

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Agent-Based Artificial Market Model for Studying the Bitcoin Trading

LUISANNA COCCO¹, ROBERTO TONELLI², AND MICHELE MARCHESI.²

¹Department of Electrical and Electronic engineering, University of Cagliari, Italy

²Department of Mathematics and Computer Science, University of Cagliari, Italy

Corresponding author: First A. Author (e-mail: luisanna.cocco@diee.unica.it).

This research is partially supported by the Research Project Fondazione di Sardegna Algorithms for Approximation with Applications [Acube], annuity 2017; and by the Research Project AIND–Amministrazioni e Imprese Native Digitali–Programmazione Unitaria 2007–2013–P.O. FESR 2007/2013–Interventi a sostegno della competitività e dell’innovazione, annuity 2013.

• **ABSTRACT** The objective of this work is to simulate the trading of the currency pair BTC/USD, investigating through the theory of the genetic algorithms the best sets of trading strategies, simulating through a realistic order book the bitcoin price formation, and reproducing a bitcoin price series that exhibits some “stylized” facts found in real-time price series.

In this artificial market model two kinds of agents, Chartists and Random traders, perform trading. Chartists trade through the application of trading rules. Specifically, a part of Chartists trades applying the best sets of trading rules selected by a genetic algorithm that simulates a trading system, based on four technical analysis indicators, searching for parameters of each indicator that guarantee the highest profits in the training period; the remaining part trades applying trading rules choosing their parameters in a random way. On the contrary Random Traders trade without applying any trading strategy, issuing in a random way sell or buy orders. Results show that the best sets of rules found guarantee the highest profits both in the training and in the testing periods, and perform well also in the artificial market model where the Chartists who adopt the best sets of trading rules are able to achieve higher profits.

• **INDEX TERMS** Artificial cryptocurrency market, Trading Strategies, Genetic Algorithm, Simulation

I. INTRODUCTION

In December 2017, the daily trade volume in the cryptocurrency market overtook the \$50 billion mark, highlighting how hot the cryptocurrency market has become in 2017 thanks to the eye-catching increase of about 1,500% of bitcoin value against the dollar. It is a huge daily volume of trade, that is only two orders of magnitude smaller than daily volumes in the foreign-exchange market, which reach daily volumes of over \$5 trillion [28]. At that time many exchanges temporarily halted on-boarding of new users because they were not able to guarantee proper ticket and verification response times to such enormous number of new account registrations [7].

Exchanges are the digital marketplaces in which the bitcoin trade takes place. They are online platforms that act as an intermediary between buyers and sellers, who can buy and sell bitcoins using different fiat currencies or altcoins. Today bitcoin exchanges are an integral part of the virtual currency world, and the higher are the trading volumes of a given exchange, the more this exchange is trusted in the market. Bitfinex, Coinbase, Kraken, Bitstamp, Binance, Huobi, and

LocalBitcoins are some of the most popular Bitcoin Exchange online.

To trade bitcoins a trader has to sign up for a bitcoin exchange to set up an account, and after the exchange confirms her/his identity, the trader can deposit funds into the account and begin to trade bitcoins. Typically, traders make profits by buying bitcoins at a low cost and selling them when the price goes up. Bitcoin price is very volatile, so traders –usually small-time, retail investors, even if some huge institutions are playing in the market– have good probability to make substantial profits.

This paper presents an artificial market model to simulate the trading of the currency pair BTC/USD. It is built on a work by Cocco and Marchesi [11]. In this artificial market, Random Traders issue orders in a random way, compatibly with their available resources, issuing buy and sell orders with the same probability. Contrary to Random Traders, Chartists open a position after having evaluated a technical analysis indicator. If the indicator gives them a signal to open a position, the traders open it. A technical analysis indicator

forecasts the price movements through the analysis of past market data [12]. In recent years algorithmic trading based on this analysis has evolved exponentially. This is because, contrary to human dealers, algorithmic trading learns from thousands of sources of information simultaneously, avoiding emotional influence.

In this work a part of Chartists trade using trading rules, previously selected by a genetic algorithm (GA). GA is an optimization technique applied to solve numerical optimization problems by simulating the mechanism by which Genes are updated in nature in order to optimize the response to changing environments. GAs were first developed by Holland [18] for simulating the process of natural evolution in which species change over time. In this process, as the population grows, the organisms fight to get resources. Only the strongest ones survive, thus they bring more descendants to the new generations. Since financial markets are continuously changing, the trading strategies need to adapt to the new conditions. As a result, GAs represent a very suitable method to be adopted for finding the strategies that allow to achieve the higher profits.

