

Identification of the Optimal Neural Network Architecture for Prediction of Bitcoin Return

Tea ŠESTANOVIĆ*, Tea KALINIĆ MILIĆEVIĆ

*University of Split, Faculty of Economics, Business and Tourism
Cvite Fiskovića 5, 21000 Split, Croatia*

e-mail: tea.sestanovic@efst.hr, tea.kalinic@efst.hr

Received: February 2024; accepted: June 2024

Abstract. Neural networks (NNs) are well established and widely used in time series forecasting due to their frequent dominance over other linear and nonlinear models. Thus, this paper does not question their appropriateness in forecasting cryptocurrency prices; rather, it compares the most commonly used NNs, i.e. feedforward neural networks (FFNNs), long short-term memory (LSTM) and convolutional neural networks (CNNs). This paper contributes to the existing literature by defining the appropriate NN structure comparable across different NN architectures, which yields the optimal NN model for Bitcoin return forecasting. Moreover, by incorporating turbulent events such as COVID and war, this paper emerges as a stress test for NNs. Finally, inputs are carefully selected, mostly covering macroeconomic and market variables, as well as different attractiveness measures, the importance of which in cryptocurrency forecasting is tested. The main results indicate that all NNs perform the best in an environment of bullish market, where CNNs stand out as the optimal models for continuous dataset, and LSTMs emerge as optimal in direction forecasting. In the downturn periods, CNNs stand out as the best models. Additionally, Tweets, as an attractiveness measure, enabled the models to attain superior performance.

Key words: Bitcoin, convolutional neural networks, feedforward neural networks, long short-term memory, attractiveness measures.

1. Introduction

Neural networks have been successfully applied in fields such as finance (Sezer *et al.*, 2020), macroeconomics (Šestanović and Arnerić, 2020), engineering (Hegde and Rokseth, 2020), weather forecasting (Purwandari *et al.*, 2021), medicine (Han *et al.*, 2024), and many other (Čorić *et al.*, 2023). Their forecasting ability has recently been tested on time series data, which exhibit features that have to be taken into account and addressed appropriately (Kalinić Milićević and Marasović, 2023; Šestanović, 2024). Especially interesting are financial time series data, which are not stationary, they can have seasonal pattern or cyclical behaviour, they are nonlinear, meaning they can exhibit occasional presence of aberrant observations and the possible existence of regimes within which returns display different dynamic behaviour (Franses and van Dijk, 2003).

*Corresponding author.

The dynamic behaviour of cryptocurrencies, as financial time series, displays extreme observations, asymmetries, and several nonlinear characteristics that are difficult to model and forecast (Šestanović, 2024). Additionally, the importance of cryptocurrency forecasting lies in their constantly increasing financial market, characterized by high volatility and extreme price fluctuations.

Bitcoin prices are highly volatile since they are influenced by a vast number of factors including but not limited to the supply of bitcoins, the cost of the mining process, market demand, as well as political and economic data (Cavalli and Amoretti, 2021). Some papers use internal factors for prediction (Polasik *et al.*, 2015; Jang and Lee, 2017; Sovbetov, 2018; Liu and Tsyvinski, 2020; Spilak, 2018; Fahmi *et al.*, 2018; Ji *et al.*, 2019; Chen *et al.*, 2020; Cavalli and Amoretti, 2021), the others use only Open, High, Low, and Close (OHLC) prices (Indera *et al.*, 2017; Fahmi *et al.*, 2018; Uras *et al.*, 2020; Li and Dai, 2020), 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; Pabuccu *et al.*, 2020; Li and Dai, 2020; Cavalli and Amoretti, 2021). Few papers use macro-finance factors (Polasik *et al.*, 2015; Sovbetov, 2018; Liu and Tsyvinski, 2020; Spilak, 2018; Li and Dai, 2020; Chen *et al.*, 2020) and 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, while the results in Jang and Lee (2017) suggest that macro-financial markets can have a small impact on cryptocurrencies. Moreover, Aljinović *et al.* (2022) report significant dynamic conditional correlations between cryptocurrencies and real estate, S&P500 and gold. However, the majority of papers confirm attractiveness as an important factor that influences cryptocurrency prices (Polasik *et al.*, 2015; Sovbetov, 2018; Li and Dai, 2020; Šestanović, 2021; Cavalli and Amoretti, 2021). This factor is even included among the other indicators in appropriate, comprehensive manner in models for portfolio optimization that include cryptocurrencies (Trimborn *et al.*, 2019; Aljinović *et al.*, 2021). On the other hand, (Kalinić Miličević and Marasović, 2023) report that different models cannot agree upon the importance of Tweets and macro-financial variables in Bitcoin direction forecasting, but show that technical indicators are the most influential, followed by blockchain and market variables. Since different types of attractiveness measures can be found in the literature, the most commonly used measures and their predictive power are compared in this paper, i.e. Google Trends and Tweets.

Previous research has also ambiguous conclusions regarding the appropriate NN model, which calls for further investigation (Cavalli and Amoretti, 2021; Zhang *et al.*, 2021; Li and Dai, 2020; Livieris *et al.*, 2021; Lahmiri and Bekiros, 2019; Ji *et al.*, 2019). Additionally, Lahmiri and Bekiros (2019) revealed that cryptocurrencies exhibit fractal dynamics, long memory and self-similarity. Therefore, an accurate and reliable forecasting model is an essential tool for portfolio managers, which has to be developed and tested.

The feedforward neural networks (FFNNs) are the most popular NN model. Despite their power and proven properties as universal approximators (Hornik *et al.*, 1989), FFNNs have limitations that each input (independent variable) and output (dependent variable) are handled independently, i.e. temporal or space information is not incorporated into the model, which is a significant drawback for time series analysis. Recurrent neural networks

(RNNs), however, are adapted to time series data as they incorporate recurrent connections from output or hidden layers and the so-called self-connected neuron, which allows learning the temporal dynamics of time series data (Madaeni *et al.*, 2022). However, their main problem is capturing long-term dependencies as they suffer from vanishing or exploding gradient. RNN with long short-term memory (LSTM) units has a cell state which enables stable gradients, while the presence of filters can control the information flow (Paranhos, 2021). LSTM emerge as a model most commonly used in financial time series prediction (Sezer *et al.*, 2020). However, convolutional neural networks (CNNs) have recently challenged the LSTMs in their predictive power when working with sequences. Namely, CNNs use filters, which help them learn spatial features from raw time series data (Fawaz *et al.*, 2019).

In this paper, the most commonly used NNs for time series prediction, i.e. FFNNs, LSTM and CNNs, which have proven their forecasting abilities on time series data, are used to forecast Bitcoin returns. The proposed models are compared across different periods, including bullish, bearish and stable market periods, using different performance measures such as means squared error (MSE), accuracy and Diebold-Mariano test, through different inputs and, finally, through different NN architectures. That is, neither input nor NN architecture selection is straightforward and should be chosen with caution. Additionally, Uras *et al.* (2020) confirmed that partitioning datasets into shorter sequences, representing different price “regimes”, enables obtaining precise forecasts. Therefore, Bai-Perron multiple structural break test is used.

To sum up, the main contributions of this paper to the current literature are:

- It defines the appropriate NN structure comparable across different NN architectures, which yields the optimal NN model for Bitcoin’s return forecasting.
- It determines which of the attractiveness measures, already proven in the literature as an important variable for Bitcoin prediction, yields the optimal results.
- It compares the results across different periods based on a non-arbitrary selection of sub-periods using Bai-Perron multiple structural break test and by employing different performance measures that include MSE and accuracy, as well as the Diebold-Mariano test.

The remainder of the paper is organized as follows. Section 2 provides a literature review of related work, Section 3 describes the proposed methodology, including dataset definition, data preprocessing, a description of neural network architectures, as well as model evaluation criteria. Section 4 presents experimental results with discussion. Finally, conclusions and directions for future research are provided in Section 5.

