Tea Šestanović *

JEL classification: C32, C45, C53
Original scientific paper
https://doi.org/10.32910/ep.75.1.3

A COMPREHENSIVE APPROACH TO BITCOIN FORECASTING USING NEURAL NETWORKS **

This paper provides a comprehensive approach to Bitcoin price, returns, direction and volatility forecasting. It compares ARIMA and GARCH models to neural network (NN) autoregression and Jordan NN in their forecasting performances, using internal and external factors. Robustness of the results is verified across bearish, bullish and stable market conditions. The results are not unambiguous considering price, returns or volatility forecasting, when compared using different performance measures or through different periods. Return and volatility forecasting yields to stable results no matter the model or period observed. NNs in general emerge as optimal for return and direction forecasting, ARIMAX and NNARX for price forecasting, while for volatility forecasting all models yield comparable results. Price forecasting yields the best prediction accuracies, while JNNX performed poorly. However, the inclusion of other machine learning methods and/or different variables as well as recent crisis emerged from war circumstances can be seen as limiting factors.

Keywords: *ARIMA, Bitcon, COVID-19, GARCH, Jordan neural network, neural network autoregression*

1. INTRODUCTION

Cryptocurrencies have attracted attention since the moment of their appearance, and especially in the last few years after their first price peak in 2018 and even more after their second and even higher peak in 2021. Numerous research confirmed their feature of being an investment asset class (Liu and Tsyvinski,

---

* T. Šestanović, Ph.D., Assistant Professor, University of Split, Faculty of Economics, Business and Tourism (e-mail: tea.sestanovic@efst.hr). The paper was received on 30.06.2022. It was accepted for publication on 20.12.2022.

** This work is supported by the Croatian Science Foundation (CSF) under Grant [IP-2019-04-7816].

2018; Stensås et al., 2019; Gil-Alana et al., 2020). Additionally, Platanakis and Urquhart (2020) and Li et al. (2021), among the others, proved their contribution in terms of Markowitz diversification to portfolios including also other traditional and/or alternative assets. Moreover, its role as a hedge and/or safe heaven has been intensively analysed. The results indicate that cryptocurrencies can serve primarily as a diversifier (Corbet et al., 2018; Stensås et al. 2019; Bouri et al., 2020). However, some empirical research found their hedge and/or safe haven properties in short investment horizons, for some specific assets or markets (Arnerić and Mateljan, 2019; Stensås et al. 2019; Bouri et al., 2020).

Additionally, there are research trying to predict either cryptocurrency prices (Indera et al., 2017; Poyser, 2017; Sovbetov, 2018; Jang and Lee, 2017; Fahmi et al., 2018; Lahmiri and Bekios, 2019; Ji et al., 2019; Uras et al., 2020; Pabuçcu et al., 2020), volatility (Walther et al., 2019), returns (Polasik et al., 2015; Abu Bakar and Rosbi, 2017; Liu and Tsyvinski, 2018; Azari, 2019) or direction (Greaves and Au, 2015; Spilak, 2018; Ji et al., 2019). Nevertheless, the research confirmed that the dynamics of cryptocurrency time series are quite complex, displaying extreme observations, asymmetries, and several nonlinear characteristics that are difficult to model and forecast.

This paper relies on findings documented in Šestanović (2021) where feed-forward neural networks (FNNs) and logistic regression (LR) for Bitcoin direction forecasting, i.e. predicting whether the prices will go up or down in the next trading day, are compared. Different internal and external factors mutually led to high and stable accuracy of 62% no matter the period. Therefore, this paper uses the variables extracted from Šestanović (2021) as the best predictors. On the other hand, it uses a comprehensive approach to find the appropriate model for price, returns and volatility forecasting. It compares models using root mean squared error (RMSE) and accuracy performance measures to reach the conclusion regarding the optimal methodology for Bitcoin prediction.

Additionally, most commonly used ARIMA and GARCH models for prices/returns and volatility forecasting are compared to NNs. Namely, due to unfulfilled assumptions of cryptocurrency time series, i.e. non-normality and nonlinearity, using linear parametric models is not suitable and leads to misinterpretation of the influence of certain variables as well as inability to predict properly the dependent variable. Therefore, NNs are viewed as nonlinear models with relaxed model assumptions. FNNs are the most commonly used with a property of "the universal approximator of any functional form of relationship between the observed variables" (Hornik et al., 1989). They can estimate the dependent variable (output) with a high degree of accuracy, enabled by their flexible but complex structure given the number of independent variables (inputs), hidden layers and neurons, estimated parameters, activation functions, learning algorithm and goal function. This paper presents and compares NNs for time series modelling and forecasting, i.e. NN autoregression (NNAR) and Jordan NN (JNN). Most of NN structure is held constant while hidden neurons are varied to obtain the optimal NN which simultaneously has high in-sample and out-of-sample performances via both RMSE and accuracy measures.

Previous research included only bullish or stable market conditions, while the predictions in bearish markets and especially in crisis periods were not tested. Therefore, this paper emphasises the comparison between different periods and market conditions, including the COVID crisis. Moreover, the chosen sub-periods are not selected arbitrary. Since arbitrary selection might lead to spurious results, the Bai-Perron multiple structural break test is utilized to determine the appropriate sub-periods. They coincided with different bullish, bearish and stable market conditions.

This paper contributes to the existing literature in several ways. Firstly, by defining the appropriate NN model for Bitcoin price, returns and volatility forecasting which will be comparable to ARIMA and GARCH models as benchmarks to test which of the models yields better predictive performances in terms of both RMSE and accuracy. Secondly, by determining whether prediction in terms of price, return or direction yield the optimal results. Thirdly, by determining the optimal volatility estimator for NN model as well as in terms of forecasting the volatility of cryptocurrencies using NNs. Finally, by comparing the results through different periods based on nonarbitrary selection of sub-periods using Bai-Perron multiple structural break test and by using Diebold-Mariano test of predictive accuracies. Different periods included bearish, bullish and stable market conditions, as well as including the recent COVID crisis.

The remainder of the paper is organized as follows. Section 2 provides literature review, Section 3 describes the data and methodology. Section 4 presents empirical findings with discussion of the results. Finally, conclusions and directions for future research are provided in Section 5.

2. LITERATURE REVIEW

Although the application of NNs in time series analysis is in its initial stage, they are proven an excellent tool for time series forecasting. Khashei and Bijari (2010) demonstrate that out of 96 studies, only in 18% of the cases traditional methods outperformed NNs while NNs have either performed well or outperformed in 72% of the cases. That is the reason of their immense application in forecasting cryptocurrencies as well. Although they are extensively used for classification purposes, this paper deals with their prices, returns and volatility forecasting.

Using ARIMA models for Bitcoin forecasting led to high prediction errors (Azari, 2019; Abu Bakar and Rosbi, 2017) since they are unable to capture sharp fluctuations in prices. High prediction errors are reported for ARIMA models in McNally (2018) who predicts Bitcoin and reaches error percentages of the RNN, ARIMA, and LSTM models of 5.45%, 53.47%, and 6.87% respectively, i.e. RNNs outperformed other linear and nonlinear models. Although, when used for short-term predictions or in sub-periods in which the behaviour of the time series is almost unchanged, ARIMA models can be an efficient tool (Azari, 2019).