The aim of this study is to investigate, through the theory of GAs, the best sets of trading strategies to be applied in the BTC/USD market, simulating through a realistic order book the bitcoin price formation, and reproducing a bitcoin price series that exhibits some "stylized facts" found in real-time price series. The best sets of trading strategies are investigated through a GA that simulates a trading system equal to that proposed by Boboc and Dinica [5]. This algorithm takes into account four opening rules, based on four technical analysis indicators, and two exiting rules (see work [5] for more details or the section III-B and the section IV).

Our findings show that it is possible to forecast Bitcoin market movements by only analyzing historical prices. This is in contrast with the efficient market hypothesis [5], [14]. The efficient market hypothesis [14] assumes that traders cannot take their trading decision solely based on historical data since at every instant the price depends on all information in the market at that instant. Boboc and Dinica [5] examined the efficiency of EUR/USD market through a GA that simulates a trading system. The results of their analysis are in accordance with the efficient market hypothesis. Their trading system is able to get good returns only in the training period, and not in the testing period.

However, as emphasized in [5] empirical findings have shown that this hypothesis may be questionable. Hasan et al. [17] noticed that factors like return, market capitalization, book-to-market ratio and market value influence the share returns. Alvarez et al. [2], Abounoori et al. [1], and Kim et al. [22] observed that the efficiency degree of financial markets changes over time.

To our knowledge this work is the first one that aims to study the BTC/USD market through an agent based model including realistic trading strategies. The model was validated by performing several statistical analysis in order to study the stylized facts of bitcoin price and returns as in work [11].

The research proceeds as follows. Section II describes related works. Section III presents the artificial market model in which agents buy and sell bitcoins. Specifically, section III-A presents the agents present in the market, section III-B presents the trading rules, and III-C the orders and the mechanism of price formation. Section section IV presents the GA used to select the best sets of trading rules. Section V presents and discusses the performance of the GA and the results obtained simulating the proposed artificial financial market. Finally Section VI concludes the paper.

II. RELATED WORK

To our knowledge there are very few works in literature about Bitcoin trading strategies. It is worth to cite the work by Detzel et al. [13] that, according to authors claim, is the first study of technical analysis applied to Bitcoin. In this work the authors analyze trading strategies based on moving averages of Bitcoin prices over different horizons to predict Bitcoin returns. In the last years the first works on bitcoin price prediction start to appear. The work by Indera et al. [19] presents a multi layer perceptron to predict the bitcoin price. The work by Jang and Lee [21] predicts the bitcoin price using bayesian neural network and blockchain information. Nowadays, there are numerous on-line platforms, such as CoinTracking, BitcoinCharts, Bitcoinity and BitcoinWisdom, that allow traders to use several technical analysis tools to identify trends and market sentiment useful for entering a trade.

The model proposed in this paper is built on the work by Cocco and Marchesi [11], including realistic trading strategies in order to reproduce the trading of the currency pair BTC/USD, and the main stylized facts of the bitcoin price series, following the well known agent-based approach (see works [6], [8] and [20] for reviews about this approach).

Contrary to the work [11], in this paper traders use rules to form expectations on prices or on gains, as in the works by Chiarella et al. [9] and by Licalzi and Pellizzari [24], in which traders use rules to form expectations on stock returns. Specifically, in this work, as already mentioned in Section I, Chartists open or close a position according to the evaluation of a technical analysis indicator (such as the Exponential Moving Average, the Moving Average Convergence Divergence, the Relative Strength Index, or the Genetic Filter). In addition, the GA, that selects the best set of trading rules applied by a part of Chartists, implements a mechanism similar to that proposed in the work by Tedeschi et al. [27]. In the work just quoted traders imitate the expectations of the most successful traders. In this work the GA selects the trading rules that guarantee the higher profits.

The proposed model implements a mechanism for the formation of the bitcoin price based on an order book equal to that proposed in [11], without maximizing a their own expected utility function before placing a buy or a sell order, and choosing the size of the order. This is in contrast with the approach adopted in the works [9], [24] and [27], in which traders maximize a their own expected utility function before issuing an order.

III. THE MODEL

This work presents an agent-based artificial cryptocurrency market in which two kinds of agents, Random traders and Chartists, perform trading by buying or selling bitcoins. Random traders issue orders in a random way. Chartists issue orders following the signals generated from precise trading rules.

A. THE AGENTS

All traders present in the market at the initial time $t = 0$ hold both an amount $c_i(0)$ of fiat currency (cash, in US dollars) and an amount $b_i(0)$ of cryptocurrency (bitcoins), where i is the trader's index. The i -th trader entering the market at time $t_i^E > 0$, holds only an amount $c_i(t_i^E)$ of fiat currency (cash, in dollars), where the apex E stands for entering.