2. Related Work

There are papers that compare FFNNs to other linear and nonlinear models and find NNs to have the highest predictive performances (Greaves and Au, 2015; Pabuccu *et al.*, 2020). Namely, Greaves and Au (2015) compare Support Vector Machine (SVM), Logistic regression (LR), Baseline model and NNs in classification. They obtain the highest classification accuracy of 55.1% with NNs. Pabuccu *et al.* (2020) aim to forecast Bitcoin prices

applying four different machine learning (ML) methods, i.e. SVM, NNs, Naive Bayes (NB), Random Forest (RF) and LR as a benchmark model. They conclude that in a continuous dataset, RF has the best forecasting performance, while NB, the worse. In a discrete dataset, NN has the best forecasting performance, while NB again lags behind other models. Šestanović (2021) confirmed the ability of simple FNNs with lower number of hidden neurons to accurately predict the Bitcoin price direction, compared both to previous research considered and to LR, while Šestanović (2024) predicted Bitcoin price, returns, direction, and volatility. Return and volatility predictions are stable regardless of model or period. Return and direction prediction is best with NNs. ARIMAX and NNARX models predicted prices effectively. All models predict volatility in a similar way. The price prediction was the most accurate, whereas JNNX showed poor performance. However, these papers did not use any sophisticated machine learning models for prediction, which have been proven in the literature to have superior performance.

Since CNN and LSTM methods have proven their forecasting abilities, more and more research has recently been testing their abilities in new circumstances. Several studies confirm that CNN has superior prediction abilities in comparison to LSTM and other NN architectures (Cavalli and Amoretti, 2021; Zhang *et al.*, 2021; Šestanović and Kalinić Milićević, 2023), while some improve the accuracy by combining the CNN and LSTM (Li and Dai, 2020; Livieris *et al.*, 2021). Other research confirms that LSTM exhibits superior predictive abilities when compared to various NN architectures (Lahmiri and Bekiros, 2019; Ji *et al.*, 2019; Spilak, 2018). Contrary, Uras *et al.* (2020) indicate linear regression models outperform NNs, while in Chen *et al.* (2020) LR and Discriminant Analysis outperform more complicated machine learning algorithms. Since the literature does not give a unique answer, this calls for further investigation. Table 1 provides a brief overview of the key features of related research and a comparison of related work concerning variables, data, models and key findings.

Cavalli and Amoretti (2021) predict Bitcoin direction with One-Dimensional (1D) CNN and demonstrate, using large datasets collected in a cloud-based system, that the 1D CNN allows for the prediction of the Bitcoin trend with higher accuracy compared to LSTM models.

Among other papers that combine different deep neural network architectures is Livieris *et al.* (2021) who proposed a multi-input deep learning (MICDL) model based on CNN-LSTM approach for predicting prices of Bitcoin, Ether and XRP. The proposed model is compared to two CNN-LSTM models: model trained with only one cryptocurrency and model trained with all three cryptocurrencies. The utilization of all cryptocurrencies in the training data of the MICDL yielded a forecasting model with the best return and direction predictions.

Motivated by the high correlations among different cryptocurrencies as well as the powerful modelling efficiency exhibited by DL models, Zhang *et al.* (2021) propose a CNN-based Weighted and Attentive Memory Channels model to predict the daily closing price of cryptocurrencies. The results indicate that the proposed model outperforms the baseline models (ARIMA, RF, XGBoost, MLP, LSTM, CNN, GRU, SVM) in predictive performances. The hyperparameter setting of baseline models is chosen by default. Additionally, this paper does not use any other inputs.

Table 1
Key features of prior studies.

	Cryptocurrency	Time period, frequency	Variables	Approach	The best model
Greaves and Au (2015)	Bitcoin	2012-02-01 to 2013-04-01 test set. hourly	Internal factors	SVM, LR, NNs, Logistic Regression	NNs
Spilak (2018)	Bitcoin, Dash, XRP, Monero, Litecoin, Dogecoin, NXT, Namecoin	2014-07 to 2017-10 daily	Internal factors, technical indicators, macroeconomic variables	FFNN, RNN, LSTM	LSTM
Ji <i>et al.</i> (2019)	Bitcoin	2011-11-29 to 2018-12-31 daily	Internal factors	DNN, LSTM, CNN, ResNet, and their combinations, and SVM, GRU, linear/logistic R	LSTM
Lahmiri and Bekiros (2019)	Bitcoin, Bitcoin Cash and XRP	Bitcoin: 2010-07-16 to 2018-10-01, Digital Cash: 2010-02-08 to 2018-10-01, Ripple: 2015-01-21 to 2018-10-01 daily	Internal factors	LSTM, GRNN	LSTM
Chen <i>et al.</i> (2020)	Bitcoin	2017-02-02 to 2019-02-01 daily	Internal factors, sentiment analysis variables, macroeconomic factors	Logistic Regression, DA, RF, XGBoost, Quadratic DA, SVM, LSTM	LR and DA
Li and Dai (2020)	Bitcoin	2016-12-30 to 2018-08-01 daily	Internal factors, technical indicators, macroeconomic variables, sentiment analysis variables	BPNN, CNN, LSTM, CNN-LSTM	CNN-LSTM
Pabuccu <i>et al.</i> (2020)	Bitcoin	2008 to 2019 daily	Internal factors, technical indicators	SVM, NNs, NB, RF, Logistic Regression	RF (continuous dataset), NNs (discrete dataset)
Uras <i>et al.</i> (2020)	Bitcoin, Litecoin and Ether	2015-11-15 to 2020-03-12 daily	Internal factors	MLR, FFNN, LSTM	LR
Cavalli and Amoretti (2021)	Bitcoin	2013-04-28 to 2020-02-15 daily	Internal factors, technical indicators, sentiment analysis variables	CNN, LSTM	CNN
Livieris <i>et al.</i> (2021)	Bitcoin, Ether and XRP	2017-01-01 to 2020-10-31 daily	Internal factors,	Three CNN-LSTM models based on different sets of inputs	MICDL

(continued on next page)

Table 1
(continued)

	Cryptocurrency	Time period, frequency	Variables	Approach	The best model
Zhang <i>et al.</i> (2021)	Bitcoin, Bitcoin Cash, Litecoin, Ether, EOS, and XRP	2017-07-23 to 2020-07-15 daily	Internal factors	ARIMA, RF, XGBoost, MLP, LSTM, CNN, GRU, SVM	CNN
Šestanović (2021)	Bitcoin	2016-04 to 2021-04	Internal factors, macroeconomic factors, sentiment analysis variables	Logistic Regression, FFNN	FFNN
Jaquart <i>et al.</i> (2022)	100 cryptocurrencies	2018-02-08 to 2022-05-15 daily	Internal factors	LSTM, GRU, TCN, GB, RF, LR	GRU, LSTM
Šestanović and Kalinić Miličević (2023)	Bitcoin	2017-07-05 to 2022-01-01 daily	Internal factors, sentiment analysis variables, macroeconomic factors	FFNN, CNN, LSTM	CNN
Šestanović (2024)	Bitcoin	2016-04-09 to 2021-04-09 daily	Internal factors, macroeconomic factors, sentiment analysis variables	ARIMAX, NNARX, JNNX, GARCH, NNAR, JNN, FNN, Logistic Regression	NNs for return and direction forecasting, ARIMAX and NNARX for price forecasting

Note: Autoregressive Integrated Moving Average (ARIMA), ARIMA with exogenous inputs (ARIMAX), Back Propagation Neural Network (BPNN), Discriminant Analysis (DA), Convolutional Neural Network (CNN), Deep Neural Network (DNN), Deep Residual Network (ResNet) Extreme Gradient Boosting (XGBoost), Feed Forward Neural Network (FFNN), Gated Recurrent Units (GRUs), General Regression Neural Network (GRNN), Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Jordan Neural Networks (JNN), Jordan Neural Networks with exogenous inputs (JNNX), Linear Regression (LR), Long Short-Term Memory (LSTM), Multiple-Input Cryptocurrency Deep Learning Model (MICDL), Multilayer Perceptron (MLP), Multiple Linear Regression (MLR), Naive Bayes (NB), Neural Networks (NNs), Neural Network Autoregression (NNAR), Neural Network Autoregression with Exogenous Input (NNARX), Random Forest (RF), Recurrent Neural Network (RNN), Support Vector Machine (SVM), Temporal Convolutional Networks (TCN).

