	1.438	6	0.082	**1.130**	1	1.000 *
GARCH	0.881	12	0.000	1.415	12	0.000	1.693	11	0.000	1.661	12	0.000
LASSO	0.750	10	0.000	1.114	10	0.000	1.411	2	0.872 *	1.147	2	0.840 *
RR	**0.688**	1	1.000 *	1.015	3	0.744 *	**1.406**	1	1.000 *	1.149	4	0.840 *
SVR-G	0.712	6	0.033	1.025	4	0.548 *	1.427	4	0.266 *	1.183	10	0.245 *
SVR-L	0.714	8	0.003	1.011	2	0.744 *	1.431	5	0.082	1.163	7	0.697 *
MLP	0.691	2	0.897 *	1.073	9	0.001	1.449	8	0.082	1.148	3	0.840 *
FNM	0.699	4	0.853 *	**1.001**	1	1.000 *	1.423	3	0.872 *	1.152	5	0.753 *
RF	0.697	3	0.897 *	1.031	6	0.494 *	1.441	7	0.082	1.169	9	0.316 *
LSTM	0.856	11	0.000	1.332	11	0.000	2.009	12	0.000	1.577	11	0.000

Note: The values of MAE are multiplied by 10^2, the lowest values of MAE are in bold, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

Generally, both for the MSE and MAE, the most accurate forecasts of volatility are from the RR model. In order to evaluate the statistical significance of the results, the MCS procedure is applied. For each cryptocurrency several methods belong to the model confidence set. Only two models, FNM and RR, are included in the set of best models for all cryptocurrencies and both evaluation measures (eight times). The SVR-G belongs is contained in such a set seven times and the SVR-L, MLP and RF models six times. The methods that are least often included in the set of best models are LSTM (zero times), HAR-R and GARCH (one time). For XMR/USD cryptocurrency rate most models belong to the set of best models.

Weekly forecasting errors are considerably higher than daily errors, for the MSE measure they are more than twenty times bigger and for the MAE criterion about seven times. Main conclusions from daily forecasts are also valid for weekly forecasts, however, as we pointed out, different models dominate in both cases.

In the robustness check, we consider three additional evaluation measures: R^2, MAPE, and QLIKE, the results of which are presented in Tables A.4-A.6 in the Appendix. Generally, the findings from these measures align with those obtained from MSE and MAE, with the exception of the GARCH model, which exhibits high forecast accuracy according to the QLIKE measure.

Table 8
Evaluation of weekly volatility forecasts based on the MAE measure for 5% of the lowest values of realized variances.

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	MAE	Rank	MAE	Rank	MAE	Rank	MAE	Rank
HAR	0.858	6	7.260	7	10.485	8	8.892	11
HAR-R	0.591	2	6.956	5	10.255	5	6.297	10
ARFIMA	1.104	9	**6.321**	1	**9.881**	1	4.610	3
GARCH	6.422	12	15.854	12	16.472	12	20.998	12
LASSO	2.002	11	7.313	8	10.295	6	5.036	6
RR	1.044	7	6.838	3	10.109	3	4.756	4
SVR-G	**0.586**	1	7.120	6	10.900	11	4.481	2
SVR-L	0.835	4	6.894	4	10.657	10	5.093	7
MLP	1.060	8	7.848	11	10.100	2	**4.429**	1
FNM	0.852	5	7.394	9	10.370	7	5.178	8
RF	0.715	3	7.467	10	10.148	4	5.346	9
LSTM	1.875	10	6.651	2	10.615	9	4.773	5

Note: the values of MAE are multiplied by 10^3, the lowest values of MAE are in bold. The evaluation period is January 1, 2019 - December 31, 2021.

Table 9
Evaluation of weekly volatility forecasts based on the MAE measure for 5% of the highest values of realized variances.

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	MAE	Rank	MAE	Rank	MAE	Rank	MAE	Rank
HAR	6.027	6	0.426	8	1.337	8	0.532	5
HAR-R	6.016	5	0.439	9	1.364	10	0.564	10
ARFIMA	6.082	8	0.413	5	1.241	2	0.494	2
GARCH	6.181	10	0.558	11	1.307	7	**0.437**	1
LASSO	6.372	11	**0.357**	1	**1.227**	1	0.498	3
RR	5.883	3	0.412	4	1.247	3	0.534	6
SVR-G	5.700	2	0.444	10	1.288	5	0.540	8
SVR-L	5.902	4	0.415	7	1.269	4	0.550	9
MLP	**5.500**	1	0.413	6	1.440	11	0.540	7
FNM	6.060	7	0.412	3	1.301	6	0.503	4
RF	6.107	9	0.399	2	1.346	9	0.569	11
LSTM	6.526	12	0.690	12	2.037	12	1.060	12

Note: the values of MAE are multiplied by 10^2, the lowest values of MAE are in bold. The evaluation period is January 1, 2019 - December 31, 2021.

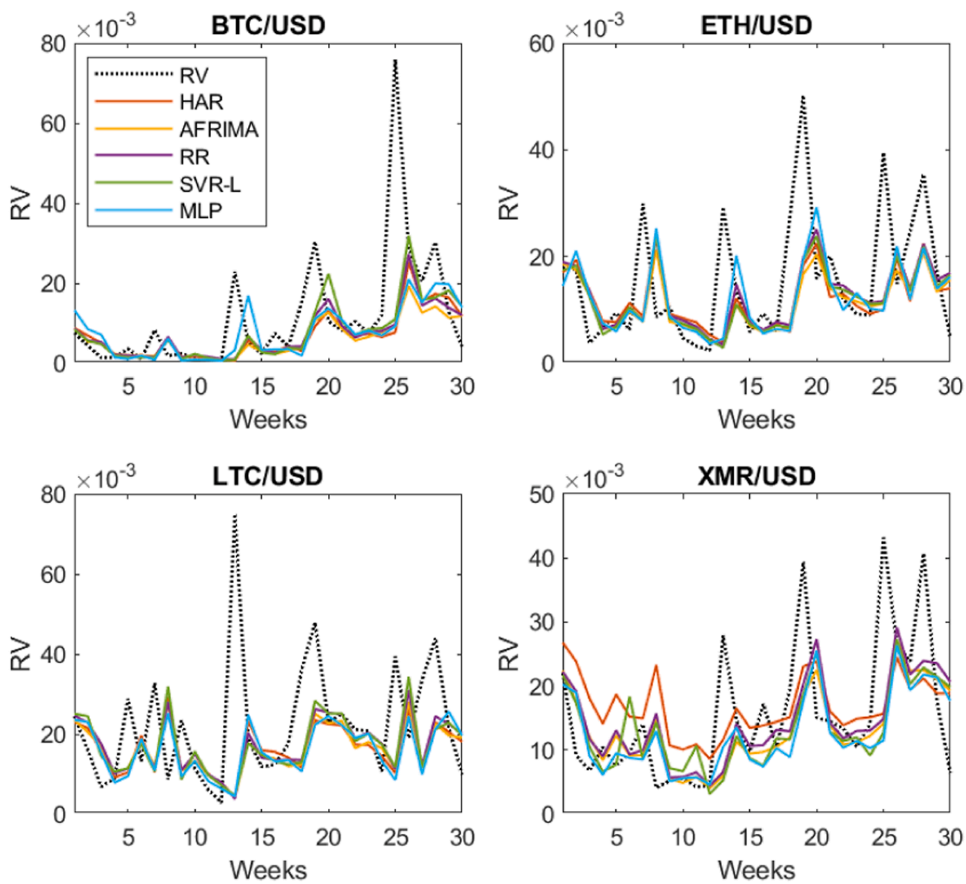

Fig. 4. Examples of forecasts of the weekly volatility.