Most research used NNs to predict cryptocurrency prices or returns. Indera et al. (2017) demonstrate the ability of Non-Linear Autoregressive with Exogenous Inputs (NARX) to predict Bitcoin using OHLC prices, together with Moving Average (MA) technical indicators of different intervals. However, without comparison to other models.

Further research compares different models to NNs. Fahmi et al. (2018) predict Bitcoin prices using internal factors while comparing Linear Regression (LR), NNs, Bayesian LR and Boosted Decision Tree Regression. The results indicate that regression-based models yield more usable predictions. However, they neither explain dataset where testing of the model is performed, nor provide detailed explanation of the methodology. Jang and Lee (2017) show that Bayesian NN performs better, compared to linear and nonlinear models, in predicting Bitcoin price and explaining its high volatility, using internal and external predictors. Lahmiri and Bekios (2019) implement LSTM and generalized regression NN (GRNN) to forecast the prices of Bitcoin, Digital Cash and Ripple. LSTM predictability is significantly higher compared to GRNN. Uras et al. (2020) forecast prices of Bitcoin, Litecoin and Ethereum, using lagged OHLC prices and volumes while implementing Simple and Multiple LR, as well as FNN and LSTM models. The best results are obtained using more than one previous price and with both regression models and LSTM, while NNs performed poorly. However, they used only in-sample comparisons and it is not unusual for LR to perform well in-sample. The model should always be tested out-of-sample, i.e. part where NNs perform better. Ji et al. (2019) compare deep NN (DNN), LSTM, convolutional NN, deep residual network, and their combinations as well as support vector machine (SVM), gated recurring unit (GRU) and linear/logistic regression for Bitcoin price prediction. GRU and linear/logistic regression models performed worse or equal to SVM. The results show that LSTM slightly outperformed other models for Bitcoin price prediction. DNN performed the best for price direction prediction. They used internal factors to predict Bitcoin prices and concluded that 20 inputs are sufficient for regression and 50 inputs for classification purposes. However, they used random sample splitting instead of sequential which is not justifiable from econometric perspective, as well as too many inputs. Dutta et al. (2020) used a fixed set of internal and external factors as exogenous and endogenous variables to predict daily Bitcoin prices and show that GRU model performs better than traditional NNs and LSTM. Moreover, RNN and LSTM perform better than traditional time series models in cryptocurrency price prediction. Additionally, Chen et al. (2020) predict Bitcoin price at different frequencies (i.e. daily and high-frequency). LR and Discriminant Analysis (DA) for Bitcoin daily price prediction with high-dimensional features achieve an accuracy of 66%, outperforming more complicated models. Random Forest, XGBoost, Quadratic DA, SVM and LSTM for Bitcoin 5-minute interval price prediction are superior to statistical methods, with accuracy reaching 67.2%. Jalali and Heidari (2020) predict the price of Bitcoin using the first order grey model (GM (1,1)). GM outperformed RNN and BNN, however they never explained how they are compared considering different methodology.

On the other hand, research on cryptocurrency volatility prediction is limited while relying only on high frequency data (Zhang et al., 2021) or only on focusing on GARCH-type methodology (Chu et al., 2017; Walther et al., 2019). The research on stock market volatility prediction indicates the usefulness of NNs in volatility prediction in comparison to other linear and nonlinear models. Donaldson and Kamstra (1997) reveal that NNs capture volatility effects overlooked by GARCH-type models. Mantri et al. (2010) conclude that there is no difference in the volatilities estimated under the GARCH, EGARCH, GJR - GARCH, IGARCH and NN models. In their later paper, Mantri et al. (2012) concluded that NN can be used as a best choice for measuring the volatility of stock market., compared to ARCH and GARCH models. Sarangi and Dublish (2013) prove that NN is ranked best with minimum forecasting error, compared to GARCH family models. Arnerić et al. (2014) compared GARCH to JNN models in forecasting the conditional variance of stock returns and confirm superiority of NNs versus other linear and nonlinear models. However, Nybo (2021) indicates that the NNs should be used for predicting volatility of assets with low volatility profiles, and GARCH models should be used when predicting volatility of medium and high volatility assets. Due to the lack of research on cryptocurrency volatility prediction with NNs, this paper can contribute to the existing research.

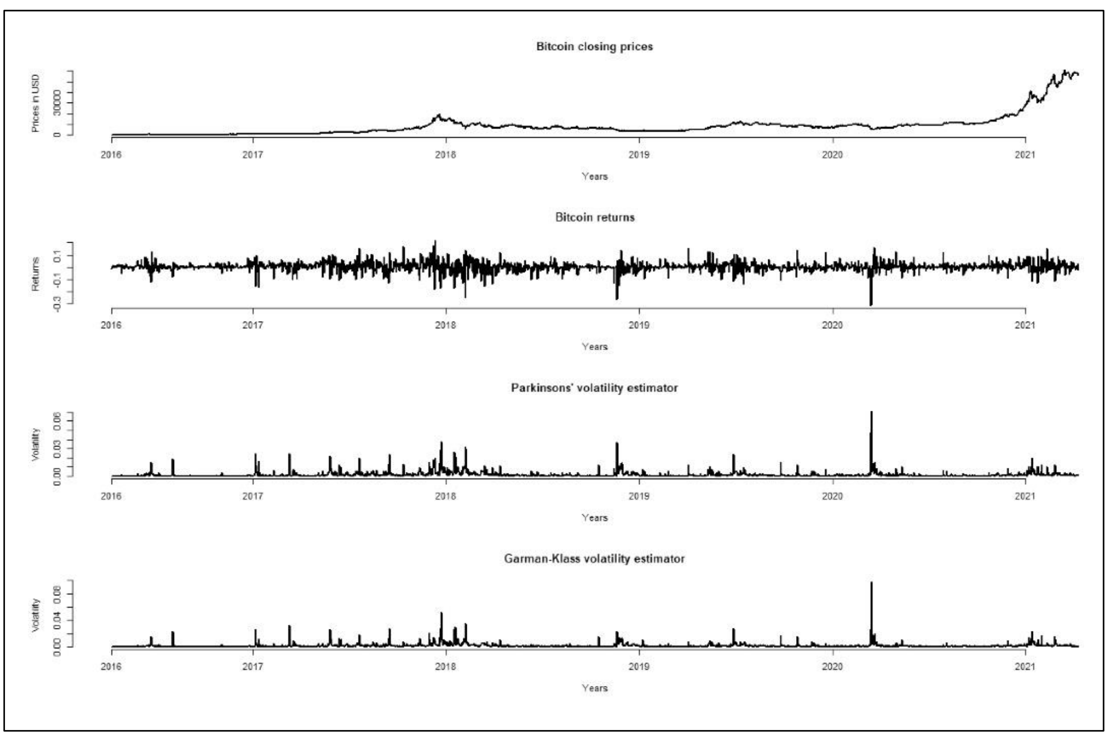
3. DATA AND METHODOLOGY

The Bitcoin daily closing prices, returns and volatilities obtained with Parkinsons' and Garman-Klass volatility estimators are given in Figure 1. From the appearance of Bitcoin in 2014 to 2017 the market was characterised by a sluggish movement of closing prices, as well as market capitalization and volumes.

