Applied Soft Computing Journal 101 (2021) 107065

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

CNN-based multivariate data analysis for bitcoin trend prediction

Stefano Cavalli, Michele Amoretti [*]

Department of Engineering and Architecture, University of Parma, Italy

ARTICLE INFO

Article history:
Received 3 August 2020
Received in revised form 23 November 2020
Accepted 21 December 2020
Available online 28 December 2020

Keywords:
Blockchain
Sentiment analysis
Financial indicators
CNN

ABSTRACT

Bitcoin is the most widely known blockchain, a distributed ledger that records an increasing number of transactions based on the bitcoin cryptocurrency. New bitcoins are created at a predictable and decreasing rate, which means that the demand must follow this level of inflation to keep the price stable. Actually, the price is highly volatile, because it is affected by many factors including the supply of bitcoin, its market demand, the cost of the mining process, as well as economic and political world-class news.

In this work, we illustrate a novel approach for bitcoin trend prediction, based on the One-Dimensional Convolutional Neural Network (1D CNN). First, we propose a methodology for building useful datasets that take into account social media data, the full blockchain transaction history, and a number of financial indicators. Moreover, we present a cloud-based system characterized by a highly efficient distributed architecture, which allowed us to collect a huge amount of data in order to build thousands of different datasets, using the aforementioned methodology. To the best of our knowledge, this is the first work that uses 1D CNN for bitcoin trend prediction. Remarkably, an efficient and low-cost implementation is feasible due to the simple and compact configuration of 1D CNN models that perform one-dimensional convolutions (i.e., scalar multiplications and additions). We show that the 1D CNN model we implemented, trained, validated and tested using the aforementioned datasets, allow one to predict the bitcoin trend with higher accuracy compared to LSTM models. Last but not least, we introduce and simulate a trading strategy based on the proposed 1D CNN model, which increases the profit when the bitcoin trend is bullish and reduces the loss when the trend is bearish.

1. Introduction

A blockchain is a system that acts as a trusted and reliable third party, not centralized, always online, to preserve a shared state, mediate exchanges and provide secure computations [1,2]. Technically, it is a distributed ledger that stores transaction data, grouped into blocks that constitute a growing and unalterable linked list. The ledger is managed by a large group of networked servers (known as *full nodes*), each one storing a copy of the whole blockchain. Since the blockchain grows, the servers need to reach consensus on each new block to be included. A wallet is a software for making transactions and checking their validity (using asymmetric encryption).

Speaking about cryptocurrency associated to a blockchain, we need to discern between *coins*, the blockchain's own units of virtual currency, for transaction purposes, and *tokens*, which are secondary units that reside in a blockchain, with various purposes. Transactions describe payments for goods and services, made by means of coins. In a transaction, the parties are identified by their public keys. Every payment must be digitally signed.

Bitcoin is the most widely known blockchain. It was introduced in 2009 by an anonymous person (or group of people) using the pseudonym Satoshi Nakamoto [3]. The Bitcoin Core software [4] includes a transaction verification engine and connects to the network as a full node. Bitcoin coins (denoted as *bitcoins* not capitalized) are created at a predictable and decreasing rate, which means that the demand must follow this level of inflation to keep the price stable. Bitcoin prices, which are highly volatile, are influenced by many factors, including the supply of bitcoin and the market demand for it, the cost of the mining process, the number of competing cryptocurrencies, the amount of transactions based on the Bitcoin platform. The Blockchain technology gained momentum when the bitcoin price reached its all-time high value of 19891$ in 2017. From that moment on, the bitcoin time series became an object of study for the research community.

Time series forecasting is a very popular regression problem in the machine learning community [5–10]. A related, but different problem is trend prediction of time series intended as a *classification* task, which is particularly challenging when it comes to the highly speculative cryptocurrency market.

With reference to the bitcoin time series, both problems have been studied in recent years. Two major competing research

[*] Corresponding author.
E-mail address: michele.amoretti@unipr.it (M. Amoretti).

https://doi.org/10.1016/j.asoc.2020.107065
1568-4946/© 2020 Elsevier B.V. All rights reserved.

lines have emerged, the first one leveraging sentiment analysis on social media, the second one focusing on machine learning applied to financial datasets. Kaminski [11] studied correlations and causalities between Bitcoin market indicators and Twitter posts. Matta et al. [12] analyzed the bitcoin trend using Google Trends data and 1924891 tweets. Madan et al. [13] proposed a bitcoin forecasting approach based on machine learning algorithms, namely Support Vector Machine (SVM) and Random Forest. Greaves and Au [14] adopted a feed-forward neural network with two hidden layers. McNally et al. [15] proposed a Long Short Term Memory (LSTM) model. Many other remarkable papers could be cited (to this purpose, we refer the reader to Section 2). The common factor of these research works is the demonstration that the public transaction history is not enough for making accurate predictions of the bitcoin trend. Indeed, it is desirable to have a large dataset of items with heterogeneous features when it comes to train and test a machine learning model. In this work, we show that an accuracy improvement in predicting the bitcoin trend can be achieved by means of machine learning on datasets that are cleverly constructed from bitcoin historical values, financial indicators, bitcoin-related social media sentiment and Bitcoin blockchain information.

1.1. Our contributions

In this work, we illustrate a novel approach for bitcoin trend prediction based on One-Dimensional Convolutional Neural Network (1D CNN) models [16]. The proposed models predict whether the bitcoin value after z days will be lower or higher than the latest value of the time series.

First, we propose a methodology for building datasets whose items are characterized by different types of features: bitcoin historical values and financial indicators, Twitter sentiment analysis and Bitcoin blockchain information.

Moreover, we present a cloud-based system with a highly efficient distributed architecture, which allowed us to collect a huge amount of data and to create thousands of different datasets.

We show that the 1D CNN model we implemented, trained, validated and tested using the aforementioned datasets, allows one to predict the bitcoin trend with higher accuracy compared to LSTM models.

Despite CNN models are usually adopted for image recognition, 1D CNN models have been only recently proposed for prediction tasks that involve time series, such as real-time health monitoring [17], motor-fault detection [18] and multi-temporal crop classification [19], with encouraging performance results. To the best of our knowledge, our work is the first one that uses 1D CNN models for bitcoin trend prediction. Another important aspect of the proposed approach is that an efficient and low-cost implementation is feasible due to the simple and compact configuration of 1D CNNs that perform one-dimensional convolutions (i.e., scalar multiplications and additions).

The paper is organized as follows. In Section 2, a survey of the state of the art in bitcoin trend prediction is provided. In Section 3, the proposed approach and its implementation are illustrated. In Section 4, the experimental evaluation results are presented. In Section 5, the achieved results are discussed by analyzing the reasons behind them. Finally, in Section 6, the paper is concluded with a final summary of achieved results and an outline of future work.

2. Related work

Regarding bitcoin forecasting and trend prediction, two major competing research lines have recently emerged, the first one leveraging sentiment analysis on social media, the second one focusing on machine learning applied to financial datasets. Performance is measured in terms of *binary accuracy*, which is the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined. The formula for quantifying binary accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

where TP denotes the number of true positives, TN denotes the number of true negatives, FP denotes the number of false positives, and FN denotes the number of false negatives.

2.1. Social media and Bitcoin

Kaminski [11] studied correlations and causalities between Bitcoin market indicators and Twitter posts. The considered dataset spans 104 days (from November 2013 to March 2014) and contains 161 200 tweets, as well as different features extracted from different exchanges, such as BitStamp, Bitfinex, BTC-e and BTC China. The results of the data analysis led the author to the interpretation that emotional sentiments rather mirror the market than that they make it predictable. However, the considered timeframe is too short and characterized by an unusual bitcoin trend (exponential growth) to draw any general conclusion.

Matta et al. [12] analyzed the bitcoin trend using Google Trends data and 1 924 891 tweets, in a timeframe of 60 days (from January to March 2015). They observed a correlation between the bitcoin price and the tweets that express a positive sentiment. Remarkably, the tweets appear to anticipate by 3–4 days the bitcoin trend. Correlation results between the bitcoin price and Google Trends data are less convincing. Google Trends data are not easy to handle, as they are always normalized with respect to the considered timeframe, such that the period with the highest relative search intensity corresponds to an arbitrary reference value set to 100. The data presented in the paper are not normalized, suggesting some kind of pre-processing that however is not explained.

Stenqvist and Lonno [20] considered a 31-days timeframe (from May to June 2017), in which they collected 2 271 815 tweets. Their sentiment analysis was carried out by means of a powerful tool, namely VADER [21]. A careful cleaning process allowed the authors to exclude more than 50% of the tweets, i.e., those produced by bots and those carrying duplicated content. Despite the sound approach, the resulting accuracy of the predicted bitcoin value shows too much variability, depending on the chosen timeframe.

