08, 09, 10 e 11 de novembro de 2022
ISSN 2177-3866

A bidirectional LSTM model for cryptocurrency prices forecasting

GUSTAVO FRANZENER GONÇALVES DA SILVA
UNIVERSIDADE DE SÃO PAULO (USP)

LEANDRO DOS SANTOS MACIEL
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE DA UNIVERSIDADE DE SÃO PAULO - FEA

Agradecimento à orgão de fomento:
This work was supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under grants 304456/2020-9, and by the Ripple Impact Fund, a donor advised fund of the Silicon Valley Community Foundation, grant 2018-196450(5855) as part of the University Blockchain Research Initiative, UBRI.

A bidirectional LSTM model for cryptocurrency prices forecasting

1. Introduction

Cryptocurrencies were firstly introduced by Nakamoto (2009), that proposed a digital currency and a payment system based on cryptographic proof instead of the trust of third parties such as financial institutions. In this way, mediation costs are avoided, potentially reducing transaction costs and preventing double-spending. The implementation of a cryptocurrency would use a per-to-per network which timestamps transactions into an ongoing chain, requiring minimal structure, forming a record that cannot be changed and is secure as long as honest nodes collectively control more CPU power than any group of attacker nodes. This electronic cash was named Bitcoin and still stands as the main and more traded cryptocurrency in the market. Since the creation of Bitcoin, a big number of cryptocurrencies were created and gained significant share of this emerging market.

Cryptocurrency market prices are marked by a high volatility, presenting opportunities to investors searching for higher returns and diversification. This gives an important role in predicting its futures prices to enable trading strategies and providing information to investors. Several studies were recently made comprising which factors could affect a cryptocurrency price, how to predict Bitcoin futures prices using as input economic and technological determinants (Guo, Zhang, Liu, Wang, & Ding, 2021; Liu, Li, Li, Zhu, & Yao, 2021; W. Chen, Xu, Jia, & Gao, 2021; Cavalli & Amoretti, 2021), previous lagged values (Nakano, Takahashi, & Takahashi, 2018), and concerning the implementation of trading systems (Silva de Souza et al., 2019). However these articles are mainly focused on the Bitcoin price prediction only, with a limited number of them embracing a wider number of currencies such as the work of Alonso-Monsalve, Suárez-Cetrulo, Cervantes, and Quintana (2020).

In the field of price prediction, machine learning algorithms were proposed by several papers, given their ability to capture and appropriate modeling the non-linearities of complex time series. Hu, Tang, Zhang, and Wang (2018), Chalvatzis and Hristu-Varsakelis (2020) and Peng, Albuquerque, Kimura, and Saavedra (2021) applied different approaches using these algorithms to predict stocks prices directions and implement trading architectures. Bidirectional LSTM (BiLSTM) models, primarily used on natural language processing, are still weakly explored on its use for financial time series prediction but already presented promising results, as observed in the work of Yang and Wang (2022). This kind of models distinguishes itself by considering the influence of future factors on the present, in a way that the time series are trained on both directions. Testing the application of this architecture on forecasting cryptocurrency closing prices is a important contribution of this paper. Varied features can be used as input to those models, such as lagged values. The search engine Google enables the access of the relative volume of queries over time via Google Trends platform which could also be used as input for forecasting methods. Using this available data is possible to observe patterns between the analyzed asset and its popularity on the internet. As analyzed by Preis, Moat, and Stanley (2013) on stocks prices movements not only the search trends data reflects price changes but also can be used to anticipate future movements.

This paper aims to evaluate the one-step prediction of 10 cryptocurrencies on a daily frequency using neural network models, especially a bidirectional LSTM neural network. These cryptocurrencies are Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB), Cardano (ADA), Ripple (XRP), Dogecoin (DOGE), Litecoin (LTC), Chainlink (LINK), Tron (TRX) and Stellar (XLM). The neural networks algorithms implemented were Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM) and BiLSTM. The results obtained were compared with results of

state of the art models, such as Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ETS), and with a Random Walk (RW) method. Previous exchange rate (lagged values) are used as input to all the models. A BiLSTM approach is tested combining Google Trends data as input.

After this brief introduction, this paper is organized as follows. Section 2 describes a literature review, detailing other papers achievements on the use of machine learning for financial prediction and then more specifically for cryptocurrencies price prediction. The methodology is described in Section 3, followed by the results discussed in Section 4. Section 5 concludes the work and presents topics for further investigation.

2. Literature review

Machine learning and more specifically neural network models have been intensively applied in the financial market literature. In recent years, different architectures were proposed and analyzed such as in Hu et al. (2018), that introduced the ISCA-BPNN by combining improved sine cosine algorithm (ISCA) and back propagation neural networks (BPNN) to predict the direction of stocks opening prices on the S&P 500 and Dow Jones Industrial Average Indices (DJIA), from January 2010 to June 2017. Google Trends data was also took into consideration for improving stock prediction. The ISCA-BPNN model presented better results than other five models in predicting the direction of the opening price, demonstrating that it is capable of predicting stock prices and that Google trends can help forecast financial returns.

Another method was proposed by Chalvatzis and Hristu-Varsakelis (2020), via an automated trading architecture in which the prediction model was tuned to enhance profitability instead of accuracy. The proposed approach was tested for major U.S. stock indices (S&P 500, DJIA, NASDAQ and Russel 2000) from January 2010 to December 2019 and compared with four different models: Random Forest (RF), gradient boosted trees and two variations of LSTM. It outperformed the indices and compared models in terms of the cumulative or annualized returns, leading to better long-term risk profile than those of the stock indices (buy-and-hold) but more volatile than some in the recent literature.

Deep neural network models were applied by Peng et al. (2021), that discussed feature selection with different architectures and regularization parameters to predict stock prices directions. A set of 124 technical analysis indicators were submitted to three feature selection methods eliminating redundant information. Using daily data from stocks of seven global market indexes between January 2008 and March 2019, the neural networks were tested and compared by their accuracy, precision, recall, and F-Score, and taking into account profitability and transaction costs levels to analyze economic gains. The profitability of the strategies did not manage to significantly outperform the Buy-and-Hold strategy, even showing fairly large negative values for some hyper parameter combinations, representing that not always neural network models can succeed and the importance of comparisons between different methods.

Machine learning models were also evaluated concerning cryptocurrencies price forecasting. Nakano et al. (2018) used multi-layer neural networks for BTC price direction prediction from technical indicators using high frequency data from July 2016 to January 2018. Trading strategies were applied to buy (sell) when the price is likely to go up (down) and then compared in terms of risk-return measures considering transactions costs. The proposed model presented higher performance than buy-and-hold approach and trading strategies based only on the time series of returns. Trading strategies were also investigated by Silva de Souza et al. (2019), using price direction forecasts obtained from Support Vector Machines (SVM) and three-layer Artificial Neural Networks (ANN) models for BTC, Gold and Silver prices. The models used daily closing prices as inputs from July 2012 to April 2017. Results were compared with a

simple buy-and-hold strategy in terms of profitability and risk performance indicators. ANN presented great potential to generate abnormal returns in short run bull trends even accounting for transaction costs.