We calculate also the MAE measure for 5% of the lowest and 5% of the highest values of realized variances. These results are given in Tables 8 and 9.

Generally, the most accurate forecasts of low volatility are based on the RR and ARFIMA models, whereas the least precise forecasts are from the GARCH and HAR models. When volatility is high, the best forecasting models are LASSO, RR and ARFIMA, while the worst performing are the HAR-R and LSTM models.

Fig. 4 shows examples of the weekly forecasts produced by selected models for the first 30 weeks of the test period (January-July 2019). It is clear from the figure that during the periods of increased volatility all models clearly underestimate it. As for daily forecasting, none of the

Table 10
Training and prediction times for ML models in seconds.

Method	LASSO	RR	SVR-G	SVR-L	MLP	FNM	RF	LSTM
Matlab function	fitrlinear	fitrlinear	fitrsvm	fitrsvm	train	-	TreeBagger	trainNetwork
Time	0.017	0.016	0.028	0.040	0.360	< 0.001	0.250	504/16.4/6.4

Note: the times for LSTM are for training using CPU / NVIDIA GeForce GTX 1050 Ti 4 GB / NVIDIA Tesla V100 32 GB.

methods can predict spikes in volatility and delay is also observed in forecasts.

4.4. Computation time

Table 10 provides a comparison of the total time required for a single training session and for generating forecasts for the next seven days for the ML models. Additionally, this table references the Matlab functions that implement these models. The experiments were conducted using Matlab 2023a, on a computer equipped with a ten-core CPU (Intel(R) Core(TM) i7–6950X 3.00 GHz). Regarding LSTM, we present three distinct times: one for training using CPU, and two additional times for training using different GPUs, specifically the NVIDIA GeForce GTX 1050 Ti 4 GB and NVIDIA Tesla V100 32 GB.

Table 10 reveals distinct variations in computation time among the different forecasting models. Notably, FNM emerges as the fastest model, primarily because it is a lazy learner model that does not require training. However, it is important to highlight that FNM's forecast calculation relies on computing distances between the query pattern and training patterns, which happens to be the most intricate and time-consuming operation within the FNM framework. Following closely, in terms of computation time, are the linear models with regularization, RR and LASSO. These models involve estimating only four parameters. Additionally, both methods employ an efficient fitrlinear implementation, which swiftly minimizes objective functions. SVR-based models exhibit a slightly slower computation time than RR and LASSO. The fitrsvm implementation accommodates predictor data mapping using kernel functions and supports potent optimization algorithms, including the Sequential Minimal Optimization algorithm, Iterative Single Data algorithm, and L_1 soft-margin minimization via quadratic programming. RF, on the other hand, requires even more training time than SVR due to being an ensemble model composed of numerous regression trees trained sequentially. The MLP neural model, trained using the backpropagation method with the Levenberg-Marquardt algorithm and Bayesian regularization, requires additional training time compared to the aforementioned methods. The Levenberg-Marquardt algorithm involves iterative calculations of Jacobians, naturally extending the training duration. However, LSTM stands out as the slowest model, owing to its substantially larger parameter set than other models. Furthermore, given the recursive nature of LSTM, the optimization challenge here is notably more intricate than in the case of MLP, necessitating training using the backpropagation through time algorithm. Nevertheless, leveraging parallel training with GPU significantly reduces the training time of LSTM to approximately 6 seconds, although it remains relatively time-consuming compared to other models.

4.5. Comparative approaches

In this section, we compare the best results achieved by our models with the results obtained using alternative approaches. These alternative approaches utilize ensembling, time series decomposition, and hybridization of classical and deep learning methods. They include:

- Ensemble – simple averaging of predictions generated by our 12 forecasting models.
- cES-adRNN – a contextually enhanced hybrid and hierarchical architecture combining exponential smoothing and recurrent neural network proposed in [77]. This state-of-the-art model represents an advanced approach to forecasting with a wide range of sophisticated techniques and features. It incorporates various mechanisms and procedures to improve forecasting accuracy, including two simultaneously trained tracks, multiple stacked recurrent layers with hierarchical dilations, attentive dilated recurrent cells, dynamic exponential smoothing component, cross-learning, ensembling, and mechanisms to mitigate overfitting.

Table 11
Evaluation of daily volatility forecasts for comparative approaches.

Method	BTC/USD				ETH/USD				LTC/USD				XMR/USD			
	MSE	*p*-value	MAE	*p*-value	MSE	*p*-value	MAE	*p*-value	MSE	*p*-value	MAE	*p*-value	MSE	*p*-value	MAE	*p*-value
Best	0.186	0.359 *	**0.102**	1.000 *	**0.315**	1.000 *	**0.146**	1.000 *	0.580	0.398 *	**0.206**	1.000 *	0.542	0.748 *	**0.165**	1.000 *
Ensemble	0.194	0.359 *	0.102	0.838 *	0.335	0.620 *	0.148	0.366 *	0.593	0.398 *	0.207	0.379 *	0.554	0.725 *	0.165	0.665 *
cES-adRNN	**0.174**	1.000 *	0.113	0.000	0.320	0.913 *	0.156	0.009	**0.543**	1.000 *	0.211	0.341 *	**0.522**	1.000 *	0.180	0.000
RandNN+STD	0.195	0.359 *	0.106	0.110 *	0.327	0.873 *	0.153	0.053	0.585	0.398 *	0.215	0.020	0.551	0.748 *	0.172	0.057

Note: the values of MAE are multiplied by 10^2, the values of MSE are multiplied by 10^4, the lowest values of MSE and MAE are in bold, 'Best' means the supreme methods from Section 4.2, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

Table 12
Evaluation of weekly volatility forecasts for comparative approaches.

Method	BTC/USD				ETH/USD				LTC/USD				XMR/USD			
	MSE	p-value	MAE	p-value	MSE	*p-value*	MAE	*p-value*	MSE	*p-value*	MAE	*p-value*	MSE	*p-value*	MAE	*p-value*
Best	0.358	0.899 *	**0.688**	1.000 *	**0.713**	1.000 *	**1.001**	1.000 *	1.034	0.911 *	**1.406**	1.000 *	**1.125**	1.000 *	**1.130**	1.000 *
Ensemble	0.380	0.353 *	0.690	0.757 *	0.740	0.347 *	1.014	0.407 *	1.061	0.013	1.439	0.070	1.152	0.497 *	1.131	0.936 *
cES-adRNN	**0.352**	1.000 *	0.800	0.000	0.719	0.899 *	1.075	0.190 *	**1.024**	1.000 *	1.441	0.481 *	1.153	0.704 *	1.258	0.029
RandNN+STD	0.409	0.274 *	0.787	0.007	0.814	0.247 *	1.183	0.148 *	1.130	0.013	1.570	0.010	1.440	0.255 *	1.381	0.026

Note: the values of MAE are multiplied by 10^2, the values of MSE are multiplied by 10^3, the lowest values of MSE and MAE are in bold, 'Best' means the supreme methods from Section 4.3, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

- RandNN+STD – an ensemble forecasting method proposed in [28], that combines randomized neural networks (RandNNs) and seasonal-trend-dispersion decomposition of time series (STD). The STD products are employed to encode the input and output time series data, simplifying the relationship between them and streamlining the forecasting process. RandNNs are used as ensemble members for their capability to control diversity and very fast training.