2.2. Machine learning for bitcoin forecasting and trend prediction

Madan et al. [13] proposed a bitcoin forecasting approach based on machine learning algorithms. In particular, they predicted the sign of the future change in price using a binomial generalized linear model (GLM), leveraging both support vector machine (SVM) and random forest. The considered dataset has 26 features relating to the bitcoin price and payment network over the course of five years (from 2009 to 2014). The proposed solution achieves 50%–55% accuracy in predicting the sign of future price change using 10 min time intervals.

The same result was obtained by Greaves and Au [14], with a reduced timeframe for the dataset (1 year) and 12 features. In this case, the authors solved the prediction problem by means of a feed-forward neural network with two hidden layers.

Using a GPU-enhanced deep learning approach, with a Long Short Term Memory (LSTM) network, McNally et al. [15] achieved a 52% accuracy. The dataset takes into account the bitcoin value (collected from Coindesk.com), the hash power, the mining difficulty and other information extracted from the blockchain. For

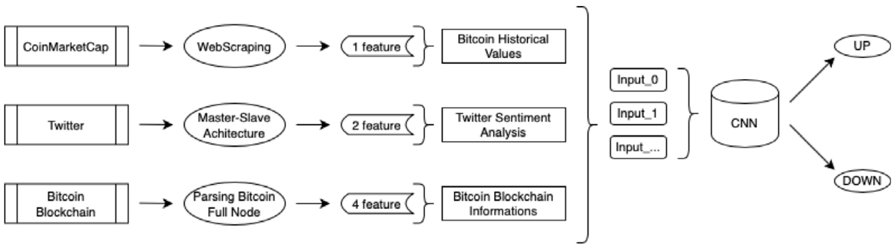


Fig. 1. Overview of the proposed approach.

the first time, a financial index was used, namely the simple moving average (SMA). The considered timeframe spans 3 years (from August 2013 to July 2016).

Mittal et al. [22] studied the correlation between bitcoin price, Twitter and Google search patterns. Using different machine learning techniques, they concluded that there is a relevant degree of correlation of Google Trends and Tweet volume data with the bitcoin price, and no significant relation with the sentiments of tweets. In particular, the authors achieved a 62.4% accuracy in predicting bitcoin price fluctuations based on Google Trends and Tweet Volume using a Recurrent Neural Network (RNN) model.

Altan et al. [23] proposed a hybrid forecasting model based on LSTM and empirical wavelet transform (EWT) decomposition along with cuckoo search (CS). The proposed model is tested for one-step forecasting of cryptocurrency prices such as Bitcoin, Ripple, Digital Cash and Litecoin. The experimental results show that the proposed model can successfully capture nonlinear features of the bitcoin time series.

In a recent work, Linardatos and Kotsiantis [24] analyzed 7M tweets, Google Trends data, the bitcoin price and other features, over a 2-years timeframe (from January 2017 to December 2018). They used VADER [21] for the sentiment analysis of the tweets, and an LSTM network for the prediction task. The resulting accuracy was 52%.

Chen et al. [25] included the gold trend into the dataset. In this way, still using an LSTM network, they achieved a 67.2% accuracy.

3. Proposed approach

Schematically illustrated in Fig. 1, the proposed approach for bitcoin trend prediction has been implemented on a cloud-based system characterized by a highly efficient distributed architecture. Three independent subsystems extract specific data from CoinMarketCap, Twitter and the Bitcoin blockchain, respectively. With the extracted data, another subsystem builds several datasets using a parameterized approach. Each dataset is then used to train a Convolutional Neural Network (CNN) model. Once trained, the model can be used to predict the bitcoin trend. In the following, we describe every subsystem in detail. As a premise, we illustrate and discuss the volatility of the bitcoin asset.

3.1. Bitcoin volatility analysis

The value of the bitcoin asset has been historically volatile. Let us define the Volatility Index (VI) as the percentage variation of an asset over time. In the following, we compare the average daily and monthly VI of different assets, using the US Dollar (USD) as reference asset. The reason why we focus on GOLD/USD, EUR/USD and AAPL/USD is because they are the top traded assets within the three world trading markets: Commodities, Forex and Stocks (Table 1).

Since these markets are closed during the weekends and festivities, while cryptocurrencies is a 24/7 trading market, in order to provide a consistent analysis we adopted two different approaches for computing the daily and monthly VI: the first one includes all the bitcoin values, while the second one does not consider the bitcoin values related to weekends and festivities. The considered data have been retrieved from Investing.com [26], the world's leader financial platform, according to Similarweb [27], that provides real-time data on multiple assets for over 44 millions people per month.

Table 1
Market assets.

Market	Asset
Commodities	GOLD
Forex	EUR
Stocks	AAPL
Cryptocurrencies	BTC

Table 2
Absolute daily VI and absolute monthly VI averaged over a seven years long period (from March 2013 to January 2020).

Asset	Absolute average daily VI	Absolute average monthly VI
GOLD/USD	0.3679%	0.0817%
EUR/USD	0.6294%	0.1585%
AAPL/USD	1.0891%	0.3297%
BTC/USD	3.1988%	1.005%

The bitcoin daily VI in Fig. 2 and the bitcoin monthly VI in Fig. 3 show a more fluctuating trend compared to those of the VI values of the other assets. The average daily and monthly VI values of all the four assets are reported in Table 2. It is evident how volatile the bitcoin value is, and consequently how useful an accurate predictive model may be.

3.2. Bitcoin historical values and financial indicators

We used the Bitcoin market (USD) daily historical values that we obtained from CoinMarketCap [28] by means of *web scraping*. Using the Python package denoted as Beautiful Soup [29], we analyzed the HTML page for retrieving all the required data. In particular, we focused on the timeframe that goes from the 28th of April 2013 to 15th of February 2020 (2485 days). Leveraging the algorithms proposed by TradingView [30], we reconstructed ten of the most important financial indicators and oscillators in financial trading, still using Python as a programming language.

Financial indicators are used considering ranges that are congruent with each other: if we take a look at one chart with a daily timeframe, it makes sense to combine indicators with the same daily timeframe together, while it would be a mistake to consider minutes and hours jointly. Furthermore, when using financial oscillators and indicators, most of the time the dimensions are also congruous: comparing the 10-period Relative Strength Index (RSI) with the 12-period Simple Moving Average (SMA) is considered as an error. In fact, all indicators are constructed dynamically, choosing a *start_value* and a *final_value*, and each of them represents a table in a database. These tables always

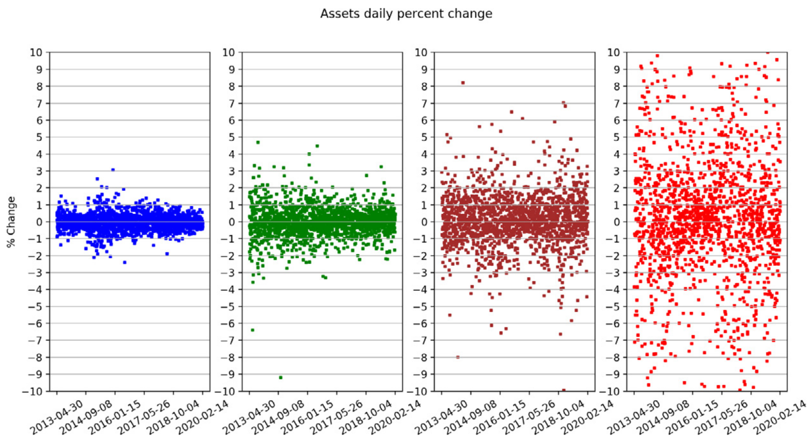


Fig. 2. Bitcoin daily VI.

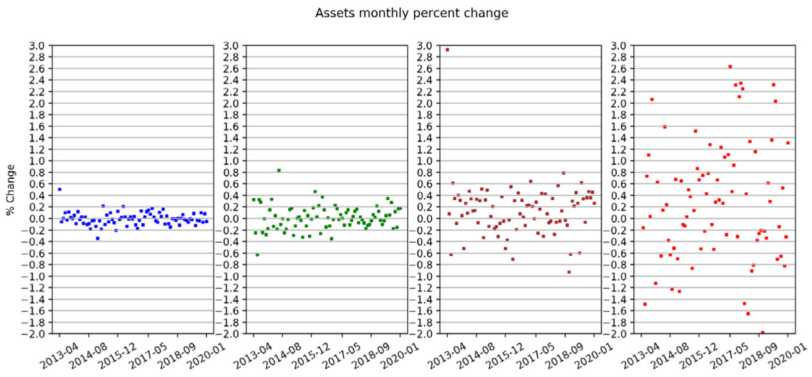
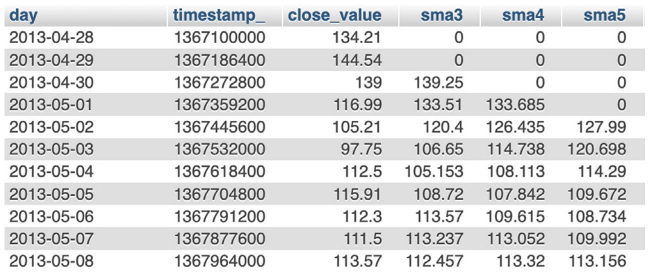


Fig. 3. Bitcoin monthly VI.



day	timestamp_	close_value	sma3	sma4	sma5
2013-04-28	1367100000	134.21	0	0	0
2013-04-29	1367186400	144.54	0	0	0
2013-04-30	1367272800	139	139.25	0	0
2013-05-01	1367359200	116.99	133.51	133.685	0
2013-05-02	1367445600	105.21	120.4	126.435	127.99
2013-05-03	1367532000	97.75	106.65	114.738	120.698
2013-05-04	1367618400	112.5	105.153	108.113	114.29
2013-05-05	1367704800	115.91	108.72	107.842	109.672
2013-05-06	1367791200	112.3	113.57	109.615	108.734
2013-05-07	1367877600	111.5	113.237	113.052	109.992
2013-05-08	1367964000	113.57	112.457	113.32	113.156

Fig. 4. The bitcoin SMA table where *start_value* = 3 and *final_value* = 5.

show days, timestamps, bitcoin closing values and each indicator together with its size (Fig. 4).