Therefore, the period of monotonous price movements and trading from their appearance to 2016 is excluded from further calculations. It is more interesting and challenging to test the proposed model in more volatile periods. After a significant peak at the end of the 2017 the prices dropped sharply at the beginning of the 2018, followed by the intensive trading period characterized by high volatility. Continuous fluctuations in Bitcoin prices remained until the end of 2019 but with rather stable market conditions with both low returns and volatility. Beginning of the year 2020 was characterised by a significant slump in prices due to the unknown circumstances of COVID crisis and high volatility. However, the market recovered fast and experienced a significant upward trend in the 2020 that continued in 2021, maybe mostly because of its potential to hedge against inflation because of an increased government spending during the COVID pandemic but also because some of the companies announced using Bitcoin, instead of cash.

Figure 1

BITCOIN PRICES, RETURNS AND VOLATILITIES WITH PARKINSONS' AND GARMAN-KLASS VOLATILITY ESTIMATORS FROM APRIL 2016 TO APRIL 2021



Based on different features of price, returns and volatility movements in different periods the dataset is divided into four sub periods. The selection is not made arbitrary, but using the Bai-Perron (Bai and Perron, 2003) test for simultaneous estimation of multiple breakpoints. Based on Bitcoin closing prices in period from 9.4.2016 to 9.4.2021 four breaks are selected. The first sub period training set spans from 9.4.2016 to 27.11.2017, while the next 100 observations, i.e. until 7.3.2018 are used for testing. That way the prediction in the bearish market is considered. The next training set spans from 9.4.2016 to 24.12.2018 and the next 100 observations, i.e. until 3.4.2019 are left for testing. That way the prediction in stable market is considered. The third sub period for training is selected from 9.4.2016 to 24.9.2019 and the next 100 observations, i.e. 2.1.2020 are selected for testing. Finally, the data set from 9.4.2016 to 9.7..2020 is used for training and the next 100 observations, i.e. until 17.10.2020 are used for testing. That way the prediction in the bullish market is considered.

Bitcoin closing prices are retrieved from Coinmetrics (charts.coinmetrics.io/network-data) and they are used to calculate the log returns. Closing prices and log returns are given in Figure 1 in the two upper panels. However, for volatility forecasting, the problem was finding the optimal volatility estimator as the dependent variable in NNs to be comparable to volatility obtained from GARCH model. Although most of the empirical research indicate Garman and Klass (1980) as the optimal OHLC (open-high-low-close or range-based) volatility estimator, Raju and Rangaswamy (2017) prove the dominance of Yang and Zhang (2000) estimator in terms of in-sample and out-of-sample forecasting accuracy. Yarovaya et al. (2016), on the other hand, report inconclusive outcomes when examining the Garman and Klass (1980), Parkinson (1980), and Rogers and Satchell (1991) estimators. Additionally, Arnerić et al. (2018) provide evidence that in most emerging economies the accuracy of Garman and Klass estimator does not significantly differ from Parkinson. Therefore, since there is no consensus in the literature about the appropriate OHLC volatility estimator, Parkinson's and Garman-Klass volatility estimators are selected and compared as an additional robustness check. They are calculated as:

$$PA_t = \frac{1}{4\ln 2}\left(\ln\frac{H_t}{L_t}\right)^2 \tag{1}$$

where PA_t is the Parkinson's volatility estimator at time t, while H_t and L_t are the high and low prices of Bitcoin at time t.

$$GK_t = \frac{1}{2}\left(\ln\frac{H_t}{L_t}\right)^2 - (2\ln 2 - 1)\left(\ln\frac{C_t}{O_t}\right)^2 \tag{2}$$

where GK_t is the Garman-Klass volatility estimator at time t, while additionally C_t and O_t are the closing and opening prices of Bitcoin at time t. Both volatility estimators are given in Figure 1 in the lowest two panels.

The independent variables selection was not straightforward either. Poyser (2017) distinguishes between internal and external factors as cryptocurrency price drivers. Supply and demand are among the others the main internal factors while attractiveness, market trend, speculations, legalization, restrictions and macro-finance factors are external drivers. Most of the papers use the cryptomarket-related or internal factors for prediction (Polasik et al., 2015; Sovbetov, 2018; Liu and Tsyvinski, 2018; Jang and Lee, 2017; Spilak, 2018; Fahmi et al., 2018; Ji et al., 2019). Indera et al. (2017), Fahmi et al. (2018) and Uras et al. (2020) in particular use OHLC prices while Azari (2019) and Abu Bakar and Rosbi (2017) use only past closing prices. Technical indicators are also used as predictors (Indera et al., 2017; Spilak, 2018; Pabuçcu et al., 2020). Moreover, majority of the papers confirm the attractiveness as an important factor (Polasik et al., 2015; Sovbetov, 2018). Few papers use macro-finance factors (Polasik et al., 2015; Sovbetov, 2018; Liu and Tsyvinski, 2018; Spilak, 2018) and usually report the lack of statistical significance if used in parametric models. Contrary, Walther et al. (2019) found that economic activity is the most important exogenous volatility driver. This paper, however, uses the internal and external factors extracted from Šestanović (2021) for closing price and return forecasting with NNs and ARIMAX models. Namely, NNs and ARIMAX have besides lagged returns also 10 inputs that had high accuracy in the most models. The internal factors, i.e. market capitalization, 30-days volatility, total issuance, mean tx fee, mean hash rate, mean difficulty are retrieved from: Coinmetrics (charts.coinmetrics.io/network-data). The macro-financial data (gold, S&P500, VIX) are obtained from FRED database (fred.stlouisfed.org/) and attractiveness from Google trends (trends.google.com/trends/). All the variables are transformed to log-returns to become stationary, since NNs perform worse when dealing with nonstationary time series. Ji et al. (2019) confirm that log values of variables with extreme values yields better performances than using plain values. Only attractiveness is not transformed using log-returns since it is available only at weekly basis and that transformation would yield to lots of zeros. For volatility forecasting and based on GARCH methodology, the Parkinson's or Garman-Klass volatility estimator from previous period along with squared mean corrected returns with one-time lag, representing squared innovations, is used in NNs as independent variables.

The most commonly used NN is FNN. Each input (independent variable) of FNN is connected with all neurons in the hidden layer. Based on the cross-product of input values and corresponding weights, the hidden neurons perform nonlinear transformation using the activation function. Each neuron in the hidden layer is further connected with each neuron in the next layer. In a three-layered FNN the next layer is the final, called the output layer. In the output layer the expected values of the dependent variable are estimated. This represents the feedforward stage. Expected values, i.e. outputs from the NN are compared to the observed, i.e. target values and their difference represents the error term, i.e. residual. Error terms are used to adjust the network parameters in the backpropagation (BP) stage, i.e. BP learning algorithm is used for parameter correction until a minimum error is achieved. FNN, designed for the time series modelling and forecasting, is NN autoregression (NNAR(p,k)) which has p lags of dependent variable as inputs and k hidden neurons. If only one lag of dependent variable is used as input and additional exogenous inputs are added to the model, the resulting NNARX(1,k,m) has the following form:

$$y_t = f\left(\phi_{co} + \sum_{h=1}^{k} \phi_{oh} f\left(\phi_{ch} + \phi_1 y_{t-1} + \lambda_t^T x_t\right)\right) + \varepsilon_t, \tag{3}$$

where y_t is the output vector of a time series presenting dependent variable (in our case Bitcoin prices or log returns), y_{t-1} is dependent variable with one lag, x_t is the matrix with m additional exogenous inputs, while $f(\cdot)$ is the logistic activation functions. The weights w_{co} and w_{ch} denote constant terms of output and hidden neurons respectively, where $h=1,2,...,k$. The weights w_{ih} and w_{ho} denote the connections between the p-inputs and h-hidden neurons and between the h-hidden neurons and the output respectively, while ε_t is an error term. *nnetar* function from *forecast* package in *R* software by default choses number of lags of dependent variable and number of hidden neurons. To get comparable results, the number of inputs is set in advance and already explained, while the grid search of 2, 5, 10, 15, 25 and 50 hidden neurons is examined to be comparable to Šestanović (2021).