Li and Dai (2020) propose a hybrid NN model based on CNN and LSTM. CNN is used for feature extraction, which become inputs to LSTM for training and prediction of the Bitcoin price. They conclude that CNN-LSTM can effectively improve the accuracy of both value and direction prediction compared with simple NNs.

Uras *et al.* (2020) forecast daily closing prices of Bitcoin, Litecoin and Ether, using Simple and Multiple Linear Regression model (RM), as well as FFNN and LSTM models. The best results were found with RM and LSTM models. However, the linear RMs outperform NNs.

Chen *et al.* (2020) predict Bitcoin price at daily and high-frequency intervals. LR and Discriminant Analysis (DA) achieve an accuracy of 66%, outperforming more complex machine learning (ML) models. ML models include RF, XGBoost, Quadratic DA, SVM and LSTM.

Lahmiri and Bekiros (2019) implement LSTM and generalized regression NN (GRNN) to forecast the prices of Bitcoin, Bitcoin Cash and XRP. The predictability of LSTM is significantly higher compared to the GRNN. LSTM are proved highly efficient in forecasting cryptocurrency prices.

Ji *et al.* (2019) compare deep NN (DNN), LSTM, CNN, deep residual network, and their combinations for Bitcoin price prediction, as well as SVM, GRU and linear/logistic RMs which performed worse or equally to SVM. They conclude that LSTM slightly outperforms the other models. Moreover, DNN performed the best in the classification problem.

Spilak (2018) uses FFNN, RNN and LSTM in classification tasks to predict price directions of 8 major cryptocurrencies with a rolling window RM. The study reveals that LSTM has the highest accuracy for direction prediction of the most important cryptocurrencies, FFNN has the best generalization power for three cryptocurrencies, while RNN shows poor prediction performances, seemingly failing to extract the necessary information.

Jaquart *et al.* (2022) train models to predict binary relative daily market movements of the 100 largest cryptocurrencies. They use only daily closing prices and market capitalization data. GRU and LSTM models perform the best, as portfolios based on these models' predictions yield the highest performances.

Šestanović and Kalinić Miličević (2023) estimated FFNN, CNN and LSTM models in the downturn period for Bitcoin return prediction by applying the multi-criteria decision-making approach for model selection. They concluded that optimal model is CNN.

In line with previous research findings that the majority of optimal models are constructed using CNN and LSTM, this article presents a comprehensive approach to forecasting Bitcoin returns using FFNN, CNN and LSTM. The employed NN architectures are thoroughly explained and compared across various architectures and periods in order to reach a conclusion regarding the important NN configurations. Namely, this paper emerges as a stress test for NN architectures, since it tests the abilities of more sophisticated NN architectures in different sub-periods, which include bullish, bearish and stable market conditions, obtained in a logical way, i.e. using Bai-Perron structural break test. Previous researches do not include downturn periods in their analysis, which are usually more difficult to predict. The results are compared through different performance measures and tested using Diebold-Mariano test. Additionally, although some previous researches use sentiment analysis variable, they do not compare their performances. Finally, although other papers sometimes use even more inputs for prediction, they narrow them down to the use of only technical indicators or internal factors, while in this paper the representatives of important factors are used in a comprehensive manner.

3. Proposed Methodology

3.1. Dataset Definition, Preprocessing and Partitioning

In order to create the initial dataset, different types of factors are extracted. They can be divided into four main categories: internal factors, technical indicators, external factors, and attractiveness measures. The considered factors are given in Table 2.

Table 2
Factors considered in analysis.

Category	Variables	Source
Internal factors	Close price, volume, market capitalization, average block size, average block time, average hash rate, average transaction fee	https://bitinfocharts.com
Technical indicators	Moving average of close price, lag return	Calculated
External factors	S&P500, VIX, Gold	https://fred.stlouisfed.org
Attractiveness measure	Google Trends, Tweets	https://bitinfocharts.com

Note: Database is available at: https://github.com/TKalini/Sestanovic-Kalinic-Milicevic-2024/blob/main/Database_SestanovicKalinicMilicevic.csv.

Market and blockchain variables are differentiated among internal factors. Market data usually include OHLC Bitcoin prices, volume, and market capitalization. Since Bitcoin prices are not affected by seasonality like stock prices, the open, high, and low Bitcoin prices are excluded from our analysis. In that manner, the dimensionality problem is avoided. Average block size, average block time, average hash rate, and average transaction fee are selected from a set of available blockchain metrics. Namely, Poyser (2017) define supply and demand (transaction cost, reward system, mining difficulty, coins circulation, rule changes), i.e. blockchain measures, as the main internal factors that have a direct impact on their market price. Since previous research involving different technical indicators yielded good results (Pabuccu *et al.*, 2020; Li and Dai, 2020; Cavalli and Amoretti, 2021), the moving average of the close price, as well as the lag return are selected as inputs as well. Although researchers disagree on the impact of external factors on Bitcoin prices and returns, the following external factors: S&P500, Chicago Board Options Exchange volatility index (VIX), and Gold prices are considered in this paper. Finally, widely utilized indicators of attractiveness such as Google Trends are used. Google Trends provides insights into the fluctuation of interest in Bitcoin as a search term over a certain time-frame, while Tweets indicate the daily count of tweets using the word “Bitcoin”. Bitcoin closing prices are used to calculate the return for the following day, which is the dependent variable in the model (Šestanović and Kalinić Milićević, 2023).

More formally, the next-day return R_t , is calculated as follows:

$$R_t = \ln \frac{P_{t+1}}{P_t}, \quad (1)$$

where P_t are Bitcoin closing prices. Moreover, prices P_t are used to calculate lag returns as well as moving average values, for different window lengths w with equations (2) and (3):

$$\text{Lag} R_t = \ln \frac{P_t}{P_{t-1}}, \quad (2)$$

$$MA_t = \frac{1}{w} \sum_{i=t-w}^t P_i. \quad (3)$$

Using the aforementioned variables, i.e. different internal factors, technical indicators, external factors, and based on two attractiveness measures, i.e. Google Trends and Tweets,

two initial (IN) datasets are created: D_{IN}^1 containing all the variables and the variable Google Trend and D_{IN}^2 containing all the variables and the variable Tweets. Those sets were used to compare the predictive performance of the models depending on the selected measures of attractiveness.

More formally, initial datasets can be defined in the form of supervised data as $D_{IN}^1 = \{(x_{t,i}, y_t)\}_{t \in T'}$ and $D_{IN}^2 = \{(z_{t,i}, y_t)\}_{t \in T'}$, where T' is the total initial sample size, $x_t \in \mathbf{R}^p$ and $z_t \in \mathbf{R}^p$ are vectors of $p = 14$ ($i = 1, \dots, p$) independent variables differing only by the attractiveness measure, and $y_t = R_t \in \mathbf{R}$ is a dependent variable, i.e. next-day return.

The data-preprocessing phase of analysis consists of dealing with missing data, calculating the percentage changes, and scaling the data. The selected period does not include an abundance of data gaps that would reflect negatively on the quality of the chosen data collection. Missing values were found at some points in time, and were mostly related to external factors. Linear interpolation was used to fill in the gaps.