Since the 16th of February 2020, the aforementioned process runs in real time, waking up every day, early in the morning, to collect bitcoin data of the previous day and to keep the feature tables up to date.

3.3. *Twitter sentiment analysis*

Another important feature is the sentiment expressed by the people on social media, especially on Twitter. Giaglis et al. [31] realized that Twitter sentiment ratio has a positive short-run impact on bitcoin prices. Measuring collective mood based with sentiment analysis techniques can help to predict short-run movements in the value of bitcoins. To retrieve all the tweets containing the word bitcoin and written in English, related to the timeframe of interest, it was necessary to set up an highly parallel scraping system. Thanks to the Python module called TwitterScraper [32] and the use of a multiclient–server architecture, we were able to collect all the tweets we wanted.

Based on empirical considerations, we deployed a master–slave architecture where 1 server plays the master role and 22 clients play the slave role (Fig. 5). This process involves a mechanism that makes the master and the slaves interact to download the all the tweets efficiently and robustly. Once a slave has collected the tweets of the assigned time slot, it returns them to the master using the SSH protocol (Fig. 6). Every slave is a TwitterScraper instance, which may run for 72 h before its IP is banned by Twitter. Thus, in order to retrieve a large amount of data, multiple slaves are necessary.

Once all the tweets were collected, a preprocessing phase was necessary to remove all those indicating a behavior deriving from the use of bots, fake accounts and scams. Actually, we removed all the tweets containing one or more of the following words: 'join', 'joined', 'free', 'vote', 'prize', 'faucet', 'win', 'won', 'signal', 'bot', 'earn', 'last price', 'last trade', 'bitcoinews', 'bitcointalk', 'coindesk',

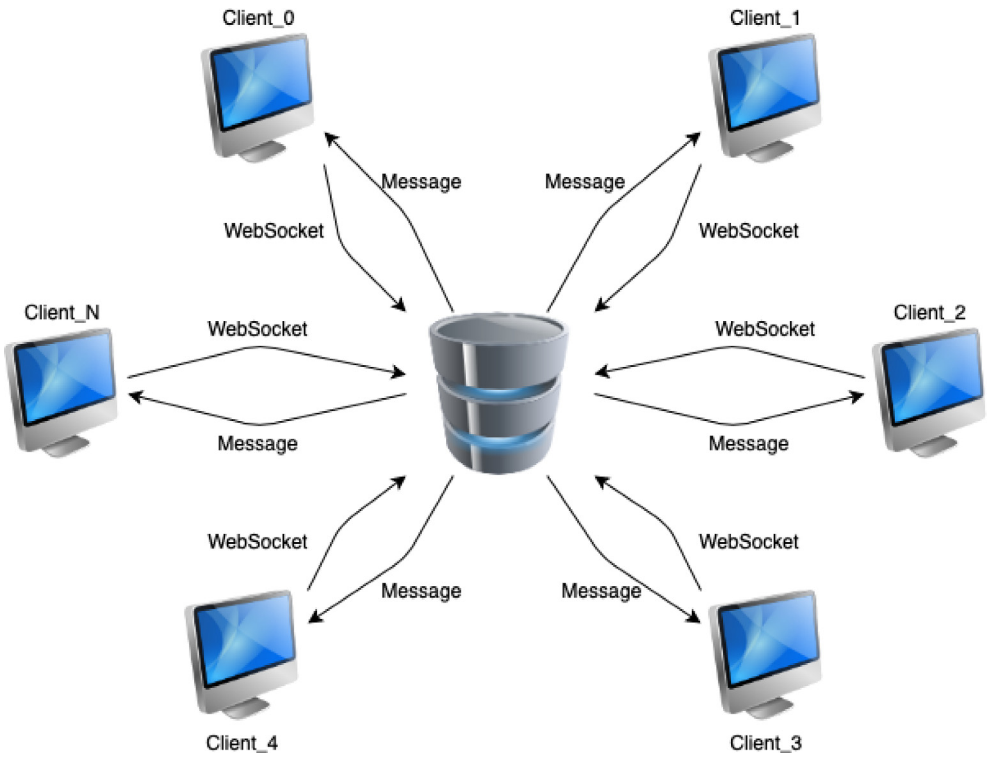


Fig. 5. Master–slave architecture: each slave establishes a connection with the master using a WebSocket. The master replies by sending a message containing the instructions for the slave to download tweets.

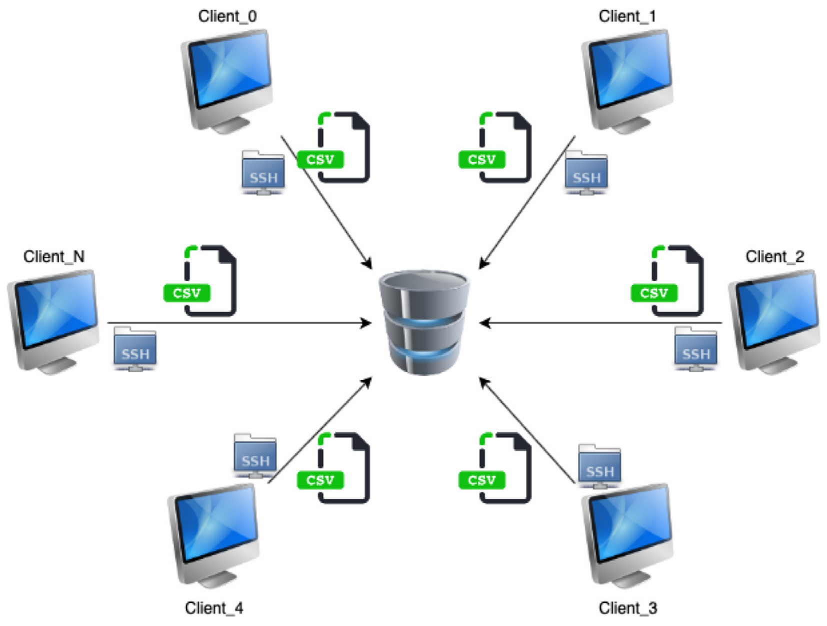


Fig. 6. The slaves open an SSH connection with the master, to send the collected tweets in CSV format.

'cointelegraph'. The remaining tweets were uploaded inside a table and subsequently analyzed one by one with VADER [21], a widely used sentiment analysis tool. VADER takes advantage of a sentiment lexicon, i.e., a list of lexical features that have been labeled according to their semantic orientation as either positive or negative. Using such an approach, VADER has been found to be quite successful when dealing with social media texts and sentiment analysis [33] in many different contexts, such as election result prediction [34], financial sentiment analysis [35] and twitter sentiment analysis [36]. In particular, its main feature is the ability to tell how positive or negative a sentiment is, instead of just classifying a sentiment as positive or negative. VADER does not require any training data, but it is already constructed from a human-curated and gold standard sentiment lexicon. Thanks to its reduced execution time, VADER can be also used online with streaming data. The sentiment classification process is briefly described as follows: once a sentence is being processed by VADER, it receives three different scores: Positive, Neutral and Negative. They sum to 1 and represent the fractions of the sentence that fall in these categories. In the example

Table 3
VADER metrics and scores for the sentence "Bitcoin is cool, but I am not sure its trend will be bullish".

Sentiment metric	Score
Positive	0.173
Neutral	0.71
Negative	0.117

Table 4
Sentiment analysis records.