The suitability of neural networks with a convolutional component for trend classification of cryptocurrency exchange rates were investigated by Alonso-Monsalve et al. (2020). Six popular cryptocurrencies were considered (BTC, Dash, Ether, Litecoin, Monero, and Ripple) from July 2018 to June 2019 using 18 technical indicators at a one-minute frequency. Four prediction models were analyzed: Convolutional Neural Networks (CNN), hybrid CNN-LSTM networks, MLP and Radial Basis Function Neural Networks (RBFNN). The results show that CNN and, especially, CNN-LSTM are suitable as predictors for the price trend of most cryptocurrencies, especially for BTC, Ether and Litecoin. Results of the CNN-LSTM architecture were significantly better than the remaining techniques, consisting on the only model that could predict the trends of Dash and Ripple.

Z. Chen, Li, and Sun (2020) leveraged machine learning techniques focusing on the feasibility of applying different modeling techniques to samples with different data structures and dimensional features. A binary classification algorithm was developed to predict the sign change of BTC price. Two datasets were employed: the first included BTC daily price, network data, trading and market data, media and investor attention and gold spot price, from February 2017 to February 2019. The second one consisted of 5-minute interval BTC trading price from July 2017 to January 2018. The most useful and meaningful features were selected for the prediction models. Two statistical methods were implemented, logistic regression (LR) and linear discriminant analysis (LDA), while the machine learning models used were RF, XGBoost (XGB), quadratic discriminant analysis (QDA), SVM and LSTM. The results showed that the statistical methods perform better for low-frequency data with high-dimensional features, while the machine learning models outperform statistical methods for high-frequency data.

Guo et al. (2021) proposed and evaluated a new price forecasting model, WT-CATCN, based on the Wavelet Transform (WT) and a Casual Multi-Head Attention Temporal Convolutional Network (CATCN), using selected and analyzed features and demonstrating the predictive power of the volume difference between big and small exchanges. Inter-exchange Transactions, Inner-exchange Market Prices and Google Trends data from January 2016 to December 2018 at a daily frequency were selected as input. The WT-CATCN was compared with baseline and state-of-the-art methods in terms of closeness and direction metrics. The correlations between the volumes of different exchanges indeed contributed to the price forecasting, while WT-CATCN model outperformed the state-of-the-art models by at least 25% in terms of Root Mean Square Error (RMSE).

Ibrahim, Kashef, and Corrigan (2021) compared various BTC price prediction models over short time-frames to predict the direction of price movement. The tested models included ARIMA, Prophet, RF, RF Lagged-Auto-Regression, and MLP. BTC information was collected in the form of tick-data dating back to 2014 and transformed into 5-min intervals and in the format of 5-min Open High Low Close (OHLC) plus volume dating back to September 2017. 5-min OHLC data for Apple, Facebook, Google, and Microsoft stocks dating back to January 2018 was also gathered. The MLP predictions outperformed all of the tested models, with 54% accuracy, while Prophet achieved better accuracy than the ARIMA and RF models. Koo and Kim (2021) focused on daily price prediction of BTC based on MLP, Recurrent Neural Networks (RNN), and LSTM, with the application of the Flattening Distribution Strategy (FDS). BTC data was analyzed from April 2013 to April 2020 with six types of data (volume of business, market capitalization and prices of BTC in terms of open, high, low, close). The FDS artificially manipulates the concentrated return into the uniform distributed data. Consequently,

it alleviated the sensitivity of the neural networks and improved the prediction performances. Besides the benefit of improvement in accuracy, it turns out that the FDS reduced the error spread and enlarged the overlapping region between the label and the prediction.

Liu et al. (2021) constructed a feature system with 40 determinants that affects the price of BTC considering aspects of the cryptocurrency market, public attention and the macroeconomic environment. A deep learning method named Stacked Denoising AutoEncoders (SDAE) was utilized to predict the price of BTC from July 2013 to December 2019. SDAE performed better in both directional and level prediction compared with popular methods, such as BPNN and SVR. Jaquart, Dann, and Weinhardt (2021) analyzed the predictability of the BTC market across prediction horizons ranging from 1 to 60 min using technical, blockchain-based, sentiment-/interest-based, and asset-based features ranging from March 2019 to December 2019 at 1-minute frequency. Various machine learning models were tested, finding that, while all models outperform a random classifier, recurrent neural networks and gradient boosting classifiers are especially well-suited for the examined prediction tasks.

Poongodi, Nguyen, Hamdi, and Cengiz (2021) investigated cryptocurrency price movement trends using social media data. Data was assembled from the bitcointalk.org forums and verifiable BTC value trade information, from April 2011 to May 2018. A neural system was developed via Keras API and to quantify the execution of the model, estimations of misfortune work in mean squared blunder and the R squared score were applied. The model predicted the snapshots of increments and downturn of the cost exceptionally well and confirmed that BTC trend prediction is possible given social media data. W. Chen et al. (2021) evaluated the prediction of BTC using LSTM with economic and technological determinants as inputs, previously selected with ANN and random forest models. The results were compared with methods based on lagged BTC exchange rates as inputs, such as LSTM, ANFIS (Adaptive Network Fuzzy Inference System), ARIMA and SVR. Data comprised BTC daily exchange rate, blockchain information, public attention measures, financial indexes, currencies ratios, crude oil and gold prices. The period from August 2011 to July 2018 was divided into four different samples to evaluate and select the forecasting determinants by their relevance for each period. The use of economic and technology determinants as inputs in the LSTM lead to a better performance in terms of statistical accuracy indicators when compared to the competing alternatives.

Cavalli and Amoretti (2021) suggested an approach for BTC trend prediction based on a One-Dimensional CNN. The proposed model predicted whether the BTC value after n days would be lower or higher than the latest value of the time series. Data was considered on a daily frequency from April 2013 to February 2020. The work proposed a methodology for building datasets used as inputs to the predictor whose items were characterized by different types of features: BTC historical values and financial indicators, Twitter sentiment analysis and BTC blockchain information. The CNN presented higher performance compared to LSTM models in terms of binary accuracy. Applying a trading strategy based on the proposed CNN model forecasts lead to an increase on the profit when the BTC trend was bullish and a reduction in losses when the trend was bearish. Hence, it is clear that prediction benefits can be achieved by the use of machine leaning models for high volatile markets, as for the digital coins, demanding the evaluation for a large number of cryptocurrencies, which is the aim of this work. 0

3. Methodology

3.1 Data

Data comprising ten of the most relevant cryptocurrencies are selected between the ones with available data. The following criteria is applied in the selection process, considering the

variables at January 25, 2022.

a. Ordering the cryptocurrencies by market capitalization (in USD).

b. Selecting the first 40 currencies.

c. Excluding the currencies with less than four years of available exchange data.

d. Excluding stablecoins (whose market value is attached to another stable asset) and currencies presenting a too strong correlation with better ranked ones, such as the cased of Wrapped Bitcoin (wBTC).