In order to improve training efficacy and NN convergence, the actual values for the majority of independent variables were replaced with percentage changes. For all the observed variables except lag returns ($i = 1$), existing values are replaced with corresponding percentage changes, i.e. $x_{t,i} \leftarrow \frac{(x_{t,i} - x_{t-1,i})}{x_{t,i}}$, $t \in T'$, $i = 2, \dots, p$ and $z_{t,i} \leftarrow \frac{(z_{t,i} - z_{t-1,i})}{z_{t,i}}$, $t \in T'$, $i = 2, \dots, p$.

Following the preparation of the initial datasets D_{IN}^1 and D_{IN}^2 , each of final dataset $D^1 = \{(x_{t,i}, y_t)\}_{t \in T}$ and $D^2 = \{(z_{t,i}, y_t)\}_{t \in T}$ for $T = 0, 1, \dots, 2337$ included in our analysis contains a total of 2338 points. Indices $t = 0$ and $t = 2337$ correspond to the dates 2016-01-06 and 2022-05-31, respectively. To conclude this step of preprocessing, given that machine learning algorithms perform better with scaled data, the min-max scaler, which rescales variables into the $[0, 1]$ range, is used.

In the majority of research papers examining Bitcoin price and return forecasting, the dataset is split into a training set and a testing set in a predetermined proportion. On the contrary, in this paper, the breakpoints in the Bitcoin price trend are detected. The model is then trained on the period preceding the trend break and tested on the part of period following the breakpoint for each breakpoint. In this way, the ability of a NN model to make predictions in a new environment which is set up in an objective and unbiased manner can be tested. The Bai-Perron structural break change test is used to detect structural breaks. Bai and Perron (1998) considered issues associated with multiple structural changes in the linear regression model derived by minimizing the sum of squared residuals. Throughout, the dates of the m breaks were treated as unknown variables that needed to be estimated. The primary considerations are the features of the estimators, particularly the break date estimates, and the design of tests that provide inferences about the presence of structural change and the number of breaks. This test is employed to identify breakpoints, resulting in the identification of indices of five dates $T_1, T_2, \dots, T_5 \in T$ at which the test recognized a structural change. For each T_k , $k = 1, \dots, 5$ two sets are defined:

1. Train set $D_{train}^k \{(x_{t,i}, y_t) | t \in \{0, 1, \dots, T_k\}\} \subset D^1$ and
2. Test set $D_{test}^k \{(x_{t,i}, y_t) | t \in \{T_k + 1, \dots, T_k + n\}\} \subset D^1$, where $n = \frac{T_k \cdot 5}{100}$.

Table 3
Dataset partitions obtained with Bai-Perron multiple structural break test.

	Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Start date	2016-01-06	2016-01-06	2016-01-06	2016-01-06	2016-01-06
Close date	2016-12-27	2017-12-15	2019-03-25	2020-03-11	2021-03-12
Number of data points	357	710	1175	1527	1893

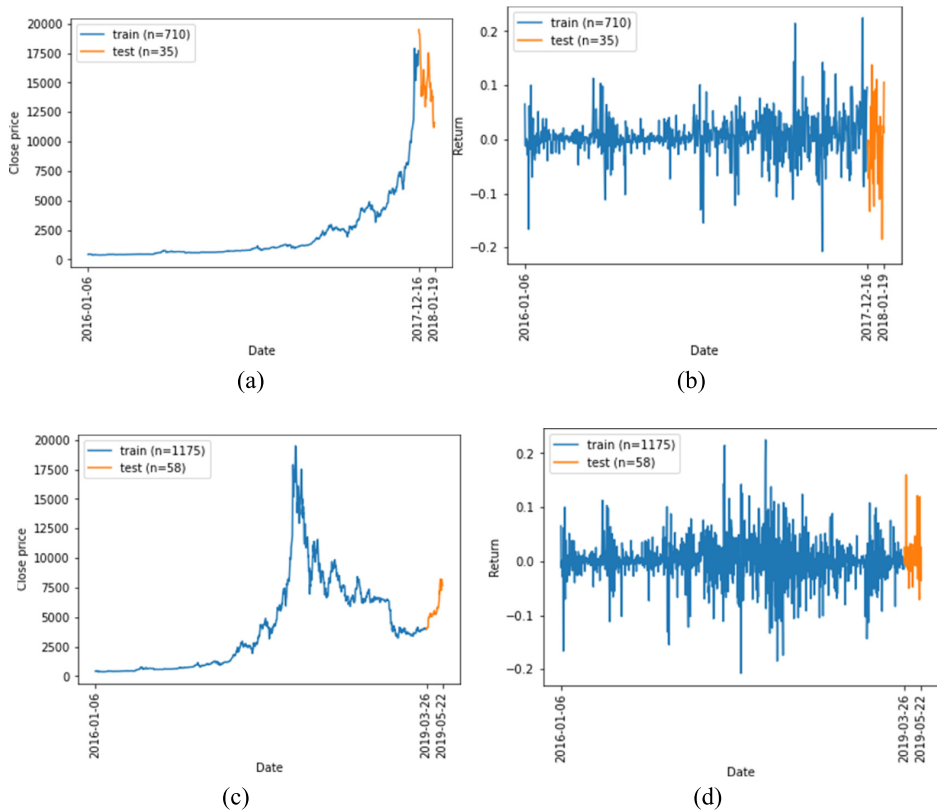


Fig. 1. Flow chart of close price and return within train and test sets for observed partitions.

Namely, each subset, or training set, is followed by a test set. The size of each test set corresponds to five percent of the size of the associated train set.

Table 3 displays the time interval and number of observations for each partition. Four out of five partitions (subsets) are used to build models with different NN architecture. Due to the low quantity of data points, the first partition is excluded from the study.

A graphical representation of price movements and next-day returns within the training and testing set for each observed partition are shown in Fig. 1.

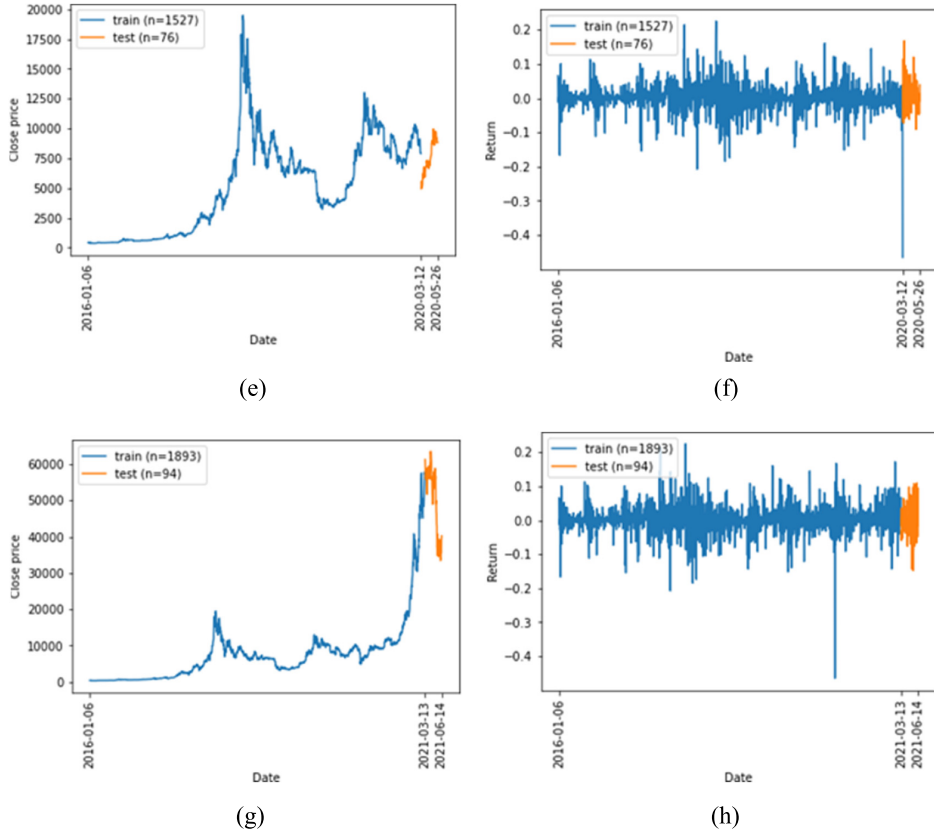


Fig. 1. (continued)

3.2. Neural Network Architectures

In this subsection, the structure of neural networks, as well as the evaluation methods for models is described. For the sake of simplicity, the preceding will be explained for set D^1 , whereas in the experiment, the same was also done for the set D^2 .