Day	Pos_score	Neu_score	Neg_score	Tweets_Num
2013-01-01	0.0685346	0.900429	0.0310369	217
2013-01-02	0.0735163	0.884746	0.0417398	366
...	...	...	...	
2020-02-15	0.0819573	0.842587	0.075456	32 121

reported in Table 3, the sentence "Bitcoin is cool, but I am not sure its trend will be bullish" was rated as 17.3% Positive, 71% Neutral and 11.7% Negative. VADER takes into account different aspects when it comes to analyze a sentence. The most useful features are described below:

- The use of punctuation matters: an exclamation mark increases the magnitude of the intensity keeping the original semantic orientation ("Bitcoin is great!" will have a Positive score greater then "Bitcoin is great").
- Capitalization emphasize a sentiment-relevant word within a sentence ("Bitcoin is increasing, what a WONDERFUL day" will have a Positive score greater then "Bitcoin is increasing, what a wonderful day").
- Emojis, English slang words and emoticons are always identified.

The results, grouped by day and stored in Table 4, show VADER's performance while working on these tweets: Pos_score, Neu_Score and Neg_score respectively represent how Positive, Negative o Neutral the overall text analyzed is; Tweets_Num indicates the number of tweets.

We chose VADER after several useless attempts to classify the sentiment expressed by the tweets by means of different kinds of neural networks. We realized that the considered topic (Bitcoin and Cryptocurrencies) is too specific and there are no classified datasets (known as "gold datasets") that meet our expectations.

Currently, the deployed system is online in a reduced but still efficient version. The server plays both master and slave roles, waking up early in the morning to retrieve all the bitcoin-related tweets of the previous day.

3.4. Bitcoin blockchain information

The last set of features comes directly from the Bitcoin blockchain. In order to get real-time information such as the number of transactions inside a block or the mining difficulty, a Bitcoin full node was necessary. Once the blockchain was synchronized, we started a local Electrum Server, namely Electrs [37], which maintains the blockchain indexed and allows one to query it whenever needed with negligible execution time (Fig. 7). Every information item was taken directly from the blockchain, converting all the blk.dat binary files into different blk.hash textual files (Fig. 8), also saving more than the 99.98% of space by reducing every blk.dat size from 128 MB to 30 kB. Using this methodology, block-related data were collected in a table (Table 5) characterized by the following columns: number of transactions (Tx_count), mining fees (Fee_count), block size (Block_size) and mining difficulty (Difficulty).

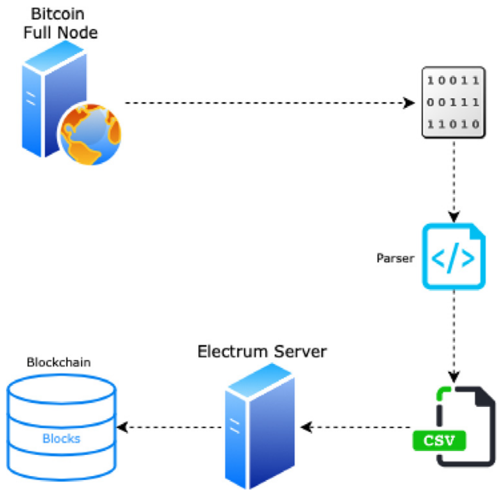
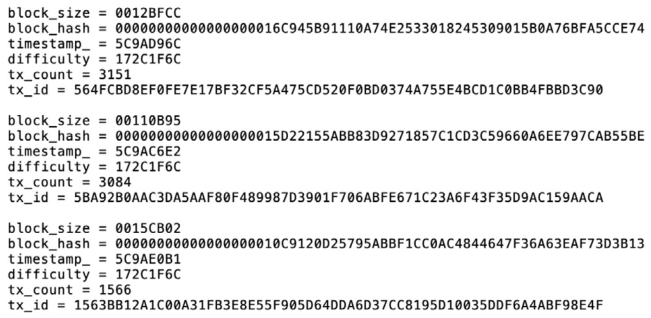


Fig. 7. All files coming from the Bitcoin blockchain are extracted and analyzed in order to produce a size reduction together with a textual interpretation.



```
block_size = 0012BFCC
block_hash = 00000000000000000016C945B91110A74E2533018245309015B0A76BFA5CCE74
timestamp_ = 5C9AD96C
difficulty = 172C1F6C
tx_count = 3151
tx_id = 564FCBD8EF0FE7E17BF32CF5A475CD520F0BD0374A755E4BCD1C0BB4FBBD3C90

block_size = 00110B95
block_hash = 00000000000000000015D22155ABB83D9271857C1CD3C59660A6EE797CAB55BE
timestamp_ = 5C9AC6E2
difficulty = 172C1F6C
tx_count = 3084
tx_id = 5BA92B0AAC3DA5AAF80F489987D3901F706ABFE671C23A6F43F35D9AC159AACA

block_size = 0015CB02
block_hash = 00000000000000000010C9120D25795ABBF1CC0AC4844647F36A63EAF73D3B13
timestamp_ = 5C9AE0B1
difficulty = 172C1F6C
tx_count = 1566
tx_id = 1563BB12A1C00A31FB3E8E55F905D64DDA6D37CC8195D10035DDF6A4ABF98E4F
```

Fig. 8. Example of the first three blocks of a blk.hash file, parsed from binary to textual visualization.

Table 5
Block-related data records.

Day	Block_size	Difficulty	Tx_count	Fee_count
2013-01-01	285	103 294	40 738	250 493 433
2013-01-02	215	103 294	41 928	329 485 949
...	...	...		
2019-02-15	2 377 364	15 466 098 932	2787	136 812 409

After having all the data stored into the Days table, to keep the process working in real time, the Bitcoin blockchain was inspected every fifteen minutes, because the block average confirmation time is about 10 min. Using this technique, the Days table was always up to date.

3.5. Dataset definition

Let us denote as $\mathcal{B}$ the vector of historical values of the bitcoin price:

$$\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}. \tag{2}$$

We also define a triple (n, k, z), where n is the input length, k is the number of elements that must be considered in order to make a prediction and z is the time shift along the $\mathcal{B}$ vector. The (n, k, z)-dataset is defined as:

$$\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{\left\lceil \frac{|\mathcal{B}|-n-k+1}{z} \right\rceil}\}. \tag{3}$$

Every element $\mathcal{D}_i$ has the following structure:

$$\mathcal{D}_i = \{\mathcal{F}_i^1, \ldots, \mathcal{F}_i^F, l_i\} \quad i \in \{1, \ldots, |\mathcal{D}|\} \tag{4}$$

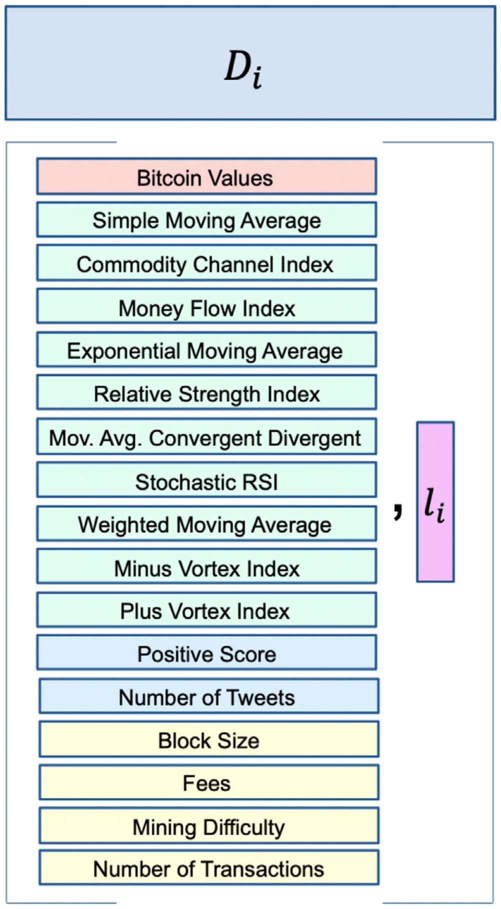


Fig. 9. Actual structure of a data item $\mathcal{D}_i$.

where

$$\mathcal{F}_i^j = \{f^j_{1+(i-1)z}, \ldots, f^j_{n+(i-1)z}\} \quad j \in \{1, \ldots, F\} \tag{5}$$

is the jth feature of the ith dataset element. Every f_x^j is a time-specific value of $\mathcal{F}_i^j$.

Actually, $\mathcal{D}_i$ is composed by $F = 17$ different features and one label l_i that is calculated as

$$l_i = \begin{cases} UP & \text{if} \quad \frac{\sum_{j=n+(i-1)z+1}^{n+(i-1)z+k} b_j}{k} \geq b_{n+(i-1)z} \\ DOWN & \text{otherwise} \end{cases} \tag{6}$$

In the following, we explain the features of a data item $\mathcal{D}_i$ that is illustrated in Fig. 9.