If the feedback connection from outputs to inputs is added to a NNARX, the resulting is a recurrent-type of NNs, i.e. Jordan NN with exogenous inputs (JNNX):

$$y_t = f\left(\phi_{co} + \sum_{h=1}^{k} \phi_{oh} f\left(\phi_{ch} + \phi_1 y_{t-1} + \lambda_t^T x_t + \phi_{rh}\varepsilon_{t-1}\right)\right) + \varepsilon_t, \tag{4}$$

where w_{rh} represents the weight of the recurrent connection, i.e. connection of the context unit to hidden neurons. JNNX given in eq. (3) incorporates AR term (y_{t-1}), MA term (ε_{t-1}) and different internal and external factors as exogenous inputs. This can be seen as ARIMAX model with one hidden layer and appropriate activation functions. Analogously to NNARX model, since JNNX has additional input, i.e. error term with one lag, it can be abbreviated as JNNX(1,k,1,m). Additional self-connected neuron keeps the content of the output that existed in the previous network training, which is called the context unit. It represents the long-term memory of the network. JNNX uses a recursive algorithm that is similar to the standard BP algorithm but requires multiple equations for weights corrections. It also provides consistent and asymptotically normal estimators (Kuan and Liu, 1995). The premise is that JNNX can bypass the overfitting problem with less parameters to estimate and at the same time satisfy the parsimony principle. In order to train JNNX, the "in-sample" is additionally divided into 70% for training and 30% for validation. *Ad hoc* settings include: initial functions randomization, the standard BP algorithm, the functions are updated in topological order, the patterns are presented in sequential order, the maximal number of iterations is 10000, learning rate is 0.0001, intervals from which the initial network weights are randomly selected is [-1, 1]), the context unit weight is set to 0.7 and the logistic and linear activation functions are used in hidden and output layer respectively. All variables are transformed using normalization function to enable the convergence stability of the learning algorithm. After training, denormalization of the output is performed to be comparable to the actual output values. The number of hidden neurons (*k*) is varied in a grid search of 2, 5, 10, 15, 25 and 50 hidden neurons as in NNARX model. JNNXs are estimated using *jordan* function in *RSNNS* package (*R – Stuttgart Neural Network Simulator*). The results of JNNX are compared to similar models, i.e. ARIMAX and NNARX models.

ARIMAX models are estimated in *R* using the *auto.arima* function in package *forecast*. This function returns the best ARIMAX model according to either AIC, AICc or BIC value. ARIMAX model has therefore the same representation as in eq. (3) if there is no hidden layers and activation function is linear.

FNNs and LR for direction forecasting from Šestanović (2021) are reestimated using Bai-Perron breaks to compare the results of returns, closing prices and direction forecasting.

The standard GARCH (1,1) models, defined by Bollerslev (1986), are estimated in *R* using *rugarch* package. For results to be comparable, NNAR and JNN models are estimated using previously explained models in eq. (2) and (3) without any additional exogenous inputs, where y_t in this case are PA_t and GK_t - the Parkinson's and Garman-Klass volatility estimators at time *t* respectively, and independent variables are volatility estimators from previous period along with squared mean corrected returns with one time lag.

All models are compared through out-of-sample RMSE, i.e.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(y_t - \hat{y}_t\right)^2}\,, \tag{5}$$

where y_t and $\hat{y}_t$ are the observed and predicted output values respectively and n is the sample size. Additionally, Diebold-Mariano (DM) test (Diebold and Mariano, 1995) for predictive accuracy for each sub-period and each model in pairs is performed for 1 and 7 days forecasting horizon. It formally compares RMSEs. However, other than standard comparison of the models using RMSEs, the accuracy measure is calculated. The accuracy is a proportion of true positive (TP) and true negative (TN) in the whole sample n, i.e.

$$ACC = \frac{TP + TN}{n}, \tag{6}$$

where TP are correctly predicted positive returns and NP correctly predicted negative returns, i.e. accuracy measures the models' ability to correctly predict direction of Bitcoin movement. It is especially important in comparison between models that have different units of measurement, where comparison using RMSE is not possible.

4. RESULTS AND DISCUSSION

Results of ARIMAX, NNARX and JNNX models for one day ahead return and closing prices forecasting are given through 2 measures: RMSE and accuracy in Tables 1 and 2 respectively. Additionally, Table 3 presents the reestimated accuracies from Šestanović (2021) with FNNs and Logistic regression using Bai-Perron breaks. GARCH, NNAR and JNN models for one day ahead volatility forecasting using both Parkinson's and Garman-Klass volatility estimators are compared only according to RMSE (Table 4). Results are presented for 4 selected sub-periods. Only out-of-sample performance measures are given since all the models performed well in-sample[1]. Additionally, DM test for predictive accuracy for each sub-period and each model in pairs for 1 and 7 days' prediction is given in the lowest panel of the Tables 1, 2 and 4.

[1] In-sample results are available from the author upon reasonable request.

Table 1

OUT-OF-SAMPLE RMSE, DM TEST AND ACCURACY MEASURES FOR 4 SELECTED SUB-PERIODS AND BITCOIN RETURNS

	RMSE				ACCURACY			
NNARX (k)	1	2	3	4	1	2	3	4
2	0.0773	0.0305	**0.0329**	0.0228	0.57	**0.49**	0.43	0.60
5	**0.0772**	0.0305	0.0331	0.0229	**0.63**	**0.49**	0.38	0.61
10	0.0772	0.0313	0.0343	0.0230	0.60	0.46	0.36	0.55
15	0.0757	**0.0304**	0.0339	**0.0226**	**0.63**	0.48	0.39	**0.63**
25	0.0783	0.0327	0.0352	0.0230	0.60	0.44	0.41	0.52
50	0.0802	0.0353	0.0378	0.0254	0.55	0.44	0.42	0.52
JNNX (k)	1	2	3	4	1	2	3	4
2	0.0774	0.0417	**0.0337**	**0.0287**	0.58	0.51	0.48	0.45
5	**0.0767**	0.0416	0.0357	0.0297	0.54	0.45	0.44	0.51
10	0.0802	**0.0396**	0.0387	0.0320	0.56	0.53	**0.50**	0.54
15	0.0781	0.0421	0.0370	0.0382	0.58	**0.59**	0.46	0.46
25	0.0797	0.0453	0.0418	0.0423	0.56	0.55	0.45	0.46
50	0.0814	0.0479	0.0476	0.0436	**0.60**	0.58	0.43	**0.64**
	1	2	3	4	1	2	3	4
ARIMAX	0.0855	0.0374	0.0407	0.0321	0.60	NA[1]	0.50	0.44
OPTIMAL	JNNX	NNARX	NNARX	NNARX	NNARX	JNNX	JNNX	JNNX
	DM h=1				DM h=7			
NNARX*JNNX	0.31	2.68***	0.65	3.49***	0.41	2.19**	0.62	3.33***
NNARX*ARIMAX	1.61	3.18***	1.77*	4.44***	1.77*	3.92***	2.18**	2.81***
JNNX*ARIMAX	1.51	0.62	1.76*	1.93*	1.74*	0.66	1.98*	1.44