3.2.1. Feedforward Neural Networks

Feedforward neural networks (FFNNs) are the most commonly used NNs. They consist of three layers: input, hidden, and output. Inputs and outputs are the independent and dependent variables predefined by the researcher, while hidden neurons are one of hyper-parameters that have to be fine-tuned. The unknown parameters (weights) are estimated using the backpropagation (BP) learning algorithm. FFNN can be written as follows:

$$y_t = \sigma_1 \left(w_{co} + \sum_{h=1}^q w_{ho} \sigma_2 \left(w_{ch} + \sum_{i=1}^p w_{ih} x_{t,i} \right) \right) + \varepsilon_t, \quad (4)$$

where y_t is the output vector of a time series, $x_{t,i}$ is the input matrix with p variables, while $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$ are the activation functions in output and hidden layer respectively, which can be sigmoid, hyperbolic tangent and/or linear. Weights w_{co} and w_{ch} are constant terms of output and hidden neurons respectively. Weights w_{ih} and w_{ho} are the connections between inputs and hidden neurons and between hidden neurons and output respectively. ε_t is an error term.

3.2.2. Long Short-Term Memory

Long Short-Term Memory (LSTM) merges LSTM units, composed of cells, which have input, output and forget gate to control the information flow, to form the LSTM layer. LSTM is given in Eqs. (5)–(9):

$$f_t = \sigma_g(W_f x_{t,i} + U_f y_{t-1} + b_f), \quad (5)$$

$$i_t = \sigma_g(W_i x_{t,i} + U_i y_{t-1} + b_i), \quad (6)$$

$$o_t = \sigma_g(W_o x_{t,i} + U_o y_{t-1} + b_o), \quad (7)$$

$$c_t = f_t * c_{t-1} + i_t * \sigma_y(W_c x_{t,i} + U_c y_{t-1} + b_c), \quad (8)$$

$$y_t = o_t * \sigma_y(c_t), \quad (9)$$

where $x_{t,i}$ is the input vector to the LSTM unit. f_t , i_t and o_t are the forget, input and output gate's activation vectors respectively. y_t is the output vector of the LSTM unit, c_t is the cell state vector, σ_g and σ_y are sigmoid and hyperbolic tangent functions respectively. $*$ is the element-wise (Hadamard) product, W and U are weight matrices and b are the bias vectors (Bao *et al.*, 2017; Sezer *et al.*, 2020).

3.2.3. Convolutional Neural Networks

Convolutional neural networks (CNNs) have different layers in their architecture: convolutional, max-pooling, dropout and fully connected FFNN layer. The convolutional layer consists of the convolution (filtering) operation, which is shown in Eq. (10):

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a), \quad (10)$$

where t is time, s is feature map, w is kernel, x is input, and a is variable. In addition, the convolution operation is implemented on two-dimensional images given in Eq. (11):

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n), \quad (11)$$

where I is input image, K is kernel, m and n are dimensions of images, i and j are variables. Consecutive convolutional and max-pooling layers are also a part of the deep network architecture. CNN also includes the FFNN architecture given in Eq. (12):

$$z_i = \sum_j W_{i,j} x_j + b_i, \quad (12)$$

where W are the parameters, x is input vector, b is bias vector, z is the output from the neurons which are put forward through the softmax activation function for the calculation of the output (y) in the output layer, which is shown in Eqs. (13) and (14) (Sezer *et al.*, 2020):

$$y = \text{softmax}(z_i), \quad (13)$$

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}. \quad (14)$$

3.3. Model Evaluation

Given that the observed problem of predicting the next-day returns is a regression problem, the performance of the model was evaluated using mean squared error (MSE) which is calculated using the formula:

$$MSE = \frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2, \quad (15)$$

where y_t and \hat{y}_t are observed and predicted values of returns, and T is total number of observations. Furthermore, considering that when predicting the next-day return, apart from the value itself, it is also important to accurately predict the sign of the next-day return, it was decided to analyse the obtained models from that perspective as well. Therefore, we converted the dependent variables from continuous to discrete, i.e. binary values, using the following rule:

$$y_{bin,t} = \begin{cases} 1, & \text{if } y_t \geq 0, \\ 0, & \text{if } y_t < 0. \end{cases} \quad (16)$$

The above conversion was performed on both, the observed and predicted values, of the dependent variable, as well as for train and test set. Accuracy (ACC) was used as a metric to quantify the ability of models to predict the direction of price movements. It is the ratio of number of correct predictions (i.e. true positive- TP and true negative- TN) to the total number of input samples (T).

$$ACC = \frac{TP + TN}{T}. \quad (17)$$

In addition, the Diebold-Mariano test (Diebold and Mariano, 1995) is used to test the equality of predictive ability between the two models, i.e. to test whether there is a statistically significant difference in the forecasting performances of the proposed models. It enables finding the optimal NN that has the highest forecasting performances.

4. Experimental Results

4.1. NNs Configuration

Three different NN architectures are used to build models: FFNN, LSTM and CNN. FFNNs consist of input, hidden, and output neurons. Inputs and outputs are theoretically driven and predefined by the researcher, while hidden neurons are one of the hyperparameters that have to be fine-tuned. Based on Patterson (1998), Moshiri and Cameron (2000) and Hwarng (2001), the following five Working rules for determining the number of hidden neurons are proposed:

- Patterson (1998): $q_1 = \frac{T_i}{10(p+1)}$;
- Moshiri and Cameron (2000): $q_2 = \frac{T_i}{5(p+1)}$ and $q_3 = \sqrt{p}$;
- Hwarng (2001): $q_4 = \frac{p}{2}$ and $q_5 = \frac{3p}{2}$,

where T_i stands for the number of observations in the train set, p for the number of independent variables, and there is only one dependent variable. Considering that this research is conducted on four different training sets, the value of T_i varies for each set, whereas there are always $p = 14$ independent variables. Table 4 shows the observed number of neurons calculated using each of the five above mentioned formulas and for each of the four observed subsets. The set of neurons is given in Table 5. Table 6 presents the main hyperparameters of NNs configurations.

The proposed NNs were trained on each of the four subsets and evaluated on the associated test sets for various parameter values, while the fixed seed was used for the reproducibility of the results in one experimental run for each NN. The research identified 729 unique models. In addition, for each model, for the true and predicted values of the dependent variable in the test set, equivalent binary values (signifying the direction of the Bitcoin price) were generated and used to measure the accuracy of each model. Finally, models were compared based on both MSE and accuracy (ACC).

Table 4
Number of neurons with different formulas for different subsets.

	Subset 2	Subset 3	Subset 4	Subset 5
q_1	4.73	7.83	10.18	12.62
q_2	9.47	15.67	20.36	25.24
q_3	3.74	3.74	3.74	3.74
q_4	7	7	7	7
q_5	10.5	10.5	10.5	10.5

Table 5
Set of observed neurons for different subsets.

	Subset 2	Subset 3	Subset 4	Subset 5
Set of neurons	3, 4, 7, 9, 10, 11	3, 4, 7, 10, 11, 15	3, 4, 7, 10, 11, 20	3, 4, 7, 10, 12, 25

Table 6
NNs configurations.