Tables 11 and 12 present the errors for daily and weekly volatility forecasting. As can be seen from these tables, it is evident that cES-adRNN achieved the lowest MSE for BTC, LTC, and XMR in daily forecasts. However, it also resulted in the highest MAE for BTC, ETH, and XMR. Ensembling effectively improves the average error of the base models and approaches the performance of the best models in terms of MAE but not MSE. It is important to note that ensembling is a time-consuming approach as it involves optimizing and training all 12 base models. For weekly forecasts, similar to daily forecasts, the results are inconclusive and do not indicate a clear superior approach. In conclusion, this research suggests that neither ensembling, decomposition, nor hybridization yields a definitive improvement in the accuracy of daily and weekly forecasts of volatility.

4.6. Attempts to explain the lack of significant advantage of ML methods

In order to enhance our findings, we conduct supplementary analyses in an effort to identify the factors contributing to the near-equivalence of simpler and more complex methods in terms of efficiency.

Initially, we explore the possibility that outliers within the dataset might be influencing our results. As elucidated in Section 3, the presence of substantial outliers persisted in the realized variances, even after a logarithmic transformation. It is well-documented that outliers can distort patterns in the data, thereby hampering the effectiveness of forecasting. To address this concern, we remove outliers from the time series of realized variances under investigation. In this study, outliers are defined as observations exceeding $Q3 + 1.5IQR$, where $Q3$ represents the upper quartile, and IQR denotes the interquartile range. For the BTC/USD series, this outliers removal process lead to the exclusion of 172 observations (approximately 9.4% of the data) from the period spanning January 1, 2017, to December 31, 2021. The results for daily volatility forecasts for BTC/USD can be found in Table A7 in the Appendix. Although there are some changes in the rankings of the compared methods, overall, no substantial enhancement in the performance of ML methods over simpler methods is observed. Consequently, we are unable to confirm the hypothesis that the presence of outliers is the primary reason for the observed absence of superiority of complex methods over simpler ones.

Next, we examine whether our findings could be attributed to the absence of nonlinear relationships within the processes under investigation. To address this, we employ the RESET test [71] to verify the null hypothesis of the appropriateness of the HAR model specification. Our study considers the entire investigation period from January 1, 2017, to December 31, 2021. The results obtained (see Table A8 in the Appendix) reveal no compelling evidence to suggest that the analyzed data are driven by nonlinear relationships.

However, owing to the potential existence of various forms of nonlinearity and the limited power of various tests, an additional investigation is conducted, involving the application of four measures of dependence: Pearson, Spearman, and Kendall tau correlation coefficients, as well as the mutual information measure. In our study, we not only calculate these measures but also conduct permutation tests to assess their significance.

While Pearson's correlation coefficient is the most widely used measure for quantifying relationships between two variables, it primarily assesses linear relationships. Spearman's and Kendall's coefficients are applied for ordinal data and can capture varied, including nonlinear, monotonic relationships. The mutual information measure (MI) is one of the most important tool for detecting nonlinear dependencies in time series [31,40,41]. Its conceptual link to Shannon entropy suggests that MI quantifies the extent to which knowledge of one variable enhances information about another. Specifically, it assesses the ability of one variable to predict the other.

We employ these measures to explore relationships between $\ln RV_{d,t}$ and the explanatory variables in our forecasting models, namely $\ln RV_{d,t-1}, \ln RV_{w,t-1}, \ln RV_{m,t-1}$, as well as the relationships between the residuals from the HAR model and these three variables. Detecting dependencies in the residuals would imply the presence of neglected nonlinear relationships between $\ln RV_{d,t}$ and the variables under examination.

The results of this analysis for BTC/USD are presented in the Appendix. Scatter plots (Figure A1) and the data in Table A9 illustrate strong relationships between $\ln RV_{d,t}$ and each of $\ln RV_{d,t-1}$, $\ln RV_{w,t-1}$, $\ln RV_{m,t-1}$. However, the results for the residuals from the HAR model are less definitive. Visual analysis of the scatter plots (Figure A2) do not confirm the presence of clear dependencies in the filtered data. This is also formally demonstrated in Table A10 for three out of four dependence measures applied, specifically the Pearson, Spearman, and Kendall correlation coefficients. These outcomes suggest that there is insufficient evidence to support the hypothesis of nonlinear relationships between $lnRV_{d,t}$ and the analyzed explanatory variables.

Nevertheless, a different conclusion emerges with the MI measure. Its significance for the residuals from the HAR model indicates the presence of nonlinear relationships between $\ln RV_{d,t}$ and each of the three explanatory variables. The divergent results may be attributed to the unique nature of the MI measure, as it has the potential to identify a different class of dependencies compared to the aforementioned coefficients.

In summary, it should be noted that most of the research methods employed indicate that the relationships between $\ln RV_{d,t}$ and $\ln RV_{d,t-1}$, $\ln RV_{w,t-1}, \ln RV_{m,t-1}$ are linear. This observation may explain why the more complex ML methods do not outperform the simpler linear techniques. However, the presence of nonlinear relationships as indicated by the MI measure justifies further investigation into other methods that can leverage these relationships to enhance forecast accuracy.

4.7. Discussion

Cryptocurrencies volatility is very hard to predict due to lack of a distinctive pattern in time series, no strong seasonality, occurrence of chaotic trend and large random fluctuations [56]. A number of factors contribute to the fact that cryptocurrencies are highly volatile assets, and predicting their movements is extremely difficult. They include [10,12,18,22]:

1. Lack of fundamental value. Unlike traditional assets, such as stocks or commodities, cryptocurrencies do not have a clear fundamental value that can be used to predict their price movements. Their value is largely based on market demand, which can be highly volatile and difficult to predict.
2. Market sentiment and speculation. The cryptocurrency market is largely driven by market sentiment and speculation, which can be influenced by a variety of factors such as news events, social media, and rumours. These factors can be difficult to quantify and predict, making it challenging to forecast market movements.
3. Lack of regulation. The lack of regulatory oversight in the cryptocurrency market can make it more susceptible to price manipulation and insider trading, which can further increase volatility and make predictions more difficult.
4. Immaturity and low market capitalization. The cryptocurrency market is dominated by small investors, making it inefficient. That is why a large order (from the so-called 'whales') may significantly impact prices and cause jumps. Consequently, cryptocurrencies are more susceptible to jumps when compared to more mature markets. Additionally, the relatively low market capitalization of many cryptocurrencies means that even small changes in demand can have a significant impact on price.
5. Limited historical data. Cryptocurrencies are a relatively new asset class, with limited historical data available for analysis. This makes it difficult to identify patterns and trends that can be used to make accurate predictions.