After this process, the ten first cryptocurrencies are selected. They are: Bitcoin, Ethereum, Binance Coin, Cardano, Ripple, Dogecoin, Litecoin, Chainlink, Tron and Stellar.

Data gathered from these cryptocurrencies comprehends its closing price at a daily frequency, starting at January 1, 2018 and ending at April 30, 2022. Figure 1 presents the evolution of the daily closing prices on the analysed period for the cryptocurrencies. The exchange rates are obtained at the CoinMarketCap (2022) website and are available on the tab labeled as "Historical Data" for each currency.

For the BiLSTM model including Google Trends, the use of python pytrends API is necessary. It allows automating the download of reports from Google Trends, enabling the acquisition of daily trends data. The presented names of each currency are separately used as search terms on the API for the same time range and frequency of the closing prices.

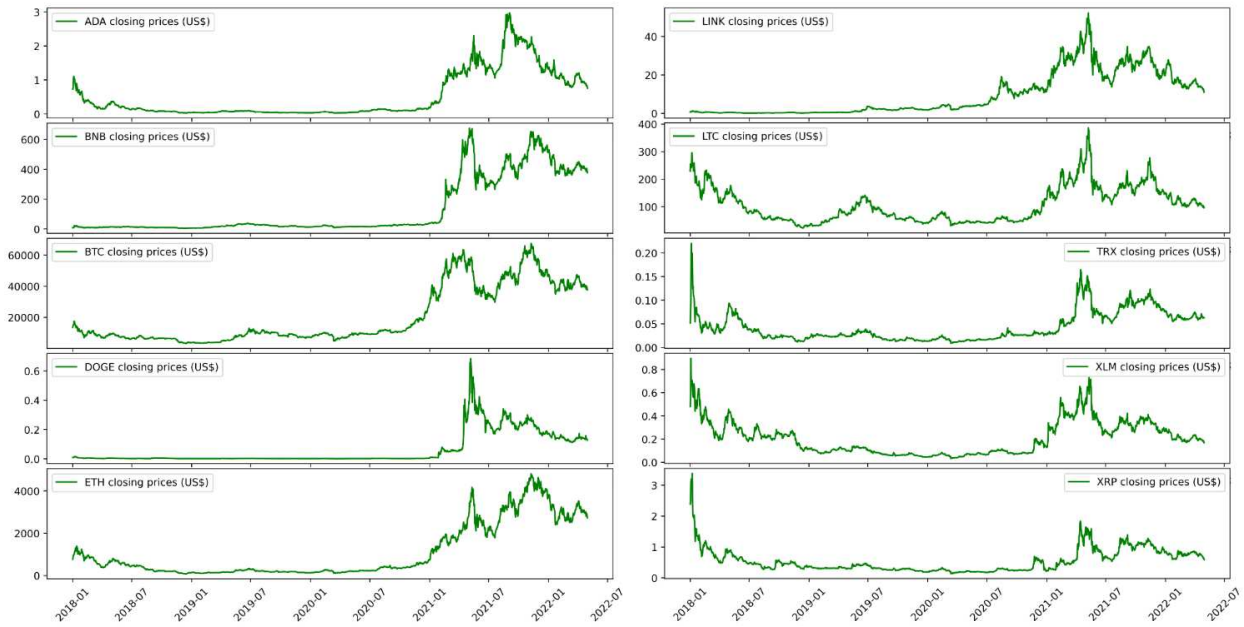

Fig. 1: Historical closing prices.

These values pass through data preparation, cleaning, organizing and visualizing the data. The resulting time series are each of then divided in two samples including training and test sets in the way presented on Table 1.

Tab. 1: Sample division.

	Train		Test	
	Start	End	Start	End
Sample1	January 1, 2018	April 30, 2020	May 1, 2020	April 30, 2021
Sample2	January 1, 2019	April 30, 2021	May 1, 2021	April 30, 2022

3.2 ARIMA

Autoregressive moving average (ARMA) models are composed by two parts. In the autoregressive part (1), the evolving variable of interest is regressed on their previous values. The moving average (2) indicates that the regression error is a linear combination of errors terms that occurred at different times in the past. Besides that, c represents a constant and e_t an aleatory error (white noise).

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + e_t \tag{1}$$

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + ... + \theta_q e_{t-q} \tag{2}$$

ARMA models requires stationary of the time series, which narrow down its applications. The ARIMA models are introduced to solve this problem, by differencing the time series to make it stationary, generalizing the ARMA approach to a wide range of time series. A general ARIMA model (3) can be expressed as ARIMA (p, d, q) where p represents the number of auto-regressive terms, q is the number of non-seasonal differences needed for stationary and d is the number of lagged forecast errors in the prediction equation.

$$\acute{y_t} = c + \phi_1 \acute{y}_{t-1} + ... + \phi_p \acute{y}_{t-p} + \theta_1 + \theta_1 e_{t-1} + ... + \theta_q e_{t-q} + e_t \tag{3}$$

The function auto.arima from forecast package in R is the chosen function to define the parameter combination (p, d, q) for each training set. auto.arima works by firstly conducting differencing tests. As default choice from the function, the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test is used to determine the order of differencing, d. The models are then fitted within ranges of defined p and q, a maximum of 10 is defined for each parameter in these experiments. The seasonal option is enabled but no seasonality is detected in any sample, as expected, showing no needs to apply a Seasonal Autoregressive Integrated Moving Average (SARIMA) model. A drift parameter is also included.

In order to find the best model, auto.arima considers the selected information criterion, Corrected Akaike Information Criterion (AICc), and returns the ARIMA model which minimizes its value. The AICc works by preventing the possibility of overfitting when using the Akaike Information Criterion (AIC). While AIC is calculated as on equation (4), with k being the estimated number of parameters and $\hat{L}$ the maximum value of the likelihood function for the model, AICc includes a correction (5) with n denoting the sample size and k the number of parameters.

$$AIC = 2k - 2 \cdot ln(\hat{L}) \tag{4}$$

$$AICc = AIC + \frac{2k(k+1)}{n-k-1} \tag{5}$$

The models are estimated and its residuals are submitted to the Ljung-Box test. The null hypothesis of Ljung-Box test is: data is independently distributed. If the p-value is larger than the specified significance level of 0.05, the null hypothesis cannot be rejected. In other words, there is no clear sign of autocorrelations and the model is valid.

An ARIMA (0,1,0) without intercept, representing a Random Walk, is also estimated for all the cryptocurrencies. On this approach, the time series is purely predicted based on the previous time point $(t-1)$. Its results are used as benchmark for the obtained models.