Each feature has a daily timeframe, starting from the 28th of April 2013 to the 15th of February 2020 (2485 days). The first and most valuable feature comes from a CoinMarketCap [28] web scraping process that allowed us to retrieve all historical bitcoin daily closes as *bitcoin values*. Ten financial indicators [30] are built on daily bitcoin open, high, low, close and volume values:

- *SMA*: Simple Moving Average
- *CCI*: Commodity Channel Index
- *MFI*: Money Flow Index
- *EMA*: Exponential Moving Average
- *RSI*: Relative Strength Index
- *MACD*: Moving Average Convergent Divergent
- *SRSI*: Stochastic Relative Strength Index
- *WMA*: Weighted Moving Average
- *MVI*: Minus Vortex Index
- *PVI*: Plus Vortex Index

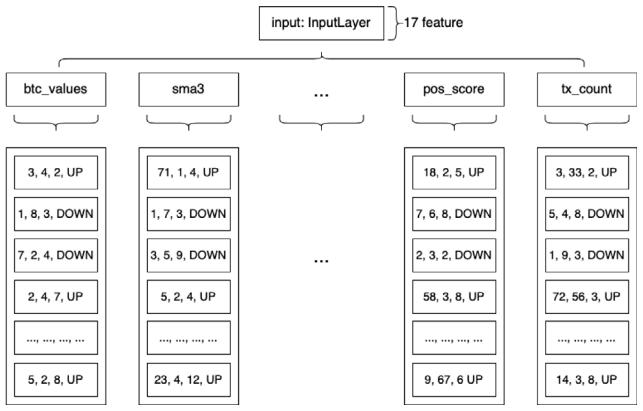


Fig. 10. Example dataset, characterized by $F = 17$ different features and an associated label defined by Eq. (6).

Two features come from a Twitter sentiment analysis process, based on the master–slave architecture described in Section 3:

- *Positive Score*: number of positive tweets analyzed with VADER [21]
- *Number of Tweets*: total number of useful tweets

Four features come from the Bitcoin blockchain analysis:

- *Block Size*: average size of blocks
- *Fees*: bitcoin fees
- *Mining Difficulty*: average mining difficulty
- *Number of Transactions*: number of transactions

3.6. Dataset balancing

Different datasets can be created by defining the values of the triple (n, k, z). As we have shown above, each element D_i has a specific label l_i. Each dataset must be balanced, i.e., must have the same number of "UP" and "DOWN" labels, in order to properly train a classifier.

The technique we used for achieving balanced datasets consists in removing some of the data items whose label appears more times than the other label. The selection is based on age, i.e., the older data items are removed, those that include bitcoin historical values typically dating back to 2013 or 2014. This choice is based on the fact that in 2013 the bitcoin marketcap was worth about 1.5 billion dollars. The CoinMarketCap website itself does not report any data on trading volumes until the 27th of December 2013. In 2013 and 2014 the trading data are inaccurate, not very useful and not even considered by other scientific studies [22,25,38–41]. In this way it is always possible to balance the dataset, even if it is necessary to reduce the number of inputs for this classification problem.

3.7. Convolutional neural network

Convolutional Neural Networks (CNN) [42–46] are generally used for image, video and audio recognition where multi-dimensional kernels are applied together with multi-dimensional inputs. In this work, we considered 1D CNN models, where data and kernels are one-dimensional vectors (Fig. 10).

The CNN model we implemented (using Keras [47]) has multiple layers and uses the following operators:

- *Convolutional Layer*: extracts features from data
- *Max-Pooling Layer*: performs matrix reduction considering just the highest values

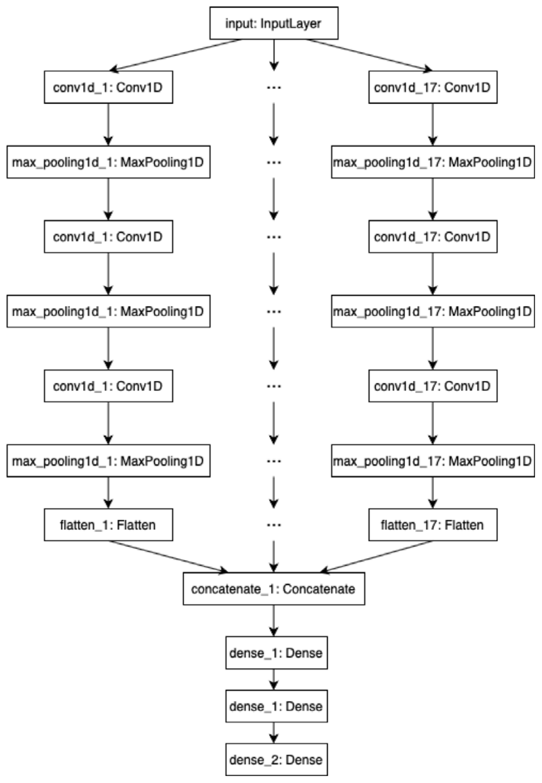

input: InputLayer
conv1d_1: Conv1D
conv1d_17: Conv1D
max_pooling1d_1: MaxPooling1D
max_pooling1d_17: MaxPooling1D
conv1d_1: Conv1D
conv1d_17: Conv1D
max_pooling1d_1: MaxPooling1D
max_pooling1d_17: MaxPooling1D
conv1d_1: Conv1D
conv1d_17: Conv1D
max_pooling1d_1: MaxPooling1D
max_pooling1d_17: MaxPooling1D
flatten_1: Flatten
flatten_17: Flatten
concatenate_1: Concatenate
dense_1: Dense
dense_1: Dense
dense_2: Dense

Fig. 11. Structure of the CNN model.

- *Flatten Operator*: unrolls the values beginning at the last dimension
- *Concatenate*: trains a model and makes a back-propagation that includes all the features
- *Dense Layer*: obtains a single output result

The number of layers depends on the dataset size and on the parameters that have to be analyzed. For the considered dataset, the CNN has one input layer, three consecutive (*Convolutional Layer*, *Max-Pooling Layer*) pairs, one *Flatten Operator* layer, one *Concatenate* operator and three *Dense Layers* (Fig. 11).

The keras.layers.Conv1D function (from Keras package) takes multiple parameters including *filters*, *kernel_size*, *activation*, *use_bias*, while the keras.layers.MaxPooling1D takes *stride* and *pool_size*. In detail:

- *filters* is the number of output filters within the convolution;
- *kernel_size* specifies the length of the 1D convolution window;
- *activation* is the activation function used in the model;
- *use_bias* when this parameter is true, a bias vector is created and added to the outputs;
- *stride*: indicates how much the pooling window moves for each pooling step;
- *pool_size* is the size of pooling window.

Kernel sizes depend on the input length n, in order to avoid convolution operations between vectors that are smaller than the kernels. More precisely, kernel sizes are defined by the following equations:

$$|kernel_1| = \left\lfloor \frac{n}{100} \right\rfloor \cdot 30 \tag{7}$$

$$|kernel_2| = \left\lfloor \frac{n}{100} \right\rfloor \cdot 20 \tag{8}$$

$$|kernel_3| = \left\lfloor \frac{n}{100} \right\rfloor \cdot 10 \tag{9}$$

The parameters of the three consecutive pairs of (*Convolutional Layer*, *Max-Pooling Layer*) are set according to the input size.

For the three instances of the *Convolutional Layer*, the following values of the parameters are used:

- *filters*: 32, 64, 128;
- *kernel_size*: $|kernel_1|$, $|kernel_2|$, $|kernel_3|$;
- *activation*: the Rectified Linear Activation Function (relu) is always used, which will output the input if it is positive, zero otherwise; it is specifically used for overcoming the vanishing gradient problem, allowing models to learn faster and performing better;

For all the three instances of *Max-Pooling Layer*, the following values of the parameters are used:

- *use_bias*: false;
- *stride*: 1;
- *pool_size*: 2.

Regarding the keras.layers.Dense function, it takes two parameters: *units*, which specifies the output space and *activation*, which indicates the activation function to use. The values for the three Dense layers are respectively: (100, relu), (50, relu), (1, sigmoid). Following the keras.model.compile function's parameters, the model has been configured with *losses*: binary_crossentropy, *metrics*: accuracy and *optimizer*: adam. In particular, the binary_crossentropy loss function is used in binary classification tasks and adam [48] is a widely used optimization algorithm.

It should be noted that it is not trivial to find the best performing network structure for a specific application, because most of the time it is unclear how the network structure relates to the network accuracy. The proposed model could be provided with self-tuning capabilities (which is left as future work).

Recent studies [49–51] demonstrated that artificial swarm intelligence algorithms are powerful tools for parameter optimization. The evolutionary approach [52] has found satisfactory results in tuning the parameters of a CNN, when it comes to classify images of the widely known MNIST dataset [53]. More in general, evolutionary algorithms have shown enormous optimization capacity for complex problems such as CNN hyperparameters tuning, where they work better than traditional mathematical programming methods. Whatever the adopted evolutionary approach, the objective is always to find a configuration h in the set of possible configurations of the CNN $\mathcal{H}$, that minimizes the application-specific prediction error.

Once trained, the CNN model is saved to a file, which is the input of the application that forecasts the bitcoin trend and calculates the accuracy of the obtained prediction. Prediction is based on the latest n days and tells the average bitcoin trend in the next k days.