Our findings on exploring the effectiveness of various modelling techniques in forecasting the volatility of cryptocurrencies can be summarized as follows:

1. It is not possible to identify the best model, because depending on the cryptocurrency and the accuracy measure used, various models were ranked at the top of the accuracy rankings. The following models were most often at the forefront: HAR-based models, AFRIMA, and SVR-based models for daily volatility forecasting, and RR, SVR-based models, and FNM for weekly volatility forecasting.
2. ML models do not produce significantly more accurate predictions than classical statistical models based on the realized variance. Typically, ML models show their advantage over statistical models due to ability to nonlinear modelling, large-scale forecasting, feature engineering and robustness. These advantages prove useless in forecasting noisy cryptocurrency volatility.
3. Optimizing the ML models is a challenging task, as the hyperparameters that perform well on the validation sets do not necessarily generalize well to the test sets. This can be attributed to the nature of time series, which are noisy and lack clear patterns, seasonal cycles, and stable trends. Therefore, the insights gained from the validation sequence may not carry over to the test sequence, making hyperparameter tuning a difficult task.
4. We try to improve the performance of the ML models by making various modifications. They include defining input data as separate 30 lags of $RV_{d,t}$, skipping log transformation of data, using principal component analysis to extract inputs, and defining separate models for each day of the week. Unfortunately, none of these modifications improve performance.
5. Among the ML methods, the SVR models are outstanding. Both SVR-L and SVR-G models stay out in terms of predictive power for both daily and weekly forecasts. Our study does not show significant differences between linear and RBF kernels in SVR models.
6. Despite being the only model among the group of ML models capable of modelling dependencies in sequential data, the LSTM model proves to be a big disappointment in our study. While LSTM can capture short- and long-range temporal dependencies in time series, it struggles to forecast cryptocurrencies data due to its highly unpredictable and chaotic nature. We observe that LSTM had difficulty processing time series data expressed as $\ln RV_t$, likely due to the presence of negative values. However, when we transform the data using $\ln(1+100RV_t)$ to obtain positive values, the LSTM model's performance improves.
7. The comparative study utilizing ensembling, time series decomposition, and hybridization of classical and deep learning models do not yield significant improvements. Despite incorporating state-of-the-art time series processing mechanisms, these models do not produce clearly superior results. The lack of clear advantage could be attributed to the chaotic nature of the time series, characterized by a lack of regularity and discernible patterns.

8. In our study, we utilize data spanning from midnight to midnight, which means that the daily realized variance in formula (2) is computed over 24 h, starting at 00:00 and concluding at 24:00. To assess the robustness of our findings, we also perform a check using data from noon to noon, ranging from 12:00 to 12:00 on the following day. The results, pertaining to daily volatility forecasts for BTC/USD, are presented in Table A11 within the Appendix. Our evaluation, based on the MSE measure, reveals that seven models are included in the model confidence set. This is two more models than the set generated using data from midnight to midnight. Similarly, for the MAE criterion, exactly the same models are retained within the model confidence set. In conclusion, our analysis indicates that the choice of the starting hour for the data does not significantly influence the results.
9. The main findings are robust to the applied forecasting evaluation measures. The conclusions which result from the MSE and MAE criteria are confirmed by the R^2, MAPE and QLIKE measures (see Tables A1-A6 in the Appendix).

5. Conclusions

Cryptocurrencies are highly volatile and their price movements are affected by a variety of factors, including market sentiment, regulatory changes, and technological advancements. Additionally, the absence of clear patterns in the price data and its strong random fluctuations make it extremely challenging to predict their volatility.

In our study, we compare ML and statistical models for forecasting cryptocurrency volatility. We find that identifying the best forecasting model for this task is challenging, as the performance of different models varies depending on the specific cryptocurrency and the choice of error metric. For daily forecasts, linear SVR consistently appears in the set of best models based on MSE and MAE measures, while for weekly forecasts, FNM and RR are the top performers. The GARCH model, commonly used for volatility modelling, proves to be inadequate for cryptocurrencies. Forecasts generated by the GARCH model using squared daily returns are poor compared to models based on realized variance calculated from intraday prices.

Interestingly, ML models do not outperform classical statistical models. Despite ML models' advantages in nonlinear modelling, robustness, and flexibility, these characteristics do not necessarily translate into improved accuracy for volatile cryptocurrency markets, characterized by chaotic behaviour, large swings, and outliers. Surprisingly, simple methods like HAR or RR are not significantly worse than more advanced techniques like LSTM or RF. Moreover, our findings show that attempts to enhance accuracy through ensembling, time series decomposition, and model hybridization do not yield significant improvements.

Our findings can provide a useful reference for the development of more sophisticated models. Incorporating additional information, such as market sentiment indicators, blockchain data, economic indicators, technical analysis indicators, and price data from related markets into the input variables may also be an interesting avenue for further research.

CRediT authorship contribution statement

Grzegorz Dudek: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition. **Piotr Fiszeder:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition, Project administration. **Paweł Kobus:** Conceptualization, Software, Validation, Investigation, Visualization, Data curation. **Witold Orzeszko:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Piotr Fiszeder reports financial support was provided by National Science Centre. Witold Orzeszko reports financial support was provided by National Science Centre. Grzegorz Dudek reports a relationship with Czestochowa University of Technology that includes: employment. Piotr Fiszeder and Witold Orzeszko report a relationship with Nicolaus Copernicus University that includes: employment. Paweł Kobus reports a relationship with Warsaw University of Life Sciences that includes: employment.

Data Availability

Data will be made available on request.

Acknowledgments

We would like to express our gratitude to Slawek Smyl for providing the results of the cES-adRNN model. Grzegorz Dudek thanks Professor Wolfgang Karl Härdle, under whose guidance he had the opportunity to learn the secrets of cryptocurrencies and blockchain technology. Piotr Fiszeder was supported by the National Science Centre project number 2021/43/B/HS4/00353 entitled “Robust methods for range-based models - Risk and comovement analysis on the cryptocurrency market” and by Institutional support of the Prague University of Economics and Business no. IP100040; Witold Orzeszko was supported by the National Science Centre project number 2019/35/B/HS4/00642 entitled “Forecasting financial markets using a machine learning approach”. Grzegorz Dudek was supported by the project number 020/RID/2018/19 financed under the program of the Polish Minister of Science and Higher Education entitled "Regional Initiative of Excellence" in the years 2019-23.

Appendix A

Table A1
Evaluation of daily volatility forecasts based on the R^2 measure.

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	R^2	Rank	R^2	Rank	R^2	Rank	R^2	Rank
HAR	0.177	2	0.231	7	0.139	7	0.185	5
HAR-R	0.176	3	0.233	6	0.143	5	0.186	3
ARFIMA	0.166	6	0.238	3	0.151	4	0.185	4

(continued on next page)

Table A1 (*continued*)

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	R^2	Rank	R^2	Rank	R^2	Rank	R^2	Rank
GARCH	0.046	12	0.235	5	0.083	12	0.189	2
LASSO	0.163	7	0.205	8	**0.159**	1	0.183	7
RR	0.168	4	0.237	4	0.155	2	0.184	6
SVR-G	0.143	10	**0.249**	1	0.142	6	0.172	9
SVR-L	0.167	5	0.242	2	0.155	3	**0.197**	1
MLP	**0.202**	1	0.196	9	0.125	8	0.178	8
FNM	0.153	8	0.132	12	0.092	10	0.095	12
RF	0.111	11	0.160	11	0.122	9	0.125	11
LSTM	0.143	9	0.165	10	0.088	11	0.150	10

Note: The highest values of R^2 are in bold. The evaluation period is January 1, 2019 - December 31, 2021.