3.3 ETS

Exponential smoothing (ETS) is based on weighted averages of past observations, with the

weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight (Hyndman & Athanasopoulos, 2018). This framework generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry. A simple exponential smoothing is expressed on (6), where $0 \leq \alpha \leq 1$ is the smoothing parameter.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots \tag{6}$$

ETS presents up to 18 variations, that may include trend and seasonal components, as on Holt-Winters' method, represented on equation (7), with b_t and s_t representing the additive trend and seasonal components respectively, with smoothing parameters for trend $0 \leq \beta^* \leq 1$ and seasonality $0 \leq \gamma \leq 1-\alpha$, m as the frequency of the seasonality, k as the integer part of $(h-1)/m$ and ℓ_t the level. Damped trends may also apply, which makes the trend to a flat line some time in the future by introducing a damping parameter $0 < \phi < 1$. Each method is referred as ETS(*cdot*,*cdot*,*cdot*) for (Error, Trend, Seasonal). This label can also be thought of as ExponenTial Smoothing. The possibilities for each component are: Error =A,M, Trend =N,A,Ad and Seasonal=N,A,M, M representing representing multiplicative, A for additive, N for none and Ad is damped additive. The error component interferes only on the prediction interval, which is not availed on these experiments, as the focus are on point forecasts.

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t + s_{t+h-m(k+1)} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}. \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned} \tag{7}$$

The models are estimated in R using the ets() function in the forecast package. The ets() can automatically estimates the model parameters and returns information about the fitted model, only requiring the time series as argument. The appropriate models are selected by their AICc (5).

3.4 MLP

MLP consists on a feedforward Artificial Neural Network (ANN) that can be used to solve classification and regression problems by simulating simplified biological neurons called perceptrons. It is composed by one input layer, which simply distributes the input features to the first hidden layer, one or more hidden layers, each of them receives input from the previous layer, and one output layer, that receives inputs from the outputs of all the neurons on the last hidden layer. The neurons on these layers are fully connected by connection weights as presented on Figure 2. This structure enables the mapping of multiple input datasets to output datasets and the learning of nonlinear functions. The accumulation of weighted values of a neuron can be expressed as on equation (8) for n inputs.

$$u(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + bias \tag{8}$$

The results of the weighted values are then passed to an activation function which generates the output of the perceptron. Rectified Linear Unit (ReLU) is the selected activation function for these networks and is represented on. ReLU functions help to achieve fast convergence and are calculated as $f(x) = max(0,x)$.

A MLP with 3 hidden layers - named as H1, H2 and H3 - is selected to model the cryp-

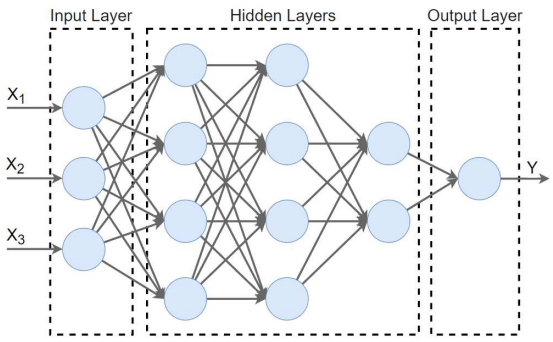

Fig. 2: One of the tested MLP architectures.

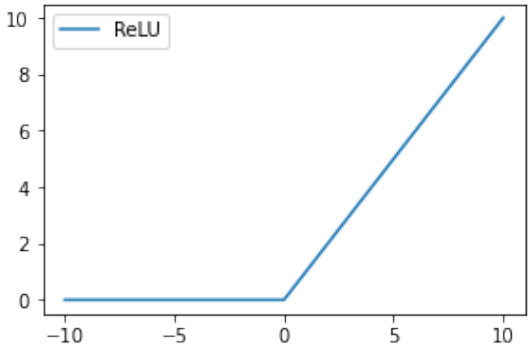

Fig. 3: ReLU activation function.

tocurrencies closing prices. For each layer a set of 4 different possibilities are established for the number of nodes: 2, 4, 6 and 8 nodes. The batch size is also part of the parameter grid with possibilities of 16, 32, 64 and 128. To fit the models during the Bayesian optimization 4 different numbers of epochs are proposed: 2, 5, 10 and 20. The implementation of the network is via Keras API for python, using a sequential model with dense layers.

Data is normalized to be fed to all the implemented ANN models via the scikit-learn package function MinMaxScaler to be comprehended between 0 and 1. Then it is divided for supervised learning with a number of 3 time lags, which resulted on the best validation results for most of the currencies, representing the features and a single output value as the one-step prediction. The same time lag value is applied to all the ANN models. Adam algorithm is selected for model optimization. Adam is a stochastic gradient method widely used given its computational efficiency and suitability for problems with high number of data. The default learning rate of 0.001 is applied to the Adam optimizer. Loss function to monitor the fit of the models is the MSE (18).

3.5 Bayesian Optimization

An heuristic approach is responsible for a directed search across possible configurations. Selecting the model hyper parameters is fundamental to an optimal working of ANNs. During the conducted experiments a Bayesian optimization algorithm (Frazier, 2018) is applied to automatize the selection process between a chosen grid. The hyper parameters are optimized using an implemented cross validation split, referred as Blocked Time Series Split. This cross-validation method works by adding 2 different limits to the next sample, a initial one between the training and validation folds and a second one between the folds used at each iteration, preventing the model from memorizing patterns of different iterations. The number of cross-validation iterations is defined to 3 and a representation of the defined approach is at Figure 4.

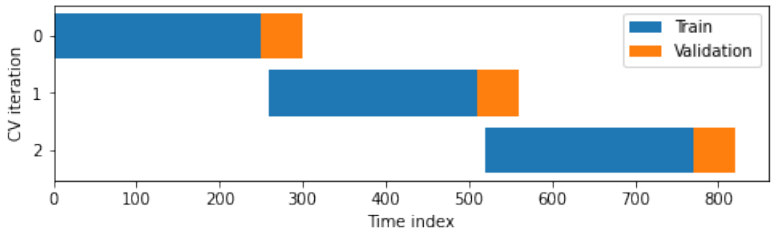

Fig. 4: Blocked Time Series Split.

The parameters selected are those that maximize the score of the held-out data, according to the model's loss function. In contrast to a traditional grid search, not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions. For this, the Bayesian optimization uses Gaussian Process regression on the objective function, aiming to find its global minimum. For this study the objective function is to achieve the minimum loss for the given model parameters. It is applied via the Sequential model-based optimization package (scikit-optimize).

3.6 Early stopping

In order to prevent overfitting, a regularization method is necessary. Early Stopping stops training the model when a monitored metric stops improving. In this case, a validation set corresponding to the last 20% of the training set is hold out to avail the quality of the model by its validation loss. In that way, Early Stopping avoids the problem of choosing the number of training epochs to use.