4. Experimental results

In this section, we describe the experiments performed with the implemented system. We extracted data from CoinMarketCap, Twitter, and the Bitcoin blockchain from the 28th of April 2013 to the 15th of February 2020. With these data (which are publicly available online[1]), we built 18 121 different datasets using the parameterized approach illustrated in Section 3.4. We then used the datasets to train and test instances of the CNN model illustrated in Section 3.7 and of an LSTM model. We report some results on the accuracy of the bitcoin trend prediction and we compare the best one with those of the state of the art.

1 https://github.com/ML-unipr/bitcointrendpredictionML.

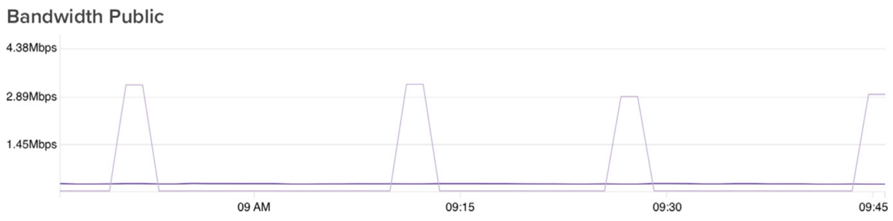


Fig. 12. Bandwidth consumption during the tweet download process: a few Mbps.

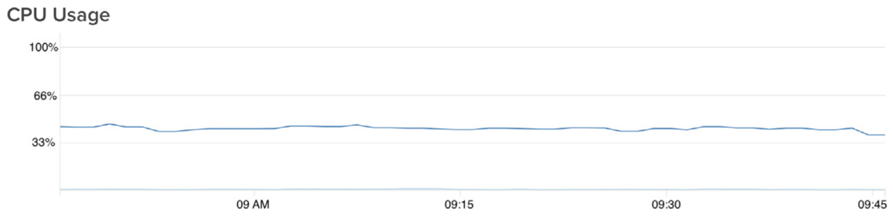


Fig. 13. CPU usage during the tweet download process: lower than 50%.

Table 6
Virtual private server configuration.

CPU	8 cores
RAM	30 GB
Disk storage	800 GB SSD
Port/Bandwidth	600 Mbps, unlimited traffic
Operating system	Linux Ubuntu 16.0
Upstream	200 Gbps

Table 7
Size of training, validation and test sets, depending on (n, k, z) values.

n	k	z	Training set size	Validation set size	Test set size
14	5	5	296	98	98
23	11	4	1950	122	122
38	**12**	**1**	**1461**	**487**	**487**
38	12	8	183	61	61
41	5	6	244	81	81
42	3	5	292	97	97
42	6	7	209	69	69

4.1. *Computational facility*

All the processes covered in Section 3 generate many outputs and large files. For this reason, we used a Virtual Private Server (VPS) as the main server. Its configuration is illustrated in Table 6.

4.2. *Bitcoin historical values and financial indicators*

Once the bitcoin historical values are retrieved by means of web scraping, any number of financial indicators can be created. The main purpose was to create a dynamic module that can be customized as much as possible, starting from the database creation up to the indicator calculation. Obtaining the BTC/USD values takes a few seconds, but the process that takes care of calculating each indicator requires 63 min, considering that each of them is based on a timeframe of 46 days (for example: simple moving average 5-days, simple moving average 6-days, simple moving average 50-days). Since CoinMarketCap may change bitcoin historical values based on adjustments, all indicators are completely recalculated from scratch every day.

4.3. *Twitter sentiment analysis*

Thanks to the multiclient–server architecture described in Section 3, we were able to download over 52 million tweets, in a running time of approximately 69 h, using 23 servers (1 master and 22 slaves). The plot in Fig. 12 shows that the bandwidth consumption per slave was quite limited. The CPU usage was also very low (Fig. 13).

During the pre-processing phase, we checked all the tweets, one by one, in order to detect those that were written by bots or fake accounts. Because of that, we lost over 50% of the data, reducing the tweets to 24 millions. With VADER we could classify all of them according to the sentiment they express.

4.4. *Bitcoin blockchain information*

The parser that deals with the transformation of blockchain data from binary to text format took about 25 h to complete its task. The binary-to-textual parser reduces every blk.dat size of almost 99.98%, preserving just the necessary information. Furthermore, there were many queries to the Electrs server, to get all the information contained in the Coinbase transactions — another task that required about 9 h.

Once the historical data were obtained, the two tasks were scheduled to work on a daily basis (to process the data of the day before), with a dramatic reduction of running time.

4.5. *Bitcoin trend prediction*

For the variables (n, k, z), we considered the following ranges:

- $n \in \{5, .., 50\}$
- $k \in \{2, .., \lceil(\frac{n}{2})\rceil + 1\}$
- $z \in \{1, .., n\}$

The dataset size ranges from the largest one (obtained with $n = 5$, $k = 2$ and $z = 1$) of 2452 elements, to the smaller one (obtained with $n = 50$, $k = 26$ and $z = 50$) of 49 elements. Thus, training each neural network did not take an excessive amount of time, also because the model is subject to overfitting using a large number of epochs. Actually, it was possible to carry out many tests in a relatively short time. By changing the values of (n, k, z) we created 18121 different datasets (some of them are described in Table 7). For all of them, 60% of the dataset was used for training, 20% for validating the CNN model and the remaining 20% for testing it. Each training takes about one minute on average, so it took almost twelve days to train all the models.

Since 18 121 different datasets were generated, we trained an equal number of neural networks. 26 epochs were considered,

Table 8
Best CNN results compared to LSTM results, for the same (n, k, z) values.

n	k	z	Test set accuracy CNN	Test set accuracy LSTM	Test set size
14	5	5	66.04%	47.83%	98
23	11	4	68.46%	49.26%	122
38	**12**	**1**	**74.24%**	47.58%	**487**
38	12	8	75.84%	49.15%	61
41	5	6	69.59%	48.68%	81
42	3	5	66.15%	48.94%	97
42	6	7	69.04%	47.48%	69

Table 9
Best LSTM results compared to CNN results, for the same (n, k, z) values.

n	k	z	Test set accuracy LSTM	Test set accuracy CNN	Test set size
13	6	5	52.52%	56.43%	247
18	**8**	**7**	**54.31%**	56.73%	**176**
22	10	3	49.46%	61.29%	409
23	4	4	52.65%	63.27%	307
32	16	4	50.86%	49.11%	305
33	3	4	53.63%	51.72%	306
34	16	5	53.38%	51.34%	244

with a batch size of 65 (higher values produced overfitting). The tests showed that overfitting is a tangible problem if the number of epochs increase too much. For all trained neural networks we performed a validation test and we decided to discard all those whose resulting accuracy was less than 65%, in order to screen out the worst models. Furthermore, all those datasets having test set size ≤ 60 were not considered as well. The less data available, the greater is the probability that results do not fit to reality. Dropout was used to prevent model overfitting [14,40,54].

Using the Keras Python package, we built a LSTM model where every layer has one input and one output tensor. The input shape is 1 time step with 17 features; we used 150 neurons to train the model, and we adopted the Root Mean Square Error (RMSE) as the loss function. This model is fit for training 26 epochs with a batch size of 40. Once we got the values from the RMSE, we then analyzed the number of times the neural network predicted the value correctly, following the binary classification described above (UP, DOWN), divided by the total number of forecasts, so that we could compare the results with those obtained by means of the CNN.

In order to provide a consistent comparison, we represent the obtained results in two distinct tables: Table 8 shows the best outcomes regarding the CNN model, compared to those produced by the LSTM model, for the same (n, k, z) values. Among these results, the most reliable is the one corresponding to $(n, k, z) = (38, 12, 1)$, because its test size is quite large (compared, for example, to the result of $(n, k, z) = (38, 12, 8)$, which is only apparently better in terms of accuracy). Table 9 shows the best results obtained with the LSTM model, compared to those obtained with CNN, for the same (n, k, z) values. We observe that the CNN model outperforms the LSTM one, except in the last three cases illustrated in Table 9, where the LSTM model appears to be slightly more accurate.

Finally, in Table 10, we compare the best CNN result with those of the state of the art (summarized in Section 2). It must be considered that the accuracy measures were obtained by evaluating completely different features, timeframes, algorithms and methodologies. Our approach considers a larger number of features and produces an accurate prediction of the average bitcoin trend in the next $k = 12$ days, when the CNN model is adopted. Other approaches use a smaller number of features, as well as a reduced dataset, and consider a few days or months.

Table 10
Comparison between our work and current state of the art.

Method	Number of features	Test set accuracy
Baseline [14]	16	53.4%
Logistic regression [25]	12	66%
SVM [25]	15	65.3%
Neural network [14]	16	55.1%
Random forest [13]	16	57.4%
LSTM [25]	12	67.2%
XGBoost [25]	12	48.3%
CNN	**17**	**74.2%**

4.6. Real-time scenario

In this subsection, we illustrate a specific test we performed with the trained model in order to evaluate it with real-time data. In particular, trying to answer the question "how much benefit a company may obtain by using this model?", we analyzed the 17 features in a time range that goes from 15 February 2020 to 20 October 2020. The model makes predictions based on its training, but it is up to the user to decide which trading strategy suits his/her best.