Table A2
Evaluation of daily volatility forecasts based on the MAPE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	MAPE	Rank	*p*-value	MAPE	Rank	*p*-value	MAPE	Rank	*p*-value	MAPE	Rank	*p*-value
HAR	59.683	2	0.000	54.394	4	0.000	56.522	5	0.000	54.108	11	0.000
HAR-R	**55.832**	1	1.000 *	**51.003**	1	1.000 *	**53.404**	1	1.000 *	51.049	8	0.407 *
ARFIMA	61.335	4	0.000	54.618	5	0.000	56.001	3	0.000	50.940	7	0.386 *
GARCH	180.127	12	0.000	145.062	12	0.000	117.588	12	0.000	120.028	12	0.000
LASSO	68.788	10	0.000	66.107	11	0.000	58.436	11	0.000	53.286	10	0.000
RR	62.408	9	0.000	57.092	10	0.000	57.558	8	0.000	52.792	9	0.000
SVR-G	61.523	6	0.000	54.636	6	0.000	57.203	7	0.000	50.398	4	0.407 *
SVR-L	61.911	7	0.000	55.225	8	0.000	57.103	6	0.000	50.689	6	0.407 *
MLP	59.704	3	0.001	54.314	3	0.000	56.051	4	0.000	49.732	2	0.407 *
FNM	61.345	5	0.000	55.826	9	0.000	57.984	10	0.000	50.677	5	0.407 *
RF	62.113	8	0.000	54.687	7	0.000	57.846	9	0.000	50.121	3	0.407 *
LSTM	78.349	11	0.000	52.764	2	0.265 *	54.479	2	0.527 *	**47.969**	1	1.000 *

Note: The lowest values of MAPE are in bold. The evaluation period is January 1, 2019 - December 31, 2021, * indicates that models belong to MCS with a confidence level of 0.90.

Table A3
Evaluation of daily volatility forecasts based on the QLIKE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	QLIKE	Rank	*p*-value	QLIKE	Rank	*p*-value	QLIKE	Rank	*p*-value	QLIKE	Rank	*p*-value
HAR	-5.476	7	0.000	-5.133	6	0.000	-4.742	4	0.000	-4.983	4	0.015
HAR-R	-5.429	10	0.000	-5.112	9	0.000	-4.718	11	0.000	-4.975	7	0.000
ARFIMA	-5.514	3	0.069	-5.136	5	0.000	-4.742	5	0.000	-4.982	5	0.015
GARCH	-5.466	8	0.000	-5.088	11	0.000	-4.722	10	0.000	-4.893	11	0.000
LASSO	-5.514	4	0.693 *	-5.138	4	0.002	**-4.759**	1	1.000 *	**-4.994**	1	1.000 *
RR	-5.525	2	0.802 *	**-5.153**	1	1.000 *	-4.757	2	0.639 *	-4.992	2	0.096
SVR-G	-5.485	6	0.000	-5.141	3	0.000	-4.755	3	0.639 *	-4.976	6	0.000
SVR-L	**-5.528**	1	1.000 *	-5.146	2	0.012	-4.738	6	0.266 *	-4.985	3	0.080
MLP	-5.501	5	0.025	-5.129	7	0.000	-4.733	8	0.000	-4.974	8	0.000
FNM	-5.465	9	0.000	-5.112	8	0.000	-4.733	7	0.000	-4.942	9	0.000
RF	-5.428	11	0.000	-5.097	10	0.000	-4.731	9	0.000	-4.938	10	0.000
LSTM	-5.312	12	0.000	-4.573	12	0.000	-4.019	12	0.000	-4.445	12	0.000

Note: The lowest values of QLIKE are in bold. The evaluation period is January 1, 2019 - December 31, 2021, * indicates that models belong to MCS with a confidence level of 0.90.

Table A4
Evaluation of weekly volatility forecasts based on the R^2 measure.

Method	BTC/USD		ETH/USD		LTC/USD		XMR/USD	
	R^2	Rank	R^2	Rank	R^2	Rank	R^2	Rank
HAR	0.102	9	0.131	9	0.116	10	0.106	11
HAR-R	0.099	10	0.133	8	0.121	9	0.114	10
ARFIMA	0.118	6	0.150	7	0.147	8	0.141	5
GARCH	0.039	12	0.074	11	0.047	12	0.048	12
LASSO	0.084	11	0.048	12	0.156	4	0.122	9
RR	0.127	4	0.162	4	0.153	5	0.141	6
SVR-G	0.115	7	0.179	3	0.148	7	0.140	7
SVR-L	0.124	5	0.160	5	0.153	6	0.143	3
MLP	**0.172**	1	0.153	6	0.159	3	0.141	4
FNM	0.137	2	0.183	2	0.168	2	0.162	2
RF	0.131	3	**0.211**	1	**0.188**	1	**0.164**	1
LSTM	0.113	8	0.099	10	0.052	11	0.139	8

Note: The highest values of R^2 are in bold. The evaluation period is January 1, 2019 - December 31, 2021.

Table A5
Evaluation of weekly volatility forecasts based on the MAPE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	MAPE	Rank	*p*-value	MAPE	Rank	*p*-value	MAPE	Rank	*p*-value	MAPE	Rank	*p*-value
HAR	**44.466**	1	1.000 *	44.443	4	0.425 *	46.742	3	0.727 *	44.525	10	0.006
HAR-R	44.613	2	0.942 *	**42.958**	1	1.000 *	**45.471**	1	1.000 *	40.337	5	0.234 *
ARFIMA	44.922	3	0.942 *	43.513	2	0.698 *	46.38	2	0.727 *	**37.717**	1	1.000 *
GARCH	128.704	12	0.000	110.854	12	0.000	84.863	12	0.000	95.038	12	0.000
LASSO	57.470	10	0.002	62.805	11	0.000	48.927	10	0.638 *	41.514	8	0.032
RR	46.022	5	0.779 *	46.315	8	0.103 *	47.407	6	0.727 *	40.933	7	0.032
SVR-G	48.129	8	0.779 *	44.692	5	0.324 *	47.442	7	0.727 *	40.746	6	0.075
SVR-L	46.957	7	0.779 *	44.883	6	0.324 *	47.707	8	0.686 *	41.626	9	0.032
MLP	50.298	9	0.491 *	49.165	9	0.0207	47.737	9	0.711 *	39.559	3	0.347 *
FNM	46.534	6	0.779 *	44.346	3	0.695 *	46.914	5	0.727 *	39.864	4	0.335 *
RF	45.348	4	0.942 *	45.088	7	0.324 *	46.802	4	0.727 *	39.168	2	0.347 *
LSTM	63.413	11	0.000	55.691	10	0.000	60.901	11	0.000	53.879	11	0.000

Note: The lowest values of MAPE are in bold. The evaluation period is January 1, 2019 - December 31, 2021, * indicates that models belong to MCS with a confidence level of 0.90.

Table A6
Evaluation of weekly volatility forecasts based on the QLIKE measure and the MCS test.

Method	BTC/USD			ETH/USD			LTC/USD			XMR/USD		
	QLIKE	Rank	*p*-value	QLIKE	Rank	*p*-value	QLIKE	Rank	*p*-value	QLIKE	Rank	*p*-value
HAR	-2.866	10	0.000	-2.914	10	0.000	-2.492	10	0.000	-2.820	8	0.000
HAR-R	-2.629	12	0.000	-2.827	11	0.000	-2.398	11	0.000	-2.800	11	0.000
ARFIMA	-3.157	6	0.000	-2.963	6	0.000	-2.554	7	0.048	-2.853	3	0.010
GARCH	**-3.429**	1	1.000 *	**-3.029**	1	1.000 *	**-2.661**	1	1.000 *	-2.830	5	0.009
LASSO	-3.218	2	0.380 *	-2.952	8	0.000	-2.584	3	0.663 *	-2.876	2	0.529 *
RR	-3.192	3	0.404 *	-3.010	2	0.883 *	-2.583	5	0.663 *	**-2.883**	1	1.000 *
SVR-G	-2.928	9	0.000	-2.938	9	0.000	-2.543	8	0.000	-2.816	9	0.000
SVR-L	-3.141	8	0.072	-2.973	5	0.000	-2.573	6	0.402 *	-2.84	4	0.000
MLP	-3.171	5	0.404 *	-2.962	7	0.000	-2.522	9	0.000	-2.814	10	0.000
FNM	-3.179	4	0.000	-2.997	4	0.701 *	-2.584	4	0.663 *	-2.824	7	0.000
RF	-3.155	7	0.000	-3.008	3	0.883 *	-2.588	2	0.663 *	-2.827	6	0.000
LSTM	-2.840	11	0.000	-2.137	12	0.000	-1.058	12	0.000	-2.046	12	0.000

Note: The lowest values of QLIKE are in bold. The evaluation period is January 1, 2019 - December 31, 2021, * indicates that models belong to MCS with a confidence level of 0.90.