Keras provides an EarlyStopping class which was applied for the models chosen by the Bayesian Optimization with a patience, the number of epochs to wait for an improvement in the model, of 30. A ModelCheckpoint callback is also applied, saving the best model in terms of validation loss for the training epochs.

3.7 LSTM

Introduced by Hochreiter and Schmidhuber (1997), LSTM networks addresses Recurrent Neural Networks (RNN) memory problem. RNN are applied to recognize patterns when past results compounds the actual result, as the hypotheses for the studied time series. However, earlier results are easily forgotten, limiting its application for the cases with influence of more than the immediate past.

LSTM solves this question with a change in the network architecture. Each neuron now presents 3 gates - input gate, output gate and forget gate (Figure 5), each of them with its own function. Equation (9) represents the forget gate, that passes the information, while equation (10) represents the input gate and equation (11) the output gate. For these calculations, σ is the sigmoid function, W_x is the weight for the respective gate, h_{t-1} the output of the previous LSTM block, x_t the current input and b_x the biases for the gates.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{9}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{10}$$

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{11}$$

The cell state is defined by equation (12), while equation (13) represents the candidate value for cell state and equation (17) the final information as the output for the cell.

$$C_t = f_t.C_{t-1} + i_t.\widetilde{C}_t \tag{12}$$

$$\widetilde{C}_t = ReLU(W_c.[h_{t-1}, x_t] + b_c) \tag{13}$$

$$h_t = O_t.ReLU(C_t) \tag{14}$$

To this research purposes a so called Vanilla LSTM is applied, which presents only one LSTM layer, that leads to optimal results and avoids high computational cost, characteristic of

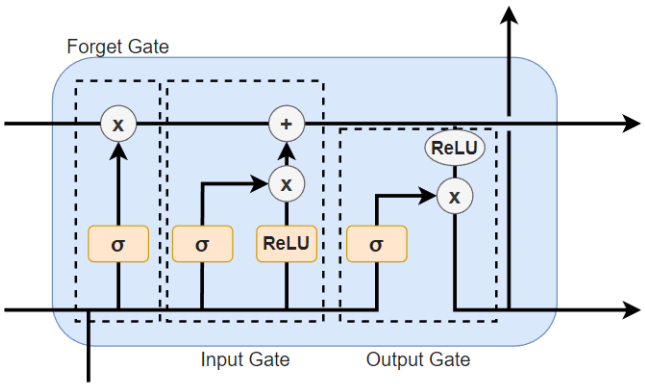

Fig. 5: LSTM neuron structure.

training these networks. Adam optimizer is applied, changing the learning rate to 0.01. LSTM applies hyperbolic tangent (tanh) as default activation function, but for these experiments ReLU is applied, leading to better results. The network is applied via Keras API, using a sequential model with LSTM and Dense layers. The described Bayesian Optimization and Early Stopping are also used in the LSTM models, selecting number of neurons (2,4 or 8), batch size (16, 32, 64 or 128) and number of epochs to fit the model in the optimization (2, 5, 10 or 20).

3.8 Bidirectional LSTM

Differently from traditional one directional LSTM, the Bidirectional LSTM, proposed by Graves and Schmidhuber (2005), is trained not only from inputs to outputs, but also from outputs to inputs. In summary, a BiLSTM model adds an extra LSTM layer to each of the previous ones. The forward layer (15) is first feed input data to an LSTM model and then repeat the training via another LSTM model on the backward layer (16) but on the reverse order of the sequence of the input data (Figure 6). The equations (15) and (16), with b_x and W_x representing the biases and weights, are combined to generate the output (17).

$$\overrightarrow{h_t} = ReLU\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h_{t-1}} + b_{\overrightarrow{h}}\right) \tag{15}$$

$$\overleftarrow{h_t} = ReLU\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h_{t+1}} + b_{\overleftarrow{h}}\right) \tag{16}$$

$$y_t = W_{\overrightarrow{\mathbf{h}}y}\overrightarrow{h_t} + W_{\overleftarrow{\mathbf{h}}y}\overleftarrow{h_t} + b_y \tag{17}$$

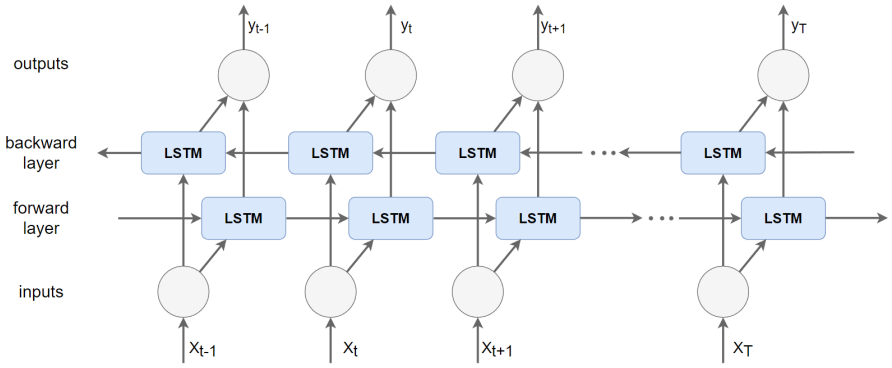

Fig. 6: BiLSTM structure.

For the implementation of this network, the Bidirectional layer wrapper from Keras API is applied over the LSTM layer for the sequential model. A model with two bidirectional hidden layer is utilized, with possibilities of 2, 4 and 8 numbers of nodes per layer, 16, 32, 64 and 128 numbers for batch size and 2, 5, 10 and 20 epochs to selection of the hyperparameters, realized by Bayesian Optimization. Early Stopping is also applied for the BiLSTM selected models.

The BiLSTM architecture is selected to the appliance of Google Trends data as exogenous variable, composing with the previous exchange rate a multivariate input. Besides the expected inputs, no other change on the methodology applied to previous BiLSTM, with only lagged values as input, is implemented with the addition of Google Trends data. As on the closing prices, Google Trends data is normalized and prepared to supervised learning, with 3 time lags, before being feed to the model.

3.9 Accuracy metrics

The forecast values are evaluated by 4 different metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Directional Accuracy (MDA). These metrics are applied by comparing the forecast values (F_t) to the actual ones (A_t) for all the n test values.

On MSE (18) the sum of the square errors is calculated and then divided by the number of observations. This gives a measure of how far the predicted values are from the actual ones, avoiding negative error values and mutual cancellation of errors. MSE emphasizes large errors due to the square used in its calculation.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2 \tag{18}$$

MAE (19) is quite similar with MSE but calculates the absolute errors instead of the square of the errors. In this way its less sensible to outliers but fails to punish large errors in prediction and might present difficulties on gradient calculation.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \tag{19}$$

MAPE (20) is the sum of the individual absolute errors divided by the actual values. It constitutes a relative measure, representing the average of the percentage errors. For MSE, MAE and MAPE, the lower are its values, better fit is the model.