Note: Sub-period defined based on Bai-Perron test, i.e. 1 - training (9.4.2016 - 27.11.2017), testing (next 100 observations, i.e. until 7.3.2018; 2 - training (9.4.2016 - 24.12.2018), testing (next 100 observations, i.e. until 3.4.2019); 3 – training (9.4.2016 - 24.9.2019), testing (next 100 observations, i.e. until 2.1.2020); 4 – training (9.4.2016 - 9.7.2020), testing (next 100 observations, i.e. until 17.10.2020). k – number of hidden neurons. [1] The model forecasted only positive returns. h – forecasting horizon of DM test of 1 and 7 days. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$, where p is the p-value of DM test.

Source: the author in R

Forecasting Bitcoin returns (Table 1) yields the most stable results no matter the model and period observed. Namely, time series models as well as NNs perform the best when dealing with stationary time series. According to RMSE for return forecasting linear ARIMAX model performed the worse no matter the period, although relatively comparable to JNNX in stable period. In turbulent times, no matter the bearish or bullish period, both NN types were successful in predicting Bitcoin returns, although with lower mistakes made in stable and bullish periods. Even though NNARX performed better in 3 out of 4 periods, according to RMSE, JNNX performed comparably well. DM test shows that NNARX model has significantly lower RMSEs than ARIMAX in all sub-periods, while it is significantly better than JNNX in only two sub-periods. JNNX performed significantly better than ARIMAX in 3 out of 4 sub-periods. However, the highest accuracy of 64% is reached with JNNX in bullish period, although 63% accuracy reached by NNARX in the same period is not unneglectable. In bearish period NNARX model had the highest accuracy of 63%, in stable market conditions in period 2 JNNX reached accuracy of 59%, while in period 3 all models performed poorly. With respect to the number of hidden neurons in terms of RMSE performance measure, JNNX model confirms the ability of better or comparable results to NNARX with lower number of hidden neurons (Šestanović and Arnerić, 2020).

Table 2

OUT-OF-SAMPLE RMSE, DM TEST AND ACCURACY MEASURES FOR 4 SELECTED SUB-PERIODS AND BITCOIN CLOSING PRICES

	RMSE				ACCURACY			
NNARX (k)	1	2	3	4	1	2	3	4
2	**4802.24**	851.54	**710.95**	2103.35	0.485	0.525	0.495	0.420
5	5715.68	851.54	920.55	2020.74	0.485	0.566	0.596	0.455
10	6036.94	**782.38**	1242.12	2024.83	0.424	**0.586**	**0.657**	0.455
15	5611.35	898.32	977.31	**1806.52**	**0.535**	0.556	0.566	0.444
25	5906.1	1970.2	1212.4	1905.4	0.485	0.515	0.636	0.455
50	6070.21	798.59	1146.05	1976.06	0.444	0.535	0.556	**0.505**
JNNX (k)	1	2	3	4	1	2	3	4
2	10495.8	3827.9	4339.1	5973.7	0.455	0.384	0.455	0.424
5	10482.6	3827.9	4385.9	5931.1	0.455	0.404	0.465	0.485
10	10502.3	3694.0	4335.2	5932.3	0.444	**0.465**	**0.525**	0.444
15	10499.3	3620.9	**4307.3**	5935.7	0.434	0.444	0.525	0.505
25	10512.4	3720.2	4392.0	**5863.9**	**0.465**	0.455	0.475	0.455
50	**10466.6**	**3620.6**	4333.5	5876.5	0.455	0.434	0.515	**0.515**
	1	2	3	4	1	2	3	4
ARIMAX	3696	360	1704	1641	0.455	0.576	0.515	0.606
OPTIMAL	ARIMAX	ARIMAX	NNARX	ARIMAX	NNARX	NNARX	NNARX	ARIMAX
	DM h=1				DM h=7			
NNARX*JNNX	57.36***	9.55***	9.58***	8.91***	28.61***	4.01***	2.87***	2.62**
NNARX*ARIMAX	11.45***	5.36***	9.78***	3.23***	3.42***	1.55	2.74***	0.91
JNNX*ARIMAX	52.99***	9.61***	9.31***	9.06***	22.06***	3.93***	2.88***	2.65***

Note: Sub-period defined based on Bai-Perron test, i.e. 1 - training (9.4.2016 - 27.11.2017), testing (next 100 observations, i.e. until 7.3.2018; 2 - training (9.4.2016 - 24.12.2018), testing (next 100 observations, i.e. until 3.4.2019); 3 – training (9.4.2016 - 24.9.2019), testing (next 100 observations, i.e. until 2.1.2020); 4 – training (9.4.2016 - 9.7.2020), testing (next 100 observations, i.e. until 17.10.2020). k – number of hidden neurons. h – forecasting horizon of DM test of 1 and 7 days. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$, where p is the p-value of DM test.

Source: the author in R

Forecasting Bitcoin prices (Table 2) reaches the highest accuracy of 65.7% with NNARX model in period 3 which included Covid crisis, followed by 60.6% accuracy obtained with ARIMAX in bullish market. The model with highest accuracy in bearish and stable market is NNARX (53.4% and 58.6% respectively). However, NNARX was the optimal model in 3 out of 4 periods according to accuracy measure, while ARIMAX model performs better in bullish period. JNNX performed poorly compared to NNARX and ARIMAX, with both performance measures in all sub-periods[2]. Namely, forecasting closing prices with JNNX led to rather high RMSEs no matter the period and no matter the hidden neuron selection. It is not clear why JNNX performed poorly in terms of RMSE for the Bitcoin price prediction as opposed to returns forecasting since most of the empirical research confirms the ability of recurrent NNs to successfully predict cryptocurrency prices (McNally, 2018; Dutta et al., 2020). Possible explanation for this is that JNNX rather successfully predicts the movement of Bitcoin prices but with significant volatilities. On the other hand, ARIMAX model emerges as optimal although its predictions are close to the average and predicts mostly a flat line. Therefore, these results should be taken with caution.

According to RMSE, ARIMAX model for closing price prediction is the optimal in 3 out of 4 sub-periods. NNARX model is optimal in only period 3, although according to DM test it performs comparably well to ARIMAX model in bullish market. Azari (2019) confirms efficiency of ARIMA model for Bitcoin prediction in stable and unchanged periods, as opposed to predictions in turbulent events when they lead to large prediction errors (Azari, 2019; Abu Bakar and Rosbi, 2017). However, result of this research confirms the opposite, i.e. the success of ARIMAX model in closing price predictions in bearish, bullish and stable market conditions. The possible explanation is that ARIMAX model was estimated on closing prices, but due to non-stationary time series the integration of order 1 was performed to reach stationarity. NNARX and JNNX models used raw data, i.e. non-stationary closing prices. The conclusion is that NNs do not predict well non-stationary time series.