Table A7
Evaluation of daily volatility forecasts for BTC/USD based on the MSE and MAE measures and the MCS test for data without outliers.

Method	MSE			MAE		
	Value	Rank	*p*-value	Value	Rank	*p*-value
HAR	0.743	5	0.001	0.526	7	0.271 *
HAR-R	0.746	7	0.000	0.524	4	0.463 *
ARFIMA	0.729	2	0.187 *	0.523	3	0.505 *
GARCH	0.977	10	0.000	0.717	12	0.000
LASSO	1.010	11	0.000	0.652	10	0.000
RR	**0.721**	1	1.000 *	**0.521**	1	1.000 *
SVR-G	0.731	4	0.020	0.525	6	0.305 *
SVR-L	0.731	3	0.039	0.522	2	0.505 *
MLP	0.745	6	0.006	0.524	5	0.505 *
FNM	0.757	9	0.000	0.533	9	0.084
RF	0.753	8	0.000	0.530	8	0.159 *
LSTM	1.138	12	0.000	0.657	11	0.000

Note: The values of MSE are multiplied by 10^6, the values of MAE are multiplied by 10^3, the lowest values of MSE are in bold, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

Table A8
The results of the RESET test for correct specification of the HAR model for BTC/USD.

Variant of the test	*F*-statistic	*p*-value
Quadratic and cubic term	0.110	0.896
Only quadratic term	0.187	0.665
Only cubic term	0.175	0.676

Note: The analysed period is January 1, 2019 - December 31, 2021.

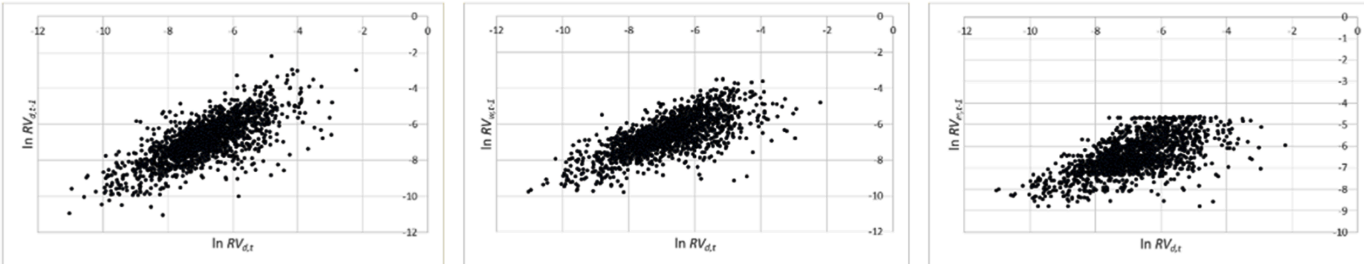

Fig. A1. Scatter plots of $\ln RV_{d,t}$ and the explanatory variables $\ln RV_{d,t-1}$, $\ln RV_{w,t-1}$, $\ln RV_{m,t-1}$ for BTC/USD (January 1, 2017 - December 31, 2021).

.

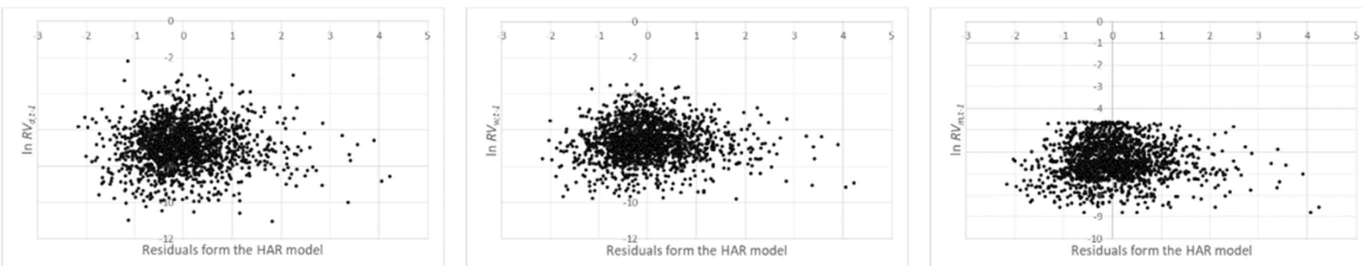

Fig. A2. Scatter plots of the residuals from the HAR model and the explanatory variables $\ln RV_{d,t-1}$, $\ln RV_{w,t-1}$, $\ln RV_{m,t-1}$ for BTC/USD (January 1, 2017 - December 31, 2021).

.

Table A9
The values of dependence measures between $\ln RV_{d,t}$ and the explanatory variables $\ln RV_{d,t-1}$, $\ln RV_{w,t-1}$, $\ln RV_{m,t-1}$ for BTC/USD.

Dependence measure	Variable		
	$\ln RV_{d,t-1}$	$\ln RV_{w,t-1}$	$\ln RV_{m,t-1}$
Pearson	0.736	0.710	0.616
	(**0.000**)	(**0.000**)	(**0.000**)
Spearman	0.715	0.696	0.605
	(**0.000**)	(**0.000**)	(**0.000**)
Kendall	0.539	0.518	0.436
	(**0.000**)	(**0.000**)	(**0.000**)
Mutual Information	0.173	0.156	0.123
	(**0.000**)	(**0.000**)	(**0.000**)

Note: Each cell contains the value of the measure and p-value in the permutation test for the significance (in brackets). P-values smaller than 0.05 are in bold. The analysed period is January 1, 2019 - December 31, 2021.

Table A10
The values of dependence measures between the residuals from the HAR model and the explanatory variables$\ln RV_{d,t-1}$, $\ln RV_{w,t-1}$, $\ln RV_{m,t-1}$ for BTC/USD.

Dependence measure	Variable		
	$\ln RV_{d,t-1}$	$\ln RV_{w,t-1}$	$\ln RV_{m,t-1}$
Pearson	0.000	0.000	0.000
	(0.972)	(0.990)	(0.998)
Spearman	0.015	0.026	0.028
	(0.519)	(0.270)	(0.237)
Kendall	0.010	0.017	0.019
	(0.527)	(0.281)	(0.234)
Mutual Information	0.014	0.017	0.021
	(**0.048**)	(**0.012**)	(**0.000**)

Note: Each cell contains the value of the measure and p-value in the permutation test for the significance (in brackets). P-values smaller than 0.05 are in bold. The analysed period is January 1, 2019 - December 31, 2021.

Table A11
Evaluation of daily volatility forecasts for BTC/USD based on the MSE and MAE measures and the MCS test for data observed from noon to noon.