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{20}$$

MDA (21) compares the forecast direction to the realized one. It works similarly to a binary evaluation, considering only the upward or downward direction and ignoring the quantitative value of increase or decrease. It uses the sign() function to determine the sign of the difference between actual/forecast and past values (t-1) and compare then for the test set. Opposing to the previous metrics a high MDA is aimed.

$$MDA = \frac{1}{n}\sum_{t}\mathbf{1}_{sign(A_t - A_{t-1}) == sign(F_t - A_{t-1})} \tag{21}$$

4. Results and discussion

Applying the described methodology the optimized configuration for each model is defined. The resulting combinations of parameters for ARIMA is presented on Table 2.

Tab. 2: Parameters selected by auto.arima. Models with drift are represented with 'T'.

	ADA		BNB		BTC		DOGE		ETH		LINK		LTC		TRX		XLM		XRP	
Sample	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
p	3	5	4	5	4	1	5	8	4	5	2	0	0	3	8	4	3	2	3	2
d	1	2	1	2	1	2	1	1	1	2	1	1	1	1	1	1	1	1	1	1
q	4	1	4	3	2	8	3	1	2	3	4	5	2	4	10	4	2	2	5	5
drift	T							T				T		T		T	T		T	

For the ETS, a multiplicative error and no seasonality are selected for all the samples by the ets() function. Trend parameters are presented on Table 3.

Tab. 3: Trend parameters selected by ETS function.

	ADA		BNB		BTC		DOGE		ETH		LINK		LTC		TRX		XLM		XRP	
Sample	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
Trend	Ad	A	Ad	A	N	A	N	A	Ad	A	Ad	A	N	N	Ad	N	Ad	N	N	N

The selected hyperparameters for MLP, LSTM, BiLSTM and Google Trends LSTM models for each cryptocurrency sample are presented respectively on Table 4, Table 5, Table 6 and Table 7.

Tab. 4: MLP hyperparameters selected by the Bayesian optimization. H1, H2 and H3 are the number of neurons on the hidden layers and Batch is the batch size.

	ADA		BNB		BTC		DOGE		ETH		LINK		LTC		TRX		XLM		XRP	
Sample	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
H1	4	4	8	4	8	4	8	4	6	6	6	8	8	4	4	2	8	8	4	2
H2	4	8	2	6	6	6	2	6	4	6	6	8	8	6	6	6	2	4	8	8
H3	4	8	6	4	6	4	6	6	8	8	8	4	8	8	6	8	6	8	8	2
Batch	32	16	32	16	16	32	32	32	16	16	128	16	16	16	16	16	16	16	16	32

Tab. 5: LSTM hyperparameters selected by the Bayesian optimization. H1 is the number of neurons on the hidden layer and Batch is the batch size.

	ADA		BNB		BTC		DOGE		ETH		LINK		LTC		TRX		XLM		XRP	
Sample	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
H1	2	8	4	8	8	8	8	4	4	2	2	8	8	8	4	2	4	8	8	8
Batch	32	16	32	16	64	16	128	16	128	128	32	16	32	16	64	64	16	64	16	16

Tab. 6: BiLSTM hyperparameters selected by the Bayesian optimization. H1 and H2 are the number of neurons on the hidden layers and Batch is the batch size.

	ADA		BNB		BTC		DOGE		ETH		LINK		LTC		TRX		XLM		XRP	
Sample	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
H1	2	4	8	4	4	8	4	4	2	8	8	8	4	8	4	2	4	2	4	2
H2	2	8	4	2	8	8	8	8	2	2	2	8	2	2	4	2	4	4	2	4
Batch	16	64	64	16	128	32	16	16	64	64	16	32	64	32	128	128	64	128	64	16

Tab. 7: Google Trends BiLSTM hyperparameters selected by the Bayesian optimization. H1 and H2 are the number of neurons on the hidden layers and Batch is the batch size.

	ADA		BNB		BTC		DOGE		ETH		LINK		LTC		TRX		XLM		XRP	
Sample	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
H1	4	8	2	4	4	4	8	8	8	8	4	2	4	2	8	8	8	8	8	4
H2	8	2	2	8	8	4	8	2	8	2	2	8	8	8	8	4	8	8	4	8
Batch	16	32	32	32	16	16	64	16	16	16	64	16	16	16	16	16	16	16	16	16

Using these configurations the results of Table 8 are obtained. Given the stochastic nature of ANNs, each result represents the mean of 20 experiments. Best results between all the models are highlighted in each row. Comparing only the ANNs, best results of each row are selected on bold.

Tab. 8: Evaluation of the models, the samples are described with the cryptocurrency abbreviation followed by the number of the sample.