FFNN	LSTM	CNN
$p = 14$ learning algorithm is stochastic gradient descent learning rates are 0.01, 0.001, 0.0001 loss function is mean square error batch sizes are 2, 32, set size number of epochs is 500		
<ul style="list-style-type: none"> • one hidden layer, • tangent hyperbolic activation functions, • set of neurons in Table 5. 	<ul style="list-style-type: none"> • one LSTM layer, • one dense layer, • neurons in Table 5 multiplied by 10. 	<ul style="list-style-type: none"> • 1-dimensional convolutional layer, • MaxPooling1D layer for the max-pooling layer, • tangent hyperbolic activation function, • 32 filters, • pool size with size 2, • kernel sizes are 210, 330, 450 and 540 for the observed four subsets,¹ • set of neurons in Table 5.

¹ Kernel size are calculated with formula $\frac{\text{setsize}}{100} \cdot 30$ (Cavalli and Amoretti, 2021).

Because the observed datasets differ in terms of the measure of attractiveness, models containing Google Trends and models containing Tweets are compared independently.

4.2. Comparison of Models for Google Trends Dataset

Models on four partitions of a dataset that included Google Trends as an attractiveness measure using three NNs architectures with different parameters set up are constructed. For each NN structure and each subset, the best model with the lowest MSE was aimed for. The best models obtained together with tuned hyperparameters are shown in Table 7.

The first section of the table shows the values of variable hyperparameters, while the second section shows the values of various performance measures for each model, along with Diebold-Mariano test of predictive performances for the models in pairs. Models are ranked according to the MSE value on the test set. The first fifty percent of the ranks are allocated to the models developed with Subsets 3 and 4. These Subsets are characterized by an upward trend in Bitcoin prices, which is well predicted by all NN models. In Subsets 3 and 4, CNN has the lowest MSE, which is significantly lower than both FFNN and LSTM.

There is no statistically significant difference between LSTM and FFNN based on the DM test. The models constructed on Subset 2 ranked the worst, followed by models built on Subset 5. In these two Subsets, i.e. the bearish market, FFNN has the best predictive performance in terms of MSE. In both Subsets, the DM test shows that there is no significant difference between FFNN and LSTM models, while they both outperform CNN. Clearly, NNs could not capture the slump in Bitcoin prices at the end of 2017 and in the beginning of the year 2021. The best two models, when comparing MSE on the test set, are CNNs for the Subsets 3 and 4. Cavalli and Amoretti (2021), Zhang *et al.* (2021) and Šestanović and Kalinić Miličević (2023) conclude similarly.

Table 7
The optimal models for each subset (S2 to S5) for Google Trends along with tuned hyperparameters, performance measures and Diebold-Mariano test.

Subsets	Nmb. of neurons	Learning rate	Batch size	Test MSE	Test ACC	Test MSE rank	Test ACC rank	FFNN	LSTM
S2-FFNN	7	0.001	32	0.00506	48.57%	10	11	/	/
S2-LSTM	40	0.0001	32	0.00511	54.29%	11	6	1.2736	/
S2-CNN	9	0.001	710	0.00535	61.54%	12	5	2.6778**	2.2926**
				0.00517	0.54799				
S3-FFNN	7	0.0001	32	0.00157	63.79%	4	3	/	/
S3-LSTM	70	0.01	2	0.00155	65.52%	3	1	0.8786	/
S3-CNN	10	0.01	1175	0.00131	65.31%	1	2	2.3097**	2.2330**
				0.00148	0.64872				
S4-FFNN	3	0.0001	32	0.00181	63.16%	5	4	/	/
S4-LSTM	40	0.01	1527	0.00201	48.68%	6	10	2.1783**	/
S4-CNN	3	0.001	32	0.00154	49.25%	2	9	2.4464**	2.2327**
				0.00179	0.53699				
S5-FFNN	25	0.001	2	0.00211	46.81%	7	12	/	/
S5-LSTM	250	0.001	2	0.00213	51.06%	8	8	0.7360	/
S5-CNN	12	0.001	32	0.00227	51.76%	9	7	4.1886***	4.1958***
				0.00217	0.49879				

Note: *, ** and *** indicate significance at the 0.1, 0.05 and 0.01 levels respectively.

Source: The authors' calculations in Python and R.

If the accuracy (ACC) of models is analysed (Table 7), the top three models come from the Subset 3, characterized by a slow and steady increase in Bitcoin prices. Excellent results come from CNN in the Subset 2, which means that CNN was able to capture the direction of Bitcoin prices in a downturn period at the end of 2017 and the beginning of 2018. In similar conditions in Subset 5, CNN performed slightly better than other models. Additionally, FFNN was able to capture the direction of Bitcoin prices movement in the Subset 4, characterized by a significant slump in Bitcoin prices at the beginning of the Covid-19 crisis followed by a sharp increase in Bitcoin prices. FFNNs are proved to have good generalization power in Spilak (2018), while the same is confirmed in this paper for the extremely volatile period of Bitcoin returns. The optimal model when considering the accuracy measure is the LSTM in Subset 3 reaching the accuracy of 65.52%. This confirms the finding of several researches about superiority of LSTM models (Lahmiri and Bekiros, 2019; Ji *et al.*, 2019; Spilak, 2018).

The most commonly used learning rate across all models is 0.001. Using a learning rate of 0.001 allows the convergence of the learning algorithm (Šestanović and Arnerić, 2020). In addition, the LSTM model used the largest number of neurons and the smallest batch size, whereas the CNN model utilized the second largest number of neurons and a batch size of 32. All three models conducted on the largest subset are the ones that used the largest number of neurons. However, the predictive performances of those models are among the worse. It is confirmed that using a lower number of hidden neurons leads to optimal models with good predictive performances.

Table 8

The optimal models for each subset (S2 to S5) for Tweets along with tuned hyperparameters, performance measures and Diebold-Mariano test.

Subsets	Nmb. of neurons	Learning rate	Batch size	Test MSE	Test ACC	Test MSE rank	Test ACC rank	FFNN	LSTM
S2-FFNN	3	0.0001	32	0.00506	57.14%	10	6	/	/
S2-LSTM	100	0.0001	32	0.00530	54.29%	12	9	0.3717	/
S2-CNN	11	0.0001	32	0.00523	61.54%	11	4	1.2283	1.2542
				0.00520	0.57656				
S3-FFNN	16	0.001	1175	0.00153	65.52%	2	1	/	/
S3-LSTM	80	0.01	2	0.00159	65.52%	3	1	0.4819	/
S3-CNN	8	0.01	1175	0.00127	63.27%	1	3	2.1081**	2.1710**
				0.00146	0.64767				
S4-FFNN	20	0.01	1527	0.00178	56.58%	5	7	/	/
S4-LSTM	70	0.001	32	0.00209	48.68%	7	11	0.4705	/
S4-CNN	10	0.0001	2	0.00172	40.30%	4	12	2.9165***	2.5437**
				0.00187	0.48521				
S5-FFNN	12	0.01	1893	0.00208	56.38%	6	8	/	/
S5-LSTM	30	0.001	2	0.00214	50.00%	8	10	1.5761	/
S5-CNN	10	0.0001	32	0.00225	58.82%	9	5	4.2440***	3.7034***
				0.00216	0.55069				

Note: *, ** and *** indicate significance at the 0.1, 0.05 and 0.01 levels respectively.

Source: The authors' calculations in Python and R.

4.3. Comparison of Models for Tweets Dataset

Three NN architectures with different parameter settings are estimated on four subsets of a dataset that includes Tweets as an attractiveness measure. The best models with the lowest MSE for each NNs structure and subset are sought, and along with tuned hyperparameters, performance measures and Diebold-Mariano test for predictive performances are shown in Table 8.