Algorithm 1 defines the following strategy. Supposing that no purchase order was placed at time t:

- if the model predicts that at time $t + 1$ the value of bitcoin will be greater than the current value at time t, then X bitcoin units are purchased;
- otherwise, if the model predicts that at time $t + 1$ the value of bitcoin will be less than the current value at time t, then no action is taken.

If a purchase order was placed at time t for X unit of bitcoin:

- if the model predicts that at time $t + 1$ the value of bitcoin will be greater than the current value at time t then no action is taken.
- otherwise, if the model predicts that at time $t + 1$ the value of bitcoin will be less than the current value at time t, then X bitcoin units are sold.

Algorithm 1 Calculate CNN model profit

```
$investment \leftarrow 0$
$profit \leftarrow 0$
$t \leftarrow 1$
while ($t < len(bitcoin\_values) - 1$) do
  $change \leftarrow bitcoin\_values[t]/bitcoin\_values[t-1]$
  $prediction \leftarrow CNN.predict(bitcoin\_values[t+1])$
  if ($investment = 0$ && $prediction > bitcoin\_values[t]$) then
    $investment \leftarrow X$
    $profit \leftarrow profit - investment * bitcoin\_values[t]$
  else if ($investment > 0$ && $prediction < bitcoin\_values[t]$) then
    $profit \leftarrow profit + investment * bitcoin\_values[t]$
    $investment \leftarrow 0$
  end if
  $t \leftarrow t + 1$
end while
```

In the considered time range, the model (trained with the dataset corresponding to $(n, k, z) = (38, 12, 1)$) correctly predicted the trend of Bitcoin on 181 days out of 248, having an accuracy of 72.9%, which is very close to the one shown in

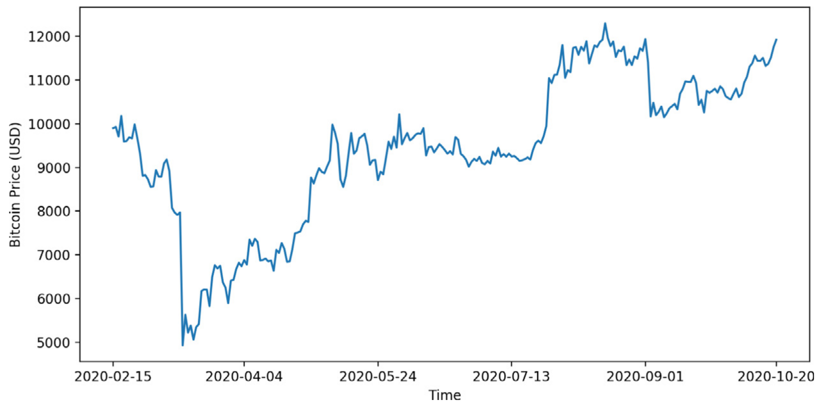


Fig. 14. Bitcoin trend from the 15th of February 2020 to 20th of October 2020.

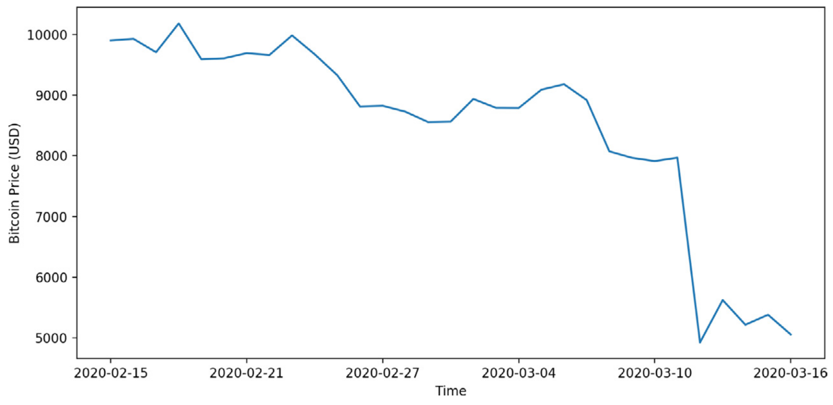


Fig. 15. Bitcoin trend from the 15th of February 2020 to the 16th of March 2020.

Table 10. I.e., the model performed as we expected. Starting from an initial investment of X bitcoin units, by following Algorithm 1, the obtained value was $1.961*X$ (i.e., $+96.1\%$ profit). It should be noted that the bitcoin trend, in the considered period, was mostly increasing (with a final $+20.44\%$), as shown in Fig. 14.

We then repeated the test considering a different time interval, which goes from 15 February 2020 to 16 March 2020. In that period, as shown in Fig. 15, the bitcoin trend was mostly decreasing. In this scenario, the goal is to limit the loss a trading strategy could achieve. The adoption of Algorithm 1 produced a value of $0.283*X$ (i.e., -28.3% loss) units for an initial investment of X units. It is worth noting that on February 15, the value of bitcoin was 9901$, while on March 16 it was 5058.5$ (-48.91%).

Thus, we may conclude that the proposed 1D CNN model significantly increases the profit when the bitcoin trend is bullish and reduces the loss when the trend is bearish.

5. Discussion

Our results are a consequence of the rich datasets we built, based on the integration of sentiment analysis information with bitcoin transaction history and financial indicators. The particular 1D CNN model that we implemented is the second pillar of the proposed bitcoin trend prediction system.

As several studies have shown [55–57], the price of Bitcoin moves following the sentiment expressed by traders, who influence both the large and the small cryptocurrency market investors. There is also a relationship between the bitcoin price and the mining process. Hayes [58] identified that the bitcoin price, as well as other cryptocurrency prices, is determined by the relative differences in the cost of production (mining) on the margin. By analyzing the following features: Block Size, Fees, Mining Difficulty and Number of Transaction, we realized that a strong correlation between the bitcoin price and the mining process persists. Financial indicators, such as those we defined in Section 3.2, have been widely used for decades in the technical and financial analysis. They help and support traders every day during their trading processes. In our system, we computed several financial indicators using the historical price of Bitcoin.

CNNs are complex feed-forward neural networks that learn key features from the dataset during the training process. This automated feature extraction makes CNNs highly suitable and accurate for computer vision tasks such as object/image classification, video/audio signals recognition and time series forecasting. By working on an entirely new approach, which has been explained in Section 3.7, we built a robust 1D CNN model that predicts the bitcoin trend with good accuracy, despite the high volatility of the bitcoin cryptocurrency.

6. Conclusion

In this work, we illustrated a novel approach for bitcoin trend prediction based on 1D CNN. We proposed a methodology for building datasets whose items are characterized by different types of features: bitcoin historical values and financial indicators, Twitter sentiment analysis, Bitcoin blockchain information. We presented a cloud-based system with a highly efficient distributed architecture, which allowed us to collect a huge amount of data and to create thousands of different datasets. We showed that the 1D CNN model we implemented, trained, validated and tested using the aforementioned datasets, allows one to predict the bitcoin trend with higher accuracy compared to LSTM models. Last but not least, we quantified the benefit that a company may obtain, if a simple trading strategy based on the proposed 1D CNN model is used by that company with assumed initial investment of X units. The conclusion was that the proposed 1D CNN model increases the profit when the bitcoin trend is bullish and reduces the loss when the trend is bearish.

6.1. Future work

Regarding future work, we plan to design and implement a system for automatically retraining the machine learning model, leveraging values acquired day by day, in order to further improve the accuracy of the predictions on the test set. Moreover, we will implement a dashboard for showing bitcoin trend prediction in real time.

CRediT authorship contribution statement

Stefano Cavalli: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Michele Amoretti:** Investigation, Methodology, Supervision, Writing - original draft, Writing - review & editing.