Sample	Metric	RW	ARIMA	ETS	MLP	LSTM	BiLSTM	Trends BiLSTM
ADA-1	**MSE**	1,29E-3	1,46E-3	1,31E-3	4,18E-3	2,89E-3	3,32E-3	**2,23E-3**
	MAE	0,0171	0,0192	0,0172	31,16E-3	26,27E-3	28,21E-3	**22,28E-3**
	MAPE	4,3268	5,1449	4,2955	6,62E+0	5,82E+0	6,33E+0	**5,06E+0**
	MDA		0,4767	0,5014	0,5248	0,5247	0,5259	**0,6098**
ADA-2	**MSE**	9,21E-3	9,43E-3	9,26E-3	28,49E-3	25,16E-3	**21,61E-3**	40,65E-3
	MAE	0,0641	0,0653	0,0172	0,1146	0,1092	**0,1008**	0,1193
	MAPE	4,0682	4,1674	4,1260	6,7154	6,4034	**6,1065**	6,3541
	MDA		0,4712	0,4904	0,4688	0,4813	0,4834	**0,5897**
BNB-1	**MSE**	219,1255	221,4299	216,9641	1.038,3114	2.960,2470	692,2588	**581,9748**
	MAE	5,5721	5,5621	5,5209	12,2868	18,5934	9,5755	**8,8072**
	MAPE	3,8456	3,8232	3,8180	6,2643	7,5154	5,5483	**4,2557**
	MDA		0,5342	0,5507	0,5014	0,5374	0,5216	**0,7311**
BNB-2	**MSE**	497,2777	520,9576	520,9824	820,6345	806,4103	**701,9061**	1.451,8455
	MAE	15,2396	15,9038	5,5209	21,1816	20,4852	**18,7513**	20,8025
	MAPE	3,5885	3,7922	3,6534	5,1608	4,8948	**4,4825**	4,5910
	MDA		0,5315	0,4822	0,5299	0,4873	0,5035	**0,6156**
BTC-1	**MSE**	1,44E+6	1,44E+6	1,46E+6	20,32E+6	18,54E+6	**11,43E+6**	25,43E+6
	MAE	675,7244	673,2031	677,2318	2.513,5285	2.329,4232	**1.746,4387**	2.497,9447
	MAPE	2,4060	2,3975	2,4012	6,5692	6,2380	**4,8619**	6,5484
	MDA		0,5205	0,5452	0,4878	0,4795	0,5048	**0,5760**
BTC-2	**MSE**	2,85E+6	2,96E+6	2,86E+6	6,59E+6	3,47E+6	3,37E+6	**2,14E+6**
	MAE	1.230,7676	1.269,3290	677,2318	1.944,1756	1.427,6399	1.389,5153	**1.131,6847**
	MAPE	2,7793	2,8709	2,7757	4,4507	3,2356	3,1687	**2,5363**
	MDA		0,5068	0,5068	0,5233	0,5428	0,5418	**0,6762**
DOGE-1	**MSE**	196,93E-6	210,27E-6	196,92E-6	428,85E-6	369,27E-6	250,21E-6	**209,75E-6**
	MAE	0,0031	0,0035	0,0031	0,0055	0,0046	**0,0035**	0,0045
	MAPE	4,7647	5,4577	4,7763	7,4487	6,6467	**5,7125**	8,2044
	MDA		0,4712	0,5178	0,5433	0,5473	0,5319	**0,5921**
DOGE-2	**MSE**	534,80E-6	849,29E-6	555,01E-6	843,26E-6	**795,20E-6**	958,62E-6	1,51E-3
	MAE	0,0119	0,0153	0,0031	0,0167	0,0162	**0,0160**	0,0163
	MAPE	4,4782	5,7029	4,5545	6,5134	6,5377	6,2716	**5,9473**
	MDA		0,5589	0,5178	0,5307	0,4965	0,5261	**0,6136**
ETH-1	**MSE**	2,79E+3	2,67E+3	2,85E+3	25,06E+3	**11,06E+3**	24,73E+3	11,67E+3
	MAE	30,7254	30,1741	30,8124	74,3797	56,8556	74,9170	**46,8973**
	MAPE	3,3744	3,3521	3,3793	5,8879	5,5853	5,9958	**3,6775**
	MDA		0,5178	0,5178	0,4766	0,4949	0,4888	**0,7102**
ETH-2	**MSE**	23,88E+3	27,13E+3	23,93E+3	**60,22E+3**	63,40E+3	77,54E+3	80,45E+3
	MAE	113,2543	122,9560	30,8124	**189,5752**	190,1298	198,6585	199,3909
	MAPE	3,6729	3,9847	3,6817	**5,8302**	6,0083	5,9774	5,8939
	MDA		0,5397	0,5014	0,5099	0,5090	0,5076	**0,5810**
LINK-1	**MSE**	1,4721	2,3342	1,4683	4,5002	3,2970	**2,0529**	10,4512
	MAE	0,7860	0,9704	0,7806	1,3231	1,1268	**0,9743**	1,8185
	MAPE	4,7462	5,6110	4,7180	7,4907	6,2109	**5,9839**	8,1839
	MDA		0,5068	0,5452	0,5325	0,5402	0,4848	**0,5994**
LINK-2	**MSE**	3,4389	3,4271	3,3783	5,6910	3,7305	4,6003	**2,2038**
	MAE	1,1898	1,1915	0,7806	1,5665	1,2898	1,5065	**0,9723**
	MAPE	4,9281	4,9672	4,9223	6,7176	5,6156	6,6142	**4,0397**
	MDA		0,5315	0,5205	0,5371	0,5380	0,5251	**0,7086**
	MSE	50,3886	50,0977	51,3899	74,8434	78,7669	70,2833	**48,5848**
	MAE	4,1442	4,1453	4,1605	5,0106	5,1424	4,8482	**3,7305**

Tab. 8: Evaluation of the models, the samples are described with the cryptocurrency abbreviation followed by the number of the sample.

Sample	Metric	RW	ARIMA	ETS	MLP	LSTM	BiLSTM	Trends BiLSTM
LTC-1	**MAPE**	3,5659	3,5560	3,5613	4,1317	4,2260	4,0291	**2,9077**
	MDA		0,5068	0,5178	0,5034	0,5163	0,5237	**0,7087**
LTC-2	**MSE**	135,5546	149,6312	134,0707	211,6212	150,5333	174,4532	**113,2340**
	MAE	6,7322	6,9581	4,1605	8,5474	7,1382	8,1270	**6,4552**
	MAPE	3,9287	4,0587	3,9099	4,9939	4,2220	4,8854	**3,6936**
	MDA		0,4904	0,5288	0,5230	0,5251	0,5139	**0,6630**
TRX-1	**MSE**	14,93E-6	24,76E-6	15,13E-6	30,45E-6	26,66E-6	**24,04E-6**	32,38E-6
	MAE	0,0018	0,0024	0,0018	0,0026	0,0024	0,0023	**0,0022**
	MAPE	3,8232	5,1849	3,7913	5,3944	4,9938	4,8746	**3,7819**
	MDA		0,5507	0,5315	0,4931	0,5190	0,5015	**0,6744**
TRX-2	**MSE**	22,62E-6	25,24E-6	22,19E-6	35,89E-6	34,46E-6	34,77E-6	**21,21E-6**
	MAE	0,0029	0,0032	0,0018	0,0041	0,0040	0,0041	**0,0033**
	MAPE	3,4762	3,8575	3,4605	5,1169	4,8978	5,0902	**3,8563**
	MDA		0,4849	0,5123	0,5380	0,5433	0,5507	**0,6104**
XLM-1	**MSE**	340,66E-6	371,14E-6	342,77E-6	575,78E-6	574,85E-6	526,99E-6	**192,03E-6**
	MAE	0,0095	0,0101	0,0095	0,0122	0,0122	0,0121	**0,0078**
	MAPE	4,1244	4,4413	4,1092	5,1209	5,1072	5,5031	**3,8518**
	MDA		0,5096	0,5205	0,5356	0,5397	0,5235	**0,6728**
XLM-2	**MSE**	480,01E-6	507,49E-6	480,00E-6	609,29E-6	593,52E-6	540,41E-6	**274,89E-6**
	MAE	0,0127	0,0127	0,0095	0,0156	0,0149	0,0144	**0,0104**
	MAPE	3,9021	3,9057	3,9025	4,9421	4,6599	4,5034	**3,3177**
	MDA		0,5589	0,5068	0,5137	0,5238	0,5182	**0,6971**
XRP-1	**MSE**	2,52E-3	3,14E-3	2,52E-3	5,66E-3	4,42E-3	4,04E-3	**2,13E-3**
	MAE	0,0218	0,0262	0,0218	0,0346	0,0299	0,0290	**0,0239**
	MAPE	4,2300	5,3953	4,2300	6,7704	5,6912	**5,5381**	5,5458
	MDA		0,4630	0,5534	0,5094	0,5014	0,5100	**0,5985**
XRP-2	**MSE**	3,56E-3	4,09E-3	3,53E-3	4,63E-3	5,99E-3	5,15E-3	**3,01E-3**
	MAE	0,0373	0,0425	0,0218	0,0472	0,0562	0,0529	**0,0385**
	MAPE	4,0090	4,5221	3,9946	5,2432	6,3003	5,7999	**4,1782**
	MDA		0,4712	0,5178	0,5152	0,4941	0,5014	**0,6152**

Besides the good results from the statistical methods, especially the ETS, the BiLSTM model enhanced with Google Trends data surpassed all the other methods by a large margin in terms of directional accuracy and presented better results for MSE, MAE and MAPE for part of the samples. For samples of BNB, ETH, LINK and LTC, it achieved a directional accuracy of over 70%, while no accuracy was lower than 57% for all the currencies. The Trends BiLSTM also lead to the best overall results in terms of MSE, presenting the lowest value for 9 samples, while ETS resulted on the best MAE results, beating other methods for 13 samples. In terms of MAPE, Trends BiLSTM and ETS also presented the best results, with the lowest values for 7 and 6 samples respectively.