Method	MSE			MAE		
	Value	Rank	*p*-value	Value	Rank	*p*-value
HAR	0.180	2	0.412 *	0.103	2	0.953 *
HAR-R	0.181	3	0.181 *	**0.103**	1	1.000 *
ARFIMA	0.183	6	0.120 *	0.104	4	0.746 *
GARCH	0.189	10	0.041	0.123	12	0.000
LASSO	0.187	8	0.010	0.107	9	0.029
RR	0.181	5	0.412 *	0.104	3	0.762 *
SVR-G	0.181	4	0.412 *	0.104	5	0.710 *
SVR-L	0.184	7	0.120 *	0.105	8	0.536 *
MLP	**0.172**	1	1.000 *	0.105	6	0.683 *
FNM	0.188	9	0.041	0.105	7	0.165 *
RF	0.193	12	0.041	0.108	10	0.029
LSTM	0.192	11	0.001	0.111	11	0.001

Note: The values of MSE are multiplied by 10^4, the values of MAE are multiplied by 10^2, the lowest values of MSE are in bold, *p*-value is for the MCS test, * indicates that models belong to MCS with a confidence level of 0.90. The evaluation period is January 1, 2019 - December 31, 2021.

References

[1] C. Alexander, M. Dakos, A critical investigation of cryptocurrency data and analysis, Quant. Financ. 20 (2) (2020) 173–188.
[2] H. Alqaralleh, A.A. Abuhommous, A. Alsaraireh, Modelling and forecasting the volatility of cryptocurrencies: A comparison of nonlinear GARCH-type models, Int. J. Financ. Res. 11 (4) (2020) 346–356.
[3] B. Amirshahi, S. Lahmiri, Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies, Mach. Learn. Appl. 12 (2023), 100465.
[4] T.G. Andersen, T. Bollerslev, Answering the skeptics: yes, standard volatility models do provide accurate forecasts, Int. Econ. Rev. 39 (4) (1998) 885–905.
[5] T.G. Andersen, T. Bollerslev, P.F. Christoffersen, F.X. Diebold, Volatility and correlations forecasting, in: G. Elliott, C.W.J. Granger, A. Timmermann (Eds.), Handbook of Economic Forecasting, Vol. 1, Elsevier B. V, Amsterdam, North-Holland, 2006.
[6] T.G. Andersen, T. Bollerslev, F.X. Diebold, P. Labys, Modelling and forecasting realized volatility, Econometrica 71 (2003) 529–626.
[7] S. Aras, On Improving GARCH Volatility forecasts for Bitcoin via a meta-learning approach, Knowl. -Based Syst. 230 (2021), 107393.
[8] S. Aras, Stacking hybrid GARCH models for forecasting bitcoin volatility, Expert Syst. Appl. 174 (2021), 114747.
[9] M. Awad, R. Khanna, Support vector regression, in: M. Awad, R. Khanna (Eds.), Efficient Learning Machines, Apress, Berkeley, CA, 2015, pp. 67–80.
[10] C. Baek, M. Elbeck, Bitcoins as an investment or speculative vehicle? A first look, Appl. Econ. Lett. 22 (1) (2015) 30–34.
[11] D.G. Baur, T. Dimpfl, K. Kuck, Bitcoin, gold and the US Dollar – a replication and extension, Financ. Res. Lett. 25 (2018) 103–110.
[12] D.G. Baur, K. Hong, A.D. Lee, Bitcoin: medium of exchange or speculative assets? *J. Int. Financ. Mark. , Inst. Money* 54 (2018) (2018) 177–189.
[13] L.Ø. Bergsli, A.F. Lind, P. Molnár, M. Polasik, Forecasting volatility of bitcoin, Res. Int. Bus. Financ. 59 (2022), 101540.
[14] T. Bollerslev, Generalised autoregressive conditional heteroskedasticity, J. Econ. 31 (1986) 307–327.
[15] E. Bouri, S.J.H. Shahzad, D. Roubaud, Co-explosivity in the cryptocurrency market, Financ. Res. Lett. 29 (2019) 178–183.
[16] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[17] S. Çelik, H. Ergin, Volatility forecasting using high frequency data: evidence from stock markets, Econ. Model. 36 (2014) 176–190.
[18] P. Chaim, M.P. Laurini, Volatility and return jumps in bitcoin, Econ. Lett. 173 (2018) 158–163.
[19] A. Charles, O. Darné, Volatility estimation for bitcoin: replication and robustness, Int. Econ. 157 (2019) 23–32.
[20] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Netw. 17 (1) (2004) 113–126.
[21] K. Christensen, M. Siggaard, B. Veliyev, A machine learning approach to volatility forecasting, Journal of Financial Econometrics 21 (5) (2023) 1680–1727.
[22] S. Corbet, B. Lucey, A. Urquhart, L. Yarovaya, Cryptocurrencies as a financial asset: a systematic analysis, Int. Rev. Financ. Anal. 62 (2019) 182–199.
[23] F. Corsi, A simple approximate long-memory model of realized volatility, J. Financ. Econ. 7 (2009) 174–196.
[24] A. Cretarola, G. Figà-Talamanca, Bubble regime identification in an attention-based model for bitcoin and ethereum price dynamics, Econ. Lett. 191 (2020), 108831.
[25] V. D'Amato, S. Levantesi, G. Piscopo, Deep learning in predicting cryptocurrency volatility, Phys. A: Stat. Mech. its Appl. 596 (2022), 127158.
[26] L. Du, R. Gao, P.N. Suganthan, D.Z.W. Wang, Bayesian optimization based dynamic ensemble for time series forecasting, Inf. Sci. 591 (2022) 155–175.
[27] G. Dudek, Pattern similarity-based methods for short-term load forecasting – part 2: models, Appl. Soft Comput. 36 (2015) 422–441.
[28] G. Dudek, Ensemble of randomized neural networks with STD decomposition for forecasting time series with complex seasonality. Proc. 17th International Work-Conference on Artificial Neural Networks, IWANN'23, LNCS 14134, pp. 690-702. Springer, Cham 2023.
[29] F.A. Enoksen, J. Landsnes Ch, K. Lučivjanská, P. Molnár, Understanding risk of bubbles in cryptocurrencies, J. Econ. Behav. Organ. 176 (2020) 129–144.
[30] M. Fałdziński, P. Fiszeder, W. Orzeszko, Forecasting volatility of energy commodities: comparison of GARCH models with support vector regression, Energies 14 (1) (2021) 6.
[31] P. Fiszeder, W. Orzeszko, Nonparametric verification of GARCH-class models for selected polish exchange rates and stock indices, Financ. a úvěr - Czech J. Econ. Financ. 62 (5) (2012) 430–449.
[32] P. Fiszeder, M. Fałdziński, P. Molnár, Range-based DCC models for covariance and value-at-risk forecasting, J. Empir. Financ. 54 (2019) 58–76.
[33] P. Fiszeder, W. Orzeszko, Covariance matrix forecasting using support vector regression, Appl. Intell. 51 (10) (2021) 7029–7042.
[34] A. García-Medina, E. Aguayo-Moreno, LSTM–GARCH hybrid model for the prediction of volatility in cryptocurrency portfolios, Comput. Econ. (2023).
[35] R. Gao, L. Du, P.N. Suganthan, Q. Zhou, K.F. Yuen, Random vector functional link neural network based ensemble deep learning for short-term load forecasting, Expert Syst. Appl. 206 (2022), 117784.
[36] V.V. Gavrishchaka, S.B. Ganguli, Volatility forecasting from multiscale and high-dimensional market data, Neurocomputing 55 (2003) 285–305.
[37] J. Geuder, H. Kinateder, N. Wagner, Cryptocurrencies as financial bubbles: the case of bitcoin, Financ. Res. Lett. 31 (2019) 179–184.
[38] K. Gkillas, M. Tantoula, M. Tzagarakis, Transaction activity and bitcoin realized volatility, Oper. Res. Lett. 49 (5) (2021) 715–719.
[39] C.W.J. Granger, R. Joyeux, An introduction to long-memory time series models and fractional differencing, J. Time Ser. Anal. 1 (1980) 15–29.
[40] C.W.J. Granger, T. Terasvirta, Modelling Nonlinear Economic Relationship, University Press, Oxford, 1993.
[41] C.W.J. Granger, J.-L. Lin, Using the mutual information coefficient to identify lags in nonlinear models, J. Time Ser. Anal. 15 (1994) 371–384.
[42] T. Guo, A. Bifet, N. Antulov-Fantulin, Bitcoin volatility forecasting with a glimpse into buy and sell orders, Proc. IEEE Int. Conf. Data Mining (ICDM), (2018) 989–994.
[43] P. Hansen, A. Lunde, Consistent ranking of volatility models, J. Econ. 131 (1–2) (2006) 97–121.
[44] P.R. Hansen, A. Lunde, J.M. Nason, The model confidence set, Econometrica 79 (2011) 453–497.
[45] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning. Data Mining, Inference, and Prediction, Springer,, New Jork, 2009.
[46] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, Int. J. Forecast. 37 (2021) 388–427.
[47] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[48] A. Hoerl, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.
[49] J.R. Hosking, Fractional differencing, Biometrika 68 (1) (1981) 165–176.
[50] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer,, Berlin, 2021.
[51] R. Khaldi, A. El Afia, R. Chiheb, Forecasting of BTC volatility: Comparative study between parametric and nonparametric models, Prog. Artif. Intell. 8 (4) (2019) 511–523.
[52] F.U. Khan, F. Khan, P.A. Shaikh, Forecasting returns volatility of cryptocurrency by applying various deep learning algorithms, Future Bus. J. 9 (2023) 25.
[53] S.J. Koopman, B. Jungbacker, E. Hol, Forecasting daily variability of the S&P100 stock index using historical, realised and implied volatility measurements, J. Empir. Financ. 12 (3) (2005) 445–475.