---

[2] JNNX estimated for closing prices with standard parameters given in methodology part provided even worse results. The results are given for the following parameters: in-sample is divided into 90% for training and 10% for validation, learning rate is 0.001 and the context unit weight is set to 0.9.

Table 3

OUT-OF-SAMPLE ACCURACY FOR 4 SELECTED SUB-PERIODS AND BITCOIN DIRECTION PREDICTION

	ACCURACY			
FNN (k)	1	2	3	4
2	NA	NA	0,39	NA
5	0,57	0,54	0,44	0,52
10	0,57	**0,58**	0,39	**0,53**
15	0,48	0,48	0,33	0,52
25	0,54	0,50	**0,51**	0,49
50	**0,60**	0,48	0,50	0,51
	1	2	3	4
LR	0,59	0,49	0,40	0,61
OPTIMAL	FNN	FNN	FNN	LR

Note: Sub-periods defined based on Bai-Perron test, i.e. 1 - training (9.4.2016 - 27.11.2017), testing (next 100 observations, i.e. until 7.3.2018; 2 - training (9.4.2016 - 24.12.2018), testing (next 100 observations, i.e. until 3.4.2019); 3 – training (9.4.2016 - 24.9.2019), testing (next 100 observations, i.e. until 2.1.2020); 4 – training (9.4.2016 - 9.7.2020), testing (next 100 observations, i.e. until 17.10.2020). k – number of hidden neurons. 1 The model forecasted only positive or negative returns.

Source: the author in R

When compared to the Bitcoin direction forecasting in Table 3 the return forecasting yields to higher accuracies in all market conditions, except for the Period 3 where the difference is negligible. This is not in line with the previous research (Ji et al, 2019; Pabuçcu et al., 2020) that proved direction forecasting to be more effective.

On the other hand, forecasting Bitcoin closing prices led to much higher accuracy compared to Bitcoin direction forecasting in Table 3 in all the observed sub-periods, except for the first bearish period where direction accuracy was much higher. Although Chen et al. (2020) point that it is easier for traders to make decisions and follow direction forecasting and although it is shown to be more effective in several research (Ji et al, 2019; Pabuçcu et al., 2020), from this research one can reach a conclusion that forecasting prices and returns incorporates much more data and information than the simple direction forecasting.

Table 4

OUT-OF-SAMPLE RMSE AND DM TEST FOR 4 SELECTED SUB-PERIODS AND BITCOIN VOLATILITY WITH PARKINSON'S AND GARMAN-KLASS VOLATILITY ESTIMATORS

	Parkinson				Garman-Klass			
NNARX (k)	1	2	3	4	1	2	3	4
2	0.00827	**0.00312**	0.00916	0.00659	0.00798	**0.00206**	**0.00249**	**0.00123**
5	0.00833	0.00312	0.00847	**0.00256**	0.00926	0.00206	0.00256	0.00148
10	**0.00683**	0.00472	**0.00324**	0.00265	**0.00729**	0.00254	0.00273	0.00152
15	0.00886	0.00637	0.00341	0.00257	0.02600	0.00265	0.00290	0.00175
25	0.00895	0.00563	0.00385	0.00257	0.00809	0.00275	0.00284	0.00213
50	0.00904	0.00563	0.00485	0.00258	0.00907	0.00267	0.00305	0.00199
JNNX (k)	1	2	3	4	1	2	3	4
2	0.00807	0.00424	**0.00269**	0.00248	**0.00844**	0.00275	**0.00240**	**0.00202**
5	0.00811	0.00424	0.00372	0.00296	0.00854	0.00275	0.00477	0.00400
10	0.00827	0.00669	0.00404	0.00574	0.00873	**0.00261**	0.00258	0.00224
15	**0.00804**	**0.00217**	0.00341	**0.00161**	0.00865	0.00339	0.00505	0.00525
25	0.00811	0.00243	0.00326	0.00203	0.00857	0.00397	0.00426	0.00484
50	0.00813	0.00241	0.00291	0.00166	0.00852	0.00275	0.00338	0.00364
	1	2	3	4	1	2	3	4
GARCH	0.00694	0.00201	0.00326	0.00189	0.00906	0.00191	0.00349	0.00120
OPTIMAL	NNAR	GARCH	JNN	JNN	NNAR	GARCH	JNN	GARCH
	DM h=1							
NNAR*JNN	1.72*	10.73***	3.78***	9.42***	2.46***	2.23**	0.41	2.74***
NNAR*GARCH	0.22	8,66***	0.08	3.88***	1.69*	0.62	2.97***	0.28
JNN*GARCH	1.59	1.06	2.57**	1.24	0.51	2.51**	3.48***	2.69***
	DM h=7							
NNAR*JNN	1.83*	6.05***	3.80***	6.22***	2.72***	1.47	0.51	1.98**
NNAR*GARCH	0.23	4.72***	0.04	1.44	1.89*	0.34	1.59	0.13
JNN*GARCH	2.17**	0.70	1.13	0.53	1.13	2.04**	1.56	2.51**

Note: Sub-period defined based on Bai-Perron test, i.e. 1 - training (9.4.2016 - 27.11.2017), testing (next 100 observations, i.e. until 7.3.2018); 2 - training (9.4.2016 - 24.12.2018), testing (next 100 observations, i.e. until 3.4.2019); 3 – training (9.4.2016 - 24.9.2019), testing (next 100 observations, i.e. until 2.1.2020); 4 – training (9.4.2016 - 9.7.2020), testing (next 100 observations, i.e. until 17.10.2020). k – number of hidden neurons. h – forecasting horizon of DM test of 1 and 7 days. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$, where p is the p-value of DM test.

Source: the author in R

Volatility forecasting is obtained with NNAR, JNN and GARCH models, meaning that no additional exogenous inputs are used. The results of Parkinson's and Garman-Klass volatility estimators are compared to test the estimators and to verify robustness of the results (Table 4). Only in period 2 for Parkinson's volatility estimator is GARCH model significantly better than NNs. For Garman-Klass estimator GARCH is the optimal model in period 2 and 4, however it is significantly better than JNN, but performs comparably well to NNAR model. For Parkinson's volatility NNAR model is the optimal for bearish market and JNN in bullish market conditions as well in period 3 of relative stability. For Garman-Klass volatility NNAR model is also the optimal one for bearish market and JNN in bullish market conditions. However, GARCH models are not outperformed by the NN models, as they have comparable results if DM test is included. The conclusion regarding the stability of the results for volatility forecasting in general can be reached. Only the first sub-period yielded higher RMSEs which can be explained by relatively small sample size. The stability of the results no matter the estimator used is also confirmed.