As in work [10], at the initial time the sum of the bitcoins owned by each trader represents 60% of the total number of bitcoins in the market, because we assumed that 40% of bitcoins are not available for trading. Over time, as it happens in reality, new bitcoins enter the market. We assumed that 60% of new bitcoins enter the market every 90 days¹.

We assigned the new bitcoins to a randomly extracted fraction of traders, giving them an amount of bitcoins proportional to those already owned, respecting the Gibrat principle of preferential attachment (richer gets richer) [16], [29].

The initial wealth distribution of traders both in crypto and fiat cash follows a Zipf law (see work [23]) and is generated before the beginning of the simulation following the approach described in the works [10] and [11].

As regards the number of traders, N_T , in the market at time t we computed it through a fitting curve. We fit the curve $N_T(t)$ through seven figures following the approach proposed in work [11]².

The fitting curve of the number of traders N_T is defined by using a general exponential model:

$$N_T(t) = a * exp^{b*t} \quad (1)$$

where a and b are the best fitting parameters and are equal to $1.744e + 04$ and 0.002465 respectively, and the initial value of t is 1824 that corresponds to January 1st, 2014, that is the date in which our simulations start.

¹To manage the computational load of the simulation we sized the artificial market at about 1/2500 of the real market. Hence we divided by 2500 both the number of mined bitcoins per day and that of the traders. As a result to not deal with a very smaller number of bitcoins, we assumed to cumulate the mined bitcoins within 90 days and then to assign them to the traders.

²Specifically, these seven figures are: 1 person in January 2009; 2769 people in May 2010; 30,589 people in September 2010; 280,000 people at the end of 2013; 1,000,000 people in April 2014; 10,000,000 people in December 31th, 2017; 2.1 billion people in December 31st, 2021. The first five figures, that are extracted from the work [11], refer to people who downloaded the Bitcoin mining software and/or owned bitcoins since the authors assumed that in the early days of Bitcoin system these people were the people who trade bitcoins (see the work above quoted for more details). Instead, the sixth and the seventh figures refer to the maximum number of people that owns bitcoins in 2017 and to the maximum possible number of people that could be interested in the future to the bitcoin trading, respectively [3].

Following the approach proposed in the works [10] and [11], given the number of traders entering the market at a given instant t , a trader belongs to the Chartists' population with a probability, $percC$, equal to 0.6, and to the Random Traders' population with a probability, $percR$, equal to 0.4, being $percC + percR = 1$.

1) Chartists

Chartists are in the market to speculate, aiming to get high profits by issuing buy and sell orders. Each Chartist trades by using a set of trading rules, composed by an opening rule and an exiting rule. This means that a Chartist enters the market opening a position, hence issuing a sell or a buy order, after having evaluated his opening rule that gives buying and selling recommendations to him.

After having opened a position the Chartist waits for the best instant to exit the position in order to take the profit or to cut the losses. Hence, a Chartist closes a position issuing a buy or a sell order, after having evaluated his closing rule that gives buying and selling recommendations to him, as the opening rule does.

Note that Chartists can issue an order (opening/closing position) only when their previous order (closing/opening position) is fulfilled completely.

Specifically Chartists trade using precise set of trading rules chosen by themselves in a random way when they enter the market from a set of best trading rules, or from a set of random trading rules³.

The former rules – the best– are selected through a GA that chooses the rules' parameters in order to maximize the profitability of traders (see Sections III-B and IV for more details). The latter rules – those random– are not optimized and their parameters are selected in a random way.

Henceforth we denote the part of Chartists who adopt the best set of rules by C_{bR} , and the other part by C_{rR} .

2) Random Traders

Random traders are in the market for various reasons, such as to diversify their portfolio, or to satisfy a need for cash, but not to speculate. They issue buy and sell orders with the same probability and compatibly with their available fiat and crypto cash.

B. TRADING RULES

As in work by Boboc and Dinica [5] our model includes 6 rules, precisely four rules for opening a position and two for exiting the position, and 24 parameters.

1) The rules for Position Opening

The rules for position opening are defined as in [5].

• Rule 1: Filter.

The well known rule called "filter" is a trading strategy based on the fractional changes in price from previous

³The probability, $percC_B$ to use the best rules is equal to 0.48, and the probability, $percC_R$, to use the random rules is equal to 0.12, being $percC_B + percC_R = percC$.

lows and highs. Its definition is based on the five parameters:

1. $filter_{flag}$;
2. $filter_{periods}$, noted n ;
3. $filter_{increaseS}$, noted p ;
4. $filter_{decreaseS}$, noted q ;
5. $filter_{booleanS}$.