[54] 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.
[55] M. Kuhn, K. Johnson, Applied Predictive Modelling, Springer, New York, 2013.
[56] N.A. Kyriazis, a survey on volatility fluctuations in the decentralized cryptocurrency financial assets, J. Risk Financ. Manag. 14 (2021) 293.
[57] S. Lee, C.K. Kim, S. Lee, Hybrid CUSUM change point test for time series with time-varying volatilities based on support vector regression, Entropy 22 (2020) 578.
[58] S.F. Lehrer, T. Xie, G. Yi, Do the hype of the benefits from using new data science tools extend to forecasting extremely volatile assets? in: S. Consoli, D. Reforgiato Recupero, M. Saisana (Eds.), Data Science for Economics and Finance Springer, Cham, 2021.
[59] Y. Li, Predictability of volatility of cryptocurrencies, Bloom., Data Sci. /Quant. (2021).
[60] X. Lu, C. Liu, K.K. Lai, H. Cui, Risk measurement in bitcoin market by fusing LSTM with the joint-regression-combined forecasting model, Kybernetes (2023).
[61] R. Miura, L. Pichl, T. Kaizoji, Artfcial neural networks for realized volatility prediction in cryptocurrency time series, in: H. Lu, H. Tang, Z. Wang (Eds.), Advances in Neural Networks_ISNN, Springer, Cham, 2019, pp. 165–172.
[62] D.B. Nelson, C.Q. Cao, Inequality constraints in the univariate GARCH model, J. Bus. Econ. Stat. 10 (1992) 229–235.
[63] W. Orzeszko, Nonlinear causality between crude oil prices and exchange rates: evidence and forecasting, Energies 14 (19) (2021) 6043.
[64] A.M. Ozbayoglu, M.U. Gudelek, O.B. Sezer, Deep learning for financial applications: a survey, Appl. Soft Comput. 93 (2020), 106384.
[65] A.J. Patton, Volatility forecast comparison using imperfect volatility proxies, J. Econ. 160 (1) (2011) 246–256.
[66] Y. Peng, P.H.M. Albuquerque, J.M.C. de Sá, A.J.A. Padula, M.R. Montenegro, The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression, Expert Syst. Appl. 97 (2018) 177–192.
[67] S. Pong, M.B. Shackleton, S.J. Taylor, X. Xu, Forecasting currency volatility: A comparison of implied volatilities and AR(FI)MA models, J. Bank. Financ. 28 (10) (2004) 2541–2563.
[68] S.-H. Poon, C. Granger, Forecasting volatility in financial markets: a review, J. Econ. Lit. 41 (2003) 478–539.
[69] T.E. Pratas, F.R. Ramos, L. Rubio, Forecasting bitcoin volatility: exploring the potential of deep learning, Eur. Econ. Rev. (2023) 285–305.
[70] Y. Qiu, Complete subset least squares support vector regression, Econ. Lett. 200 (2021), 109737.
[71] J.B. Ramsey, Tests for specification errors in classical linear least-squares regression analysis, J. R. Stat. Soc. B 31 (2) (1969) 350–371.
[72] Rodikov G., Antulov-Fantulin N., (2022), Can LSTM outperform volatility-econometric models? arXiv:2202.11581v1.
[73] F. Sabry, W. Labda, A. Erbad, Q. Malluhi, Cryptocurrencies and artificial intelligence: challenges and opportunities, IEEE Access 8 (2020) 175840–175858.
[74] M. Seo, G. Kim, Hybrid forecasting models based on the neural networks for the volatility of bitcoin, Appl. Sci. 10 (14) (2020) 4768.
[75] O.B. Sezer, M.U. Gudelek, A.M. Ozbayoglu, Financial time series forecasting with deep learning: a systematic literature review: 2005–2019, Appl. Soft Comput. 90 (2020), 106181.
[76] Z. Shen, Q. Wan, D.J. Leatham, Bitcoin return volatility forecasting: a comparative study between GARCH and RNN, J. Risk Financ. Manag. 14 (7) (2021) 337.
[77] S. Smyl, G. Dudek, P. Pełka, Contextually enhanced ES-dRNN with dynamic attention for short-term load forecasting, Neural Networks 169 (2024) 660–672.
[78] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodol.) 58 (1) (1996) 267–288.
[79] A. Urquhart, B. Lucey, Crypto and digital currencies – nine research priorities, Nature 604 (2022) 36–39.
[80] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer,, Berlin/ Heidelberg, 1995.
[81] V.N. Vapnik, S. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems 9, MIT Press, Cambridge, MA, 1997, pp. 281–287.
[82] F. Violante, S. Laurent, Volatility forecasts evaluation and comparison, in: L. Bauwens, C. Hafner, S. Laurent (Eds.), Handbook of Volatility Models and Their Applications, John Wiley & Sons, 2012, pp. 465–486.
[83] Yu J., Weng Y., Rajagopal R., (2017), Mapping rule estimation for power flow analysis in distribution grids, arXiv:1702.07948.
[84] M. Zahid, F. Iqbal, D. Koutmos, Forecasting bitcoin volatility using hybrid GARCH models with, Mach. Learn., Risks 10 (12) (2022) 237.