Declaration of competing interest

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

References

[1] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Gener. Comput. Syst. 107 (2020) 841–853.
[2] V. Gramoli, From blockchain consensus back to byzantine consensus, Future Gener. Comput. Syst. 107 (2020) 760–769.
[3] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, 2009, Cryptography Mailing list at https://metzdowd.com.
[4] W.J. van der Laan, et al., Bitcoin Core, https://bitcoin.org/en/bitcoin-core/.
[5] J. Moody, M. Saffell, Learning to trade via direct reinforcement, IEEE Trans. Neural Netw. 12 (4) (2001) 875–889.
[6] J. Seiffertt, D. Wunsch, Intelligence in markets: Asset pricing, mechanism design, and natural computation [technology review], IEEE Comput. Intell. Mag. 3 (4) (2008) 27–30.
[7] F.Z. Xing, E. Cambria, R.E. Welsch, Intelligent asset allocation via market sentiment views, IEEE Comput. Intell. Mag. 13 (4) (2018) 25–34.
[8] D.T. Tran, A. Iosifidis, J. Kanniainen, M. Gabbouj, Temporal attention-augmented bilinear network for financial time-series data analysis, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2019) 1407–1418.
[9] S. Karasu, A. Altan, S. Bekiros, W. Ahmad, A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series, Energy 212 (2020) 118750.
[10] S. Karasu, A. Altan, Recognition model for solar radiation time series based on random forest with feature selection approach, in: 2019 11th International Conference on Electrical and Electronics Engineering (ELECO), 2019, pp. 8–11.
[11] J. Kaminski, P. Gloor, Nowcasting the bitcoin market with twitter signals, 2014, arXiv:1406.7577.
[12] M. Matta, M.I. Lunesu, M. Marchesi, Bitcoin spread prediction using social and web search media, in: Workshop on Deep Content Analytics Techniques for Personalized and Intelligent Services (DeCat 2015), 2015.
[13] I. Madan, S. Saluja, A. Zhao, Automated Bitcoin Trading Via Machine Learning Algorithms, Tech. rep., Stanford University, 2014.
[14] A.S. Greaves, B. Au, Using the Bitcoin Transaction Graph To Predict the Price of Bitcoin, Tech. rep., Stanford University, 2015.
[15] S. McNally, J. Roche, S. Caton, Predicting the price of bitcoin using machine learning, in: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2018, pp. 339–343.
[16] S. Kiranyaz, T. Ince, R. Hamila, M. Gabbouj, Convolutional neural networks for patient-specific ecg classification, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 2608–2611.
[17] S. Kiranyaz, T. Ince, M. Gabbouj, Real-time patient-specific ecg classification by 1-d convolutional neural networks, IEEE Trans. Biomed. Eng. 63 (3) (2016) 664–675.
[18] T. Ince, S. Kiranyaz, L. Eren, M. Askar, M. Gabbouj, Real-time motor fault detection by 1-d convolutional neural networks, IEEE Trans. Ind. Electron. 63 (11) (2016) 7067–7075.
[19] L. Zhong, L. Hu, H. Zhou, Deep learning based multi-temporal crop classification, Remote Sens. Environ. 221 (2019) 430–443.
[20] E. Stenqvist, J. Lönnö, Predicting Bitcoin Price Fluctuation with Twitter Sentiment Analysis, Tech. rep., KTH Royal Institute of Technology, 2017.
[21] C.J. Hutto, E. Gilbert, VADER: A parsimonious rule-based model for sentiment analysis of social media text, in: ICWSM, 2014.
[22] A. Mittal, V. Dhiman, A. Singh, C. Prakash, Short-term bitcoin price fluctuation prediction using social media and web search data, in: Twelfth International Conference on Contemporary Computing (IC3), 2019, pp. 1–6.
[23] A. Altan, S. Karasu, S. Bekiros, Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques, Chaos Solitons Fractals 126 (2019) 325–336.
[24] P. Linardatos, S. Kotsiantis, Bitcoin Price Prediction Combining Data and Text Mining, Springer, 2020, pp. 49–63.
[25] Z. Chen, C. Li, W. Sun, Bitcoin price prediction using machine learning: An approach to sample dimension engineering, J. Comput. Appl. Math. 365 (2020) 112395.
[26] Investing, https://www.investing.com.
[27] Similarweb, https://www.similarweb.com.
[28] B. Chez, CoinMarketCap, https://www.coinmarketcap.com.
[29] L. Richardson, BeautifulSoup, https://www.crummy.com/software/BeautifulSoup/bs4/doc/.
[30] C. Ivanov, D. Globa, S. Bokov, Trading View, https://www.tradingview.com.
[31] G. Giaglis, I. Georgoula, D. Pournarakis, C. Bilanakos, D. Sotiropoulos, Using time-series and sentiment analysis to detect the determinants of bitcoin prices, SSRN Electron. J. (2015).
[32] A. Taspinar, Twitterscraper: a scraping tool working on twitter using python as programming language, https://github.com/taspinar/twitterscraper.
[33] A. Kumar, D. Kawahara, S. Kurohashi, Knowledge-enriched two-layered attention network for sentiment analysis, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 2, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 253–258, http://dx.doi.org/10.18653/v1/N18-2041, (Short Papers), URL https://www.aclweb.org/anthology/N18-2041.
[34] J. Ramteke, S. Shah, D. Godhia, A. Shaikh, Election result prediction using twitter sentiment analysis, in: 2016 International Conference on Inventive Computation Technologies (ICICT), Vol. 1, 2016, pp. 1–5, http://dx.doi.org/10.1109/INVENTIVE.2016.7823280.
[35] S. Sohangir, N. Petty, D. Wang, Financial sentiment lexicon analysis, in: 2018 IEEE 12th International Conference on Semantic Computing (ICSC), 2018, pp. 286–289, http://dx.doi.org/10.1109/ICSC.2018.00052.
[36] C.W. Park, D.R. Seo, Sentiment analysis of twitter corpus related to artificial intelligence assistants, in: 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 2018, pp. 495–498, http://dx.doi.org/10.1109/IEA.2018.8387151.
[37] R. Zeyde, Electrs, https://github.com/romanz/electrs.
[38] C. Bai, T. White, L. Xiao, V. Subrahmanian, Z. Zhou, C2p2: A collective cryptocurrency up/down price prediction engine, in: 2019 IEEE International Conference on Blockchain (Blockchain), 2019.
[39] R. Chowdhury, M.A. Rahman, M.S. Rahman, M.R.C. Mahdy, Predicting and forecasting the price of constituents and index of cryptocurrency using machine learning, 2019, arXiv:1905.08444.
[40] A. Dutta, S. Kumar, M. Basu, A gated recurrent unit approach to bitcoin price prediction, J. Risk Financ. Manage. 13 (2020) 23.
[41] L. Alessandretti, A. Elbahrawy, L. Aiello, A. Baronchelli, Anticipating cryptocurrency prices using machine learning, Complexity 2018 (2018) 1–16.
[42] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[43] A. Aimar, H. Mostafa, E. Calabrese, A. Rios-Navarro, R. Tapiador-Morales, I. Lungu, M.B. Milde, F. Corradi, A. Linares-Barranco, S. Liu, T. Delbruck, Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps, IEEE Trans. Neural Netw. Learn. Syst. 30 (3) (2019) 644–656.
[44] S.R. Dubey, S. Chakraborty, S.K. Roy, S. Mukherjee, S.K. Singh, B.B. Chaudhuri, Diffgrad: an optimization method for convolutional neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2019).
[45] J. Park, S. Jo, Bayesian weight decay on bounded approximation for deep convolutional neural networks, IEEE Trans. Neural Netw. Learn. Syst. 30 (9) (2019) 2866–2875.
[46] W. Shi, Y. Gong, X. Tao, J. Wang, N. Zheng, Improving cnn performance accuracies with min–max objective, IEEE Trans. Neural Netw. Learn. Syst. 29 (7) (2018) 2872–2885.
[47] F. Chollet, et al., Keras: high-level api for the implementation of neural networks with python, https://keras.io.

[48] D. Kingma, J. Ba, Adam: a method for stochastic optimization, Int. Conf. Learn. Represent. (2014).

[49] D. Tian, Z. Shi, Mpso: Modified particle swarm optimization and its applications, Swarm Evol. Comput. 41 (2018) 49–68.

[50] Y. Chen, L. Li, X. Zhao, J. Xiao, Q. Wu, Y. Tan, Simplified hybrid fireworks algorithm, Knowl.-Based Syst. 173 (2019) 128–139.

[51] A. Altan, S. Karasu, Recognition of covid-19 disease from x-ray images by hybrid model consisting of 2d curvelet transform, chaotic salp swarm algorithm and deep learning technique, Chaos Solitons Fractals 140 (2020) 110071.

[52] E. Bochinski, T. Senst, T. Sikora, Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3924–3928.

[53] The mnist database of handwritten digits. URL http://yann.lecun.com/exdb/mnist/.

[54] D. Bacciu, F. Crecchi, Augmenting recurrent neural networks resilience by dropout, IEEE Trans. Neural Netw. Learn. Syst. 31 (1) (2020) 345–351.

[55] M. Matta, M.I. Lunesu, M. Marchesi, Bitcoin spread prediction using social and web search media, 2015.

[56] K. Vytautas, D. Niels, D.W. Jochen, Using sentiment analysis to predict interday bitcoin price movements, J. Risk Financ. 19 (1) (2018) 56–75, http://dx.doi.org/10.1108/JRF-06-2017-0092.

[57] G. Cheuque Cerda, J.L. Reutter, Bitcoin price prediction through opinion mining, in: Companion Proceedings of the 2019 World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 755–762, http://dx.doi.org/10.1145/3308560.3316454.

[58] A. Hayes, Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin, Telemat. Inform. 34 (2017) 1308–1321.