This research partially confirms the finding of Mantri et al. (2010) that there is no difference in volatilities estimated with different GARCH-type and NN models. On the other hand, more research confirmed the ability of NNs in general to capture the volatility effects overlooked by GARCH-type models leading to better volatility forecasts (Donaldson and Kamstra, 1997; Mantri et al., 2012; Sarangi and Dublish, 2013; Arnerić et al., 2014). Namely, the results depend on the sub-period and the used NN model. Reasons behind twofold conclusions in literature and in this paper should be further investigated while considering other NN-types or modified structure of the proposed models as well as the inclusion of other exogenous inputs as in Walther et al. (2019). However, this can be explained by the distinct features of cryptocurrencies as opposed to the behaviour of stocks, currencies and precious metals (Liu and Tsyvinski, 2018), as well as because of their relative isolation from other financial and economic assets (Corbet et al., 2018). The conclusion regarding GARCH model is that the traders, financial analysts and economists may remain indifferent while choosing the model for the volatility estimation and forecasting. That could be explained by Nybo (2021) who indicates that the NNs should be used for predicting volatility of assets with low volatility profiles, and GARCH models should be used when predicting volatility of medium and high volatility assets like Bitcoin.

Finally, using lower number of hidden neurons in all the NN models is recommended, no matter the return, prices or volatility forecasting. NNs with up to 15 hidden neurons yield optimal results, while additional neurons lead to overfitting problem. Using forecasting horizon of 1 or 7 days in DM test does not lead to significant difference in the results and conclusions.

5. CONCLUSION

This paper provides a comprehensive approach to Bitcoin price, returns and volatility forecasting using NNAR and JNN while comparing them to the most commonly used benchmarks, i.e. ARIMA and GARCH models. It incorporates exogenous internal and external factors for return and price forecasting. For volatility forecasting plain models are tested. The results are compared using different performance measures while the robustness of the results is additionally verified across different periods, including bearish, bullish and stable market conditions. The results indicate using lower number of hidden neurons in all NN models yields better results, while forecasting horizon of 1 or 7 days does not lead to significant difference in the results and conclusions. However, the results are not unique and unambiguous when considering price, returns, direction or volatility forecasting, when comparing them using different performance measures and through different periods. Namely, forecasting Bitcoin returns and volatility yields stable results no matter the model and period observed. The exception is the first period with stable but relatively higher RMSEs due to the small sample size and bearish period On the other hand, forecasting Bitcoin prices yields unstable RMSEs but the best prediction accuracies. The worst results are obtained with ARIMAX model for return forecasting and with JNNX for price forecasting. On the other hand, due to non-stationary time series of closing prices, the NNs could not predict well as opposed to ARIMAX model that performed well due to incorporated integration in the model. Nevertheless, NNARX model was also successful in predicting Bitcoin prices. In turbulent times, no matter the bearish or bullish period, both NN types were successful in predicting Bitcoin returns, although NNARX performed significantly better according to DM test in bullish market. When compared through accuracy measure, the results indicate opposite, i.e. dominance of JNNX in bullish and stable market, and dominance of NNARX in bearish market. In terms of accuracy for price prediction, NNARX outperformed the others in bearish and stable market conditions, while ARIMAX dominated in bullish period. No matter the estimator GARCH model can be seen as optimal in stable period, while NNAR model dominates in bullish market. JNN also performs well in stable pre Covid crisis periods for both estimators. However, the results among the models are comparable and not significantly different especially in stable and bullish periods. This calls for further investigation in terms of the inclusion of more exogenous volatility drivers.

REFERENCES:

1. Abu Bakar, N., & Rosbi, S. (2017) Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction, International Journal of Advanced Engineering Research and Science (IJAERS), 4(11), November, 130-137. https://dx.doi.org/10.22161/ijaers.4.11.20
2. Arnerić, J., & Mateljan, M. (2019). Analiza međuovisnosti tržišta kapitala i tržišta kriptovaluta. Ekonomska misao i praksa, No. 2, 449-465. https://hrcak.srce.hr/230521
3. Arnerić, J., Matković, M., & Sorić, P. (2018). Comparison of range-based volatility estimators against integrated volatility in European emerging markets. Finance Research Letters. https://dx.doi.org/10.1016/j.frl.2018.04.013
4. Arnerić, J., Poklepović, T., & Aljinović, Z. (2014). GARCH based artificial neural networks in forecasting conditional variance of stock returns. Croatian Operational Research Review, 5 (2), 329-343. https://doi.org/10.17535/crorr.2014.0017
5. Azari, A. (2019). Bitcoin price prediction: An ARIMA approach, arXiv preprint arXiv:1904.05315, https://arxiv.org/pdf/1904.05315.pdf [29.3.2021]
6. Bai, J., & Perron, P. (2003). Computation and Analysis of Multiple Structural Change Models, Journal of Applied Econometrics, 18, 1-22.
7. Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics, 31, 307–327.
8. Bouri, E., Shahzadb, S.J.H., & Roubaud, D. (2020). Cryptocurrencies as hedges and safe-havens for US equity sectors. The Quarterly Review of Economics and Finance, 75: 294-307.
9. Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. Journal of Computational and Applied Mathematics, 365.
10. Chu, J., Chan, S., Nadarajah, S., & Osterrieder, J. (2017). GARCH modelling of cryptocurrencies. Journal of Risk and Financial Management, 10, 17. https://doi.org/10.3390/jrfm10040017
11. Corbet, S., Meegan, A., Larkin, C., Lucey, B., & Yarovaya, L. (2018). Exploring the dynamics relationships between cryptocurrencies and other financial assets. Economic Letters, 165: 28-34.
12. Diebold, F.X., & Mariano, R.S. (1995). Comparing Predictive Accuracy, Journal of Business and Economic Statistics, 13, 253–263.

13. Donaldson, R.G., & Kamstra, M. (1997). An artificial neural network - GARCH model for international stock return volatility, Journal of Empirical Finance, 4 (1), 17–46.

14. Dutta, A., Kumar, S., Basu, M. (2020). A Gated Recurrent Unit Approach to Bitcoin Price Prediction, Journal of Risk and Financial Management, 13, 23; https://dx.doi.org/10.3390/jrfm13020023

15. Fahmi, A.M, Samsudin, N.A., Mustapha, A., Razali, N., & Khalid, S.K.A. (2018). Regression based Analysis for Bitcoin Price Prediction, International Journal of Engineering & Technology, 7 (4.38), 1070-1073.

16. Garman, M.B., & Klass, M.J. (1980). On the estimation of security price volatilities from historical data. J. Bus. 53, 67–78

17. Gil-Alana, L.A., Abakah, E.J.A., & Rojo, M.F.R. (2020). Cryptocurrencies and stock market indices. Are they related? Research in International Business and Finance, 51, Article 101063, https://doi.org/10.1016/j.ribaf.2019.101063

18. Greaves, A.S., & Au, B. (2015). Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin. http://snap.stanford.edu/class/cs224w-2015/projects_2015/Using_the_Bitcoin_Transaction_Graph_to_Predict_the_Price_of_Bitcoin.pdf [29.3.2021]

19. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5): 359-366.