Based on the lowest MSE in the test set, three lowest ranks are again assigned to models developed with Subset 2, i.e. in the downturn of cryptocurrency market. In this Subset, FFNN has the best predictive performances in terms of MSE. However, there is no significant difference between the NN models according to DM test. The best results are obtained by models associated with Subset 3, i.e. in bullish market conditions, where CNN has the lowest MSE which is significantly lower than both FFNN and LSTM according to DM test. There is no statistically significant difference between FFNN and LSTM. Other ranks are divided between models conducted on Subsets 4 and 5, with Subset 4 models slightly predominating. In Subset 4, i.e. upward market conditions, CNN has the best predictive performances and it is slightly better than both FFNN and LSTM, while there is no statistically significant difference between FFNN and LSTM. Finally, in Subset 5, i.e. the downturn period, FFNN has the lowest MSE. DM test shows that it is not significantly different from LSTM. However, they both outperform CNN.

From the standpoint of accuracy, nine out of twelve models reach an accuracy on the test set higher than 50%, i.e. they have good predictive power. The best among them are FFNN and LSTM for the Subset 3 reaching an accuracy of 65.52%. Comparable results

Table 9
Diebold-Mariano test for comparison of Google
Trends and Tweets through subsets.

Google Trends v. Tweets	FFNN	LSTM	CNN
S2	0.7141	1.3319	1.1741
S3	0.1654	0.0195	0.1654
S4	0.8715	1.5630	0.1183
S5	0.2376	1.7107*	1.5155

Note: * indicates significance at the 0.1 level.

Source: The authors' calculations in Python and R.

are achieved in the same Subset with CNN. In contrast, during another bullish period covered by Subset 4, FFNN achieves the highest accuracy, whereas the other two models have the worst results overall. In Subsets 2 and 5, which represent periods of downturn, CNN outperforms the other models.

The most common learning rate is 0.0001 while in the previous section, it was 0.001. The most common batch size in both sections is 32. Neither of these models has the largest amount of available neurons in its configuration, which leads to the conclusion that using a lower number of hidden neurons leads to optimal models with good predictive performances.

4.4. Comparative Analysis of Google Trends and Tweets

This section provides a comparative analysis of two attractiveness measures. The last rows of Tables 7 and 8 present the average values of the observed measures for models with all three NNs architecture. Comparing the average test performances of models run on these two datasets whose attractiveness measures differ, the dataset with Tweets as the attractiveness has dominance in the downturn periods, i.e. Subsets 2 and 5, as it predicted the direction of Bitcoin prices on average better than Google Trends. However, the highest accuracy reached is in Subset 3 and it is the same for both variables. Moreover, the average accuracy of models in Subset 3 is nearly identical for both attractiveness measures. Google Trends significantly outperforms Tweets only in Subset 4. Therefore, the dataset with Tweets as the attractiveness measure enabled the models to attain superior performance in terms of accuracy. In terms of MSE, according to DM test, there is no statistically significant difference between the two attractiveness measures. The results of DM test are given in Table 9.

5. Discussion

The models with the best forecasting performances on both observed performance measures are those in Subset 3, which is characterized by an increase in price but lower volatility. This confirms the findings of previous research, in which models tested during periods of stable Bitcoin price growth performed well. Subset 4 was characterized by a significant

slump in Bitcoin prices due to the uncertainty of the Covid-19 crisis, followed by a more volatile price increase. Thus, on average, all NNs predicted the movement properly. However, predicting direction was more challenging for more complex NNs. Namely, FFNN outperformed other NNs in direction forecasting. This result comes as a surprise since it contradicts previous research findings that CNN outperforms other NN architectures (Cavalli and Amoretti, 2021; Zhang *et al.*, 2021; Šestanović and Kalinić Milićević, 2023). In Subsets 2 and 5, which cover the significant downturn period in Bitcoin prices, including extremely volatile periods, CNN managed to capture price fluctuations, while FFNN dominated on average in predicting Bitcoin returns. CNNs have the advantages inherent in their architecture, incorporating FFNN model after the convolutional layer. This enables a lower number of hidden neurons and higher predictive performances, while avoiding the overfitting problem in the process. CNNs are robust and need less training time compared to RNNs or FFNNs, and can reduce the complexity of the model (Madaeni *et al.*, 2022). Based on accuracy, the model with LSTM architecture is superior to all others, confirming the findings in Ji *et al.* (2019) and Spilak (2018). CNN outperforms other models in downturn period, while FFNN outperforms the other models in bullish market conditions. This means that simpler NNs can be used for predictions in bullish market, while more complex NNs should be used for predictions in bearish market. The results of MSE produce diametrically opposed conclusions. Namely, MSE and accuracy do not always agree when it comes to identifying the optimal prediction model. Since MSE is composed of both bias and variance, if one estimator has lower MSE than another, it is not known whether this is due to lower bias or lower variance (i.e. higher precision). Therefore, further investigation is needed to determine whether it is more appropriate to observe regression models through these two perspectives or to simply use classification models to find the model which has better prediction of price direction.

6. Conclusion

In this paper, the predictive performances of three commonly used NN models were compared using different performance measures on different subsets of datasets, differentiated by two attractiveness measures. These subsets entail different market conditions, i.e. bearish or bullish periods. Thus, this paper examined the ability of these machine learning models in all types of environments, including bullish, bearish, and stable periods, as well as periods characterized by high volatility. All NNs performed best in an environment of bullish market, where CNN stood out as the optimal NN model using MSE, while FFNN and LSTM emerged as optimal models in direction forecasting. However, based on accuracy, CNN outperformed other models in downturn periods, while FFNN outperformed other models in bullish market conditions. The results of MSE produced diametrically opposed conclusions. Moreover, based on accuracy, the dataset with Tweets as the attractiveness measure outperformed Google Trends, whereas based on MSE the results did not differ significantly. Finally, using a lower number of hidden neurons, as well as lower learning rate values and lower batch size values yielded optimal results. Future research direction

includes forecasting the cryptocurrency returns using the sentiment-enriched data. Additionally, since cryptocurrency prices are increasingly reactive to macroeconomic shocks, it is proposed to examine which macroeconomic variables have the greatest influence on their movement. Finally, sophisticated neural network models can be used for the prediction of cryptocurrency prices, returns, direction and volatility in a comprehensive manner.

Funding

This work is fully supported by the Croatian Science Foundation (CSF) under the project “Challenges of Alternative Investments” [IP-2019-04-7816].

References

- 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*, 4(11), 130–137. <https://doi.org/10.22161/ijaers.4.11.20>.
- Aljinović, Z., Marasović, B., Šestanović, T. (2021). Cryptocurrency portfolio selection—a multicriteria approach. *Mathematics*, 9(14), 1677. <https://doi.org/10.3390/math9141677>.
- Aljinović, Z., Šestanović, T., Škrabić Perić, B. (2022). A new evidence of the relationship between cryptocurrencies and other assets from the covid-19 crisis. *Journal of Economics / Ekonomicky casopis*, 70(7–8), 603–621. <https://doi.org/10.31577/ekoncas.2022.07-8.03>.
- Azari, A. (2019). Bitcoin Price Prediction: An ARIMA Approach. [arXiv:1904.05315](https://arxiv.org/abs/1904.05315).
- Bai, J., Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, 66, 47–78. <http://www.jstor.org/stable/2998540>.
- Bao, W., Yue, J., Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, 12(7), e0180944. <https://doi.org/10.1371/journal.pone.0180944>.
- Cavalli, S., Amoretti, M. (2021). Cnn-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing*, 101, 107065. <https://doi.org/10.1016/j.asoc.2020.107065>.
- 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, 112395. <https://doi.org/10.1016/j.cam.2019.112395>.
- Čorić, R., Matijević, D., Marković, D. (2023). PollenNet – a deep learning approach to predicting airborne pollen concentrations. *Croatian Operational Research Review*, 14(1), 1–13. <https://doi.org/10.17535/crorr.2023.0001>.
- Diebold, F., Mariano, R. (1995). Comparing predictive accuracy. *Journal of Business Economic Statistics*, 13, 253–263. <https://doi.org/10.1080/07350015.1995.10524599>.
- Fahmi, A., Samsudin, N., Mustapha, A., Razali, N., Ahmad Khalid, S.K. (2018). Regression based analysis for bitcoin price prediction. *International Journal of Engineering & Technology*, 7(4.38), 1070–1073. <https://doi.org/10.14419/ijet.v7i4.38.27642>.
- Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33, 917–963. <https://doi.org/10.1007/s10618-019-00619-1>.
- Franses, P.H., van Dijk, D. (2003). *Non-Linear Time Series Models in Empirical Finance*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511754067>.
- Greaves, A., 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].
- Han, L.H.N., Hien, N.L.H., Huy, L.V., Hieu, N.V. (2024). A deep learning model for multi-domain MRI synthesis using generative adversarial networks. *Informatica*, 35(2), 283–309. <https://doi.org/10.15388/24-INFOR556>.
- Hegde, J., Rokseth, B. (2020). Applications of machine learning methods for engineering risk assessment – a review. *Safety Science*, 122, 104492. <https://doi.org/10.1016/j.ssci.2019.09.015>.

- Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8).
- Hwang, H. (2001). Insights into neural-network forecasting of time series corresponding to ARMA(p, q) structures. *Omega*, 29(3), 273–289. [https://doi.org/10.1016/S0305-0483\(01\)00022-6](https://doi.org/10.1016/S0305-0483(01)00022-6).
- Indera, N., Yassin, I., Zabidi, A., Rizman, Z. (2017). Non-linear autoregressive with exogeneous input (NARX) bitcoin price prediction model using pso and moving average technical indicators. *Journal of Fundamental and Applied Sciences*, 9(3S), 791–808. <https://doi.org/10.4314/jfas.v9i3s.61>.
- 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*, 6, 5427–5437. <https://doi.org/10.1109/ACCESS.2017.2779181>.
- Jaquart, P., Köpke, S., Weinhardt, C. (2022). Machine learning for cryptocurrency market prediction and trading. *The Journal of Finance and Data Science*, 8, 331–352. <https://doi.org/10.1016/j.jfds.2022.12.001>.
- Ji, S., Kim, J., Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. *Mathematics*, 7(10), 898. <https://doi.org/10.3390/math7100898>.
- Kalinić Miličević, T., Marasović, B. (2023). What factors influence bitcoin's daily price direction from the perspective of machine learning classifiers? *Croatian Operational Research Review*, 14(2), 163–177. <https://doi.org/10.17535/corr.2023.0014>.
- Lahmiri, S., Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, 118, 35–40. <https://doi.org/10.1016/j.chaos.2018.11.014>.
- Li, Y., Dai, W. (2020). Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. *The Journal of Engineering*, 2020(13), 344–347. <https://doi.org/10.1049/joe.2019.1203>.
- Liu, Y., Tsyvinski, A. (2020). Risks and returns of cryptocurrency. *The Review of Financial Studies*, 34(6), 2689–2727. <https://doi.org/10.1093/rfs/hhaa113>.
- Livieris, I.E., Kiriakidou, N., Stavroyiannis, S., Pintelas, P. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. *Electronics*, 10(3), 287. <https://doi.org/10.3390/electronics10030287>. <https://www.mdpi.com/2079-9292/10/3/287>.
- Madaeni, F., Chokmani, K., Lhissou, R., Homayouni, S., Gauthier, Y., Tolszczuk-Leclerc, S. (2022). Convolutional neural network and long shortterm memory models for ICE-JAM predictions. *The Cryosphere*, 16(4), 1447–1468. <https://doi.org/10.5194/tc-16-1447-2022>.
- Moshiri, S., Cameron, N. (2000). Neural network versus econometric models in forecasting inflation. *Journal of Forecasting*, 19(3), 201–217. [https://doi.org/10.1002/\(sici\)1099-131x\(200004\)19:3<201::aid-for753gt;3.0.co;2-4](https://doi.org/10.1002/(sici)1099-131x(200004)19:3<201::aid-for753gt;3.0.co;2-4).
- Pabuccu, 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. <https://doi.org/10.3934/qfe.2020031>.
- Paranhos, L. (2021). Predicting Inflation with Neural Networks. [arXiv:2104.03757](https://arxiv.org/abs/2104.03757).
- Patterson, D.W. (1998). *Artificial Neural Networks: Theory and Applications*, 1st ed. Prentice Hall PTR, USA.
- 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://doi.org/10.1080/10864415.2016.1061413>.
- Poyser, O. (2017). Exploring the Determinants of Bitcoin's Price: An Application of Bayesian Structural Time Series. [arXiv:1706.01437](https://arxiv.org/abs/1706.01437).
- Purwandari, K., Sigalingging, J.W.C., Cenggoro, T.W., Pardamean, B. (2021). Multi-class weather forecasting from twitter using machine learning approaches. *Procedia Computer Science*, 179, 47–54. <https://doi.org/10.1016/j.procs.2020.12.006>.
- Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M. (2020). Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019. [arXiv:1911.13288](https://arxiv.org/abs/1911.13288).
- Sovbetov, Y. (2018). Factors influencing cryptocurrency prices: evidence from bitcoin, ethereum, dash, litcoin, and monero. *Journal of Economics and Financial Analysis*, 2(2), 1–27.
- Spilak, B. (2018). *Deep Neural Networks for Cryptocurrencies Price Prediction*. PhD thesis.
- Šestanović, T. (2021). Bitcoin direction forecasting using neural networks. In: *The 16th International Symposium on Operational Research SOR'21, Proceedings*, pp. 557–562.
- Šestanović, T. (2024). A comprehensive approach to Bitcoin forecasting using neural networks. *Ekonomski pregled*, 75(1), 62–85. <https://doi.org/10.32910/ep.75.1.3>.
- Šestanović, T., Arnerić, J. (2020). Neural network structure identification in inflation forecasting. *Journal of Forecasting*, 40(1), 62–79. <https://doi.org/10.1002/for.2698>.

- Šestanović, T., Kalinić Miličević, T. (2023). A MCDM approach to machine learning model selection: Bitcoin return forecasting. In: *Proceedings of the 17th International Symposium on Operational Research in Slovenia SOR'23*, Ljubljana, University of Maribor, pp. 77–82.
- Trimborn, S., Li, M., Härdle, W.K. (2019). Investing with cryptocurrencies – a liquidity constrained investment approach. *Journal of Financial Econometrics*, 18(2), 280–306. <https://doi.org/10.1093/jfinec/nbz016>.
- Uras, N., Marchesi, L., Marchesi, M., Tonelli, R. (2020). Forecasting bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science*, 6, e279. <https://doi.org/10.7717/peerj-cs.279>.
- 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 and Money*, 63, 101133. <https://doi.org/10.1016/j.intfin.2019.101133>.
- Zhang, Z., Dai, H.N., Zhou, J., Mondal, S.K., García, M.M., Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications*, 183, 115378. <https://doi.org/10.1016/j.eswa.2021.115378>.

T. Šestanović obtained her PhD in economics in 2017. She is currently an assistant professor at University of Split, Faculty of Economics, Business and Tourism. She is teaching statistics and similar courses on all levels of studies, as well as business decision making. She is a president of Croatian Operational Research Society (CRORS) and an editor-in-chief of *Croatian Operational Research Review (CRORR)*. Her main scientific interests are time series, neural networks, financial modelling and statistics.

T. Kalinić Miličević is a teaching assistant at University of Split, Faculty of Economics, Business and Tourism (FEBT). She graduated in mathematics from University of Split, Faculty of Science, and she finished postgraduate specialist study program in business economics on FEBT. She is teaching mathematics, quantitative methods, financial modelling and actuarial analysis. She is a treasurer at Croatian Operational Research society (CRORS). Her main scientific interests are machine learning models, financial modelling, actuarial science and optimization.