Between the ANNs using only previous exchange rates, BiLSTM lead to the best results, which explains its selection to the application of exogenous features. It was followed by LSTM and then MLP, that presented the poorest results between all the models for most currencies. MLP actually presented the best results for ETH and XRP second samples in all the metrics, but was unable to repeat these results for the other samples, beating other models only in terms of MDA for BNB and DOGE second samples. In the MLP vs LSTM comparison, LSTM presented lower forecasting errors for 15 samples in terms of MAPE and MAE and 16 in terms of MSE. On the comparison between the 3 models, BiLSTM presented the lowest MAE for 13 samples.

Comparing only the statistical methods, ETS surpassed the other methods for 14 of the 20 samples, in terms of MDA and MAE, being the best model in the recent period (Sample 2) for all the cryptocurrencies with consideration of its MAE values. RW presented better results than ARIMA in its calculated metrics, with the lowest MSE for 9 samples, which illustrates a decent applicability of using the actual value as forecast.

5. Conclusion

Neural networks constitutes a powerful method for a wide variety of problems, with several studies applying them to the financial market. This article composes a comprehensive comparison between traditional statistical methods and different neural networks approaches. It differentiate itself by applying this methodology for a wide set of cryptocurrencies, which includes Bitcoin, Ethereum, Binance Coin, Cardano, Ripple, Dogecoin, Litecoin, Chainlink, Tron and Stellar.

Traditional statistical methods lead to good and consistent results, showing why this methods are considered state of the art on time series prediction. ETS approach resulted on smaller forecasting errors and higher directional accuracy, beating ARIMA and the Random Walk. MLP networks presented poorer results, as expected, while applying LSTM lead to consistent improves on forecasts. The implemented BiLSTM surpasses the traditional LSTM, presenting the best results between the ANNs and proving it is a good fit to financial time series. An enhanced BiLSTM, by adding Google Trends data to the inputs, lead to the best overall results, beating the state of the art for several samples, with impressive results especially on forecasting future directions to next step closing prices, which may be used by investors in their decision process.

Cryptocurrencies exchange rates present itself as very volatile, with very different behaviours depending on time period. The implemented sample division addressed these changes but in sample variations also represented difficulties. Combined with the clear non-stationarity of the data, this lead the ANN models with a rather complex situation in fitting to available data. In some cases validation losses failed to decrease even with a very careful hyperparameter selection, in that case refitting the model was necessary. Additionally, Computational resources limited number of experiments and may provide additional limitations for a larger set of input variables. These limitations imply on less possibilities on hyperparameter tuning and less complexity on the models.

Further researches may consider applying different ANN architectures. As reviewed from literature, mixed and convolutional models have already presented positive results (Alonso-Monsalve et al., 2020). CNN-LSTM and ConvLSTM may be tested. Another important step is the application of a more diverse set of variables as input to the networks and comparisons with Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) approach. This step depends on availability of data for the set of cryptocurrencies. A selection process for the variables may also apply, leading to networks with even better results.

References

Alonso-Monsalve, S., Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Systems with Applications*, *149*, 113250.

Cavalli, S., & Amoretti, M. (2021). Cnn-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing*, *101*, 107065.

Chalvatzis, C., & Hristu-Varsakelis, D. (2020). High-performance stock index trading via neural networks and trees. *Applied Soft Computing*, *96*, 106567.

Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, *37*(1), 28-43.

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.

CoinMarketCap. (2022). *Today's cryptocurrency prices by market cap.* Retrieved 2022-06-20, from `https://coinmarketcap.com`

Frazier, P. I. (2018). *A tutorial on bayesian optimization.* arXiv.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, *18*(5), 602-610. (IJCNN 2005)

Guo, H., Zhang, D., Liu, S., Wang, L., & Ding, Y. (2021). Bitcoin price forecasting: A perspective of underlying blockchain transactions. *Decision Support Systems*, *151*, 113650.

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780.

Hu, H., Tang, L., Zhang, S., & Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with google trends. *Neurocomputing*, *285*, 188-195.

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

Ibrahim, A., Kashef, R., & Corrigan, L. (2021). Predicting market movement direction for bitcoin: A comparison of time series modeling methods. *Computers & Electrical Engineering*, *89*, 106905.

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science*, *7*, 45-66.

Koo, E., & Kim, G. (2021). Prediction of bitcoin price based on manipulating distribution strategy. *Applied Soft Computing*, *110*, 107738.

Liu, M., Li, G., Li, J., Zhu, X., & Yao, Y. (2021). Forecasting the price of bitcoin using deep learning. *Finance Research Letters*, *40*, 101755.

Nakamoto, S. (2009). *Bitcoin: A peer-to-peer electronic cash system.* Retrieved from `http://www.bitcoin.org/bitcoin.pdf`

Nakano, M., Takahashi, A., & Takahashi, S. (2018). Bitcoin technical trading with artificial neural network. *Physica A: Statistical Mechanics and its Applications*, *510*, 587-609.

Peng, Y., Albuquerque, P. H. M., Kimura, H., & Saavedra, C. A. P. B. (2021). Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. *Machine Learning with Applications*, *5*, 100060.

Poongodi, M., Nguyen, T. N., Hamdi, M., & Cengiz, K. (2021). Global cryptocurrency trend prediction using social media. *Information Processing & Management*, *58*(6), 102708.

Preis, T., Moat, H. S., & Stanley, H. E. (2013, April). Quantifying trading behavior in financial markets using google trends. *Scientific Reports*, *3*(1). Retrieved from `10.1038/srep01684` doi: 10.1038/srep01684

Silva de Souza, M. J., Almudhaf, F. W., Henrique, B. M., Silveira Negredo, A. B., Franco Ramos, D. G., Sobreiro, V. A., & Kimura, H. (2019). Can artificial intelligence enhance the bitcoin bonanza. *The Journal of Finance and Data Science*, *5*(2), 83-98.

Yang, M., & Wang, J. (2022). Adaptability of financial time series prediction based on bilstm. *Procedia Computer Science*, *199*, 18-25.