20. Indera, N.I, Yassin, I. M., Zabidi, A., & Rizman, Z. I. (2017). Non-linear autoregressive with exogeneous input (narx) bitcoin price prediction model using pso-optimized parameters and moving average technical indicators, Journal of Fundamental and Applied Sciences, 9(3S), 791-808, http://dx.doi.org/10.4314/jfas.v9i3s.61

21. Jalali, M.F.M., & Heidari, H. (2020). Predicting changes in Bitcoin price using grey system theory. Financial Innovations, 6, 13. https://doi.org/10.1186/s40854-020-0174-9

22. Jang, H., & Lee, J. (2017). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information, IEEE Access. Vol. 6, 5427-5437

23. Ji, S., Kim, J., & Im, H. (2019). A Comparative Study of Bitcoin Price Prediction Using Deep Learning, Mathematics, 7, 898. https://dx.doi.org/10.3390/math7100898

24. Khashei, M., & Bijari, M. (2010). An artificial neural network (p,d,q) model for timeseries forecasting, Expert Systems with Applications, vol. 37, pp. 479-489, January.

25. Kuan, C.-M., & T. Liu. 1995. "Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks." Journal of Applied Econometrics, 10(4): 347-364.

26. Lahmiri, S., Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks, Chaos, Solitons and Fractals, 118, 35–40
27. Li, J.P, Naqvi, B., Rizvi, S.K.A., & Chang, H.L. (2021). Bitcoin: The biggest financial innovation of fourth industrial revolution and a portfolio's efficiency booster. Technological Forecasting & Social Change, 162, Article 120383
28. Liu, Y., & Tsyvinski, A. (2018). Risk and Returns of Cryptocurrency, NBER Working Paper, No. 24877, August, 1-25
29. Mantri, J.K., Gahan, P., & Nayak, B.B. (2010). Artificial neural networks – An application to stock market volatility, International Journal of Engineering Science and Technology, Vol. 2(5), 1451–1460.
30. Mantri, J.K., Mohanty, D., & Nayak, B.B. (2012). Design neural network for stock market volatility: Accuracy measurement, International Journal on Computer Technology & Applications, Vol. 3 (1), 242–250.
31. McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of bitcoin using machine learning, in: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing, PDP, IEEE.
32. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. 1-9. Available at https://bitcoin.org/bitcoin.pdf
33. Nybo, C. (2021). Sector Volatility Prediction Performance Using GARCH Models and Artificial Neural Networks, preprint
34. Pabuçcu, H., Ongan, S., & Ongan, A. (2020). Forecasting the movements of Bitcoin prices: an application of machine learning algorithms, Quantitative Finance and Economics, 4(4), 679–692.
35. Parkinson, M., (1980). The extreme value method for estimating the variance of the rate of return. J. Bus. 53, 61–65. https://dx.doi.org/10.1086/296071
36. Platanakis, E., & Urquhart, A. (2020). Should investors include Bitcoin in their portfolios? A portfolio theory approach. The British Accounting Review, 52(4).
37. Polasik, M., Piotrowska, A.I., Wisniewski, T.P., Kotkowski, R., & Lightfoot, G. (2015). Price Fluctuations and the Use of Bitcoin: An Empirical Inquiry, International Journal of Electronic Commerce, 20:1, 9-49, https://dx.doi.org/10.1080/10864415.2016.1061413
38. Poyser, O. (2017). Exploring the determinants of Bitcoin's price: an application of Bayesian Structural Time Series. Dissertation.
39. Raju, K., & Rangaswamy, S. (2017). Forecasting volatility in the Indian equity market using return and range-based models. Appl Econ. 49, 5027–5039. https://doi.org/10.1080/00036846.2017.1299099

40. Rogers, L.C.G., & Satchell, S.E. (1991). Estimating variance from high, low, and closing Prices. Ann. Appl. Probab. 1, 504–512.
41. Sarangi, P., & Dublish, S. (2013). Prediction of gold bullion return using GARCH family and artificial neural network models, Asian Journal of Research in Business Economics and Management Vol. 3, No. 10, 217–230.
42. Šestanović, T., & Arnerić, J. (2020). Neural network structure identification in inflation forecasting. Journal of forecasting, 39, 6; 935-952. https://doi.org/10.1002/for.2698
43. Šestanović, T. (2021). Bitcoin direction forecasting using neural networks, The 16th International Symposium on Operational Research SOR'21, Proceedings. Ljubljana: Slovenian Society Informatika, Section for Operational Research, Bled, Slovenia, 22-24. September., 557-562
44. Sovbetov, Y. (2018). Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litcoin, and Monero, Journal of Economics and Financial Analysis, Vol:2, No:2, 1-27
45. Spilak, B. (2018). Deep Neural Networks for Cryptocurrencies Price Prediction, Master thesis, Humboldt University, Berlin. Reviewers: Härdle, W.K. and Lessmann, S.
46. Stensås, A., Nygaard, M.F., Kyaw, K., & Treepongkaruna, S. (2019). Can Bitcoin be a diversifier, hedge or safe haven tool? Cogent Economics & Finance, 7(1), Article 1593072. https://doi.org/10.1080/23322039.2019.1593072
47. Uras, N., Marchesi, L., Marchesi, M., & Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. PeerJ Comput. Sci. 6:e279 http://doi.org/10.7717/peerj-cs.279
48. Walther, T., Klein, T., & Bouri, E. (2019). Exogenous drivers of Bitcoin and Cryptocurrency volatility – A mixed data sampling approach to forecasting, Journal of International Financial Markets, Institutions & Money, vol. 63(C), https://dx.doi.org/10.1016/j.intfin.2019.101133
49. Yang, D., Zhang, Q. (2000). Drift independent volatility estimation based on high, low, open and closes prices. J. Bus. 73, 477–491. https://dx.doi.org/10.1086/ 209650
50. Yarovaya, L., Brzeczczynski, J., & Lau, C.K.M. (2016). Volatility spillovers across stock index futures in Asian markets: evidence from range volatility estimators. Finance Res. Lett. 17, 158–166. https://dx.doi.org/10.1016/j.frl.2016.03.005
51. Zhang, Y., He, M., Wen, D., & Wang, Y. (2021). Forecasting Bitcoin volatility: A new insight from the threshold regression model. Journal of Forecasting, 1– 20. https://doi.org/10.1002/for.2822

SVEOBUHVATAN PRISTUP PREDVIĐANJU BITCOINA POMOĆU NEURONSKIH MREŽA

Sažetak

Ovaj rad pruža sveobuhvatan pristup predviđanju cijena, prinosa, smjera i volatilnosti Bitcoina. Prediktivne sposobnosti ARIMA i GARCH modela se uspoređuju s autoregresijskom i Jordanovom neuronskom mrežom (NM), koristeći unutarnje i vanjske čimbenike. Robusnost rezultata verificira se u uvjetima pada, rasta i stabilnosti tržišta. Rezultati nisu jednoznačni s obzirom na predviđanje cijena, prinosa ili volatilnosti, te kada se uspoređuju pomoću različitih mjera performansi ili kroz različita razdoblja. NM općenito su optimalne za predviđanje prinosa i smjera, ARIMAX i NNARX za predviđanje cijena, dok za predviđanje volatilnosti svi modeli daju usporedive rezultate. Predviđanje cijena donosi najbolju točnost predviđanja, dok JNNX imaju najlošije rezultate. Međutim, uključivanje drugih metoda strojnog učenja i/ili različitih varijabli, kao i nedavne krize proizašle iz ratnih okolnosti mogu se smatrati ograničavajućim čimbenicima.

Ključne riječi: ARIMA, Bitcon, COVID-19, GARCH, Jordanova neuronska mreža, autoregresijska neuronska mreža