The first parameter, $filter_{flag}$, turns on or off the rule. It takes the values 0 or 1. The value 0 turns off the rule, while the value 1 turns it on. The second parameter, $filter_{periods}$, indicates the previous price to be considered. The third and fourth parameters are variables that indicate the change in a currency pair. In the forex market these changes are expressed in pip, being a pip (short for point in percentage) a very small measure of change in a currency pair. The fifth parameter, $filter_{booleanS}$, is a boolean signal that defines the trading signals. It takes the values 0 or 1 with same probability. If it acquires value equal to 0, it gives a buy signal if the price increases more than p pips and a sell signal if the price decreases more than q pips. Vice versa in the case it acquires value equal to 1. Hence, it gives a sell signal if the price increases more than p pips and a buy signal if the price decreases more than q pips.

- Rule 2: Relative Strength Index (RSI).

The RSI values are computed as follows:

$$RSI_t(n) = \frac{\sum_{i=t-n}^t \frac{max_i(P_i - P_{i-1}, 0)}{n}}{\sum_{i=t-n}^t \frac{max_i(P_i - P_{i-1}, 0)}{n} + \sum_{i=t-n}^t \frac{max_i(P_{i-1} - P_i, 0)}{n}} \quad (2)$$

where P is the closing price of the period and n is the number of the periods used to compute the RSI. The RSI value gives signals of buying and selling depending on the fact of being higher or lower than the so called overbought and oversold signals. This rule is characterized by the following parameters:

6. rsi_{flag} ;
7. rsi_n ;
8. rsi_{os} ;
9. rsi_{ob} ;
10. $rsi_{boolean}$.

The sixth parameter turns on or off the rule, hence it is a boolean signal. The seventh parameter, rsi_n , is the number of periods to be considered in the computation of the average rises and falls. The eighth and ninth parameters are the oversold and overbought signal respectively. They are constants whose values are taken from the literature [5]. Finally, the tenth parameter is a boolean signal, defining the trading signals, and takes values 0 or 1 with same probability. If it acquires value equal to 0, it gives a buy signal when the RSI value is smaller than rsi_{os} and a sell signal when the RSI value is higher than rsi_{ob} . Vice versa if it acquires value equal to 1, it gives a sell signal when the RSI value is smaller

than rsi_{os} and a buy signal when the RSI value is higher than rsi_{ob} .

- Rule 3: Exponential Moving Average (EMA).
The exponential moving average is a weighted moving average. The EMA values are defined as:

$$EMA_t(n, Close) = \frac{2}{n+1} Close_t + EMA_{t-1} \left(1 - \frac{2}{n+1}\right) \quad (3)$$

where EMA , n and $Close$ are the value of this indicator at the previous instant, the number of periods and the closing price of the period, respectively.

The EMA rule has two parameters:

11. ema_{flag} turns on and turns off the rule.
12. ema_n is the period n .

The twelfth parameter, ema_n , indicates that the closing prices for the past n days have to be taken into account to compute this average. This rule gives a buy signal when the close price is higher than $EMA(n)$ and a sell signal when the close price is lower than $EMA(n)$.

- Rule 4: Moving average convergence divergence (MACD).

The MACD indicator is defined through the computation of several moving averages. It is defined as:

$$MACD_t(p, q, m) = MACD_t(p, q) - Signal_t(m) \quad (4)$$

where:

$$MACDline_t(p, q) = EMA_t(p, close) - EMA_t(q, Close) \quad (5)$$

$$SignalLine_t(m) = EMA_t(m, MACD_t(p, q)) \quad (6)$$

- p , q , and m are the number of periods of the short and the long exponential moving average, and of the MACDline indicator respectively.
- $Close$ is the closing price of the period.

The parameters of the MACD rule are:

13. $macd_{flag}$ takes the values 0 or 1. Value 0 turns off the rule, while value 1 turns on it.
14. $macd_{periodsS}$, noted p .
15. $macd_{periodsL}$, noted q and with the restriction $q > p$
16. $macd_{periodsN}$, noted m , is the moving average of the difference between the short and the long moving average.
17. $macd_{booleanS}$ takes the values 0 or 1 with same probability.

The value of the $macd_{booleanS}$ parameter defines the trading signals. If it acquires value equal to 0, it gives a buy signal when the short moving average is higher than the long one and a sell signal otherwise.

On the contrary, if it acquires value equal to 1, it gives a buy signal when the difference between the short moving average and the long one is higher than the value of the signal, and a sell signal otherwise.

Note that if there is a currently open position, the opening rule is ignored. In addition when all the flag that turn on or off the rules take the value 0, or when more than one takes the value 1, then we randomly change the value for one of them to 1 and set to 0 the others. The flags have the same probability to be chosen.

2) The Rules for Exiting the Position

As already mentioned, once a position is opened the traders wait the best instant in which exiting the position in order to take the profit or to cut the losses. The rules for exiting the position, as those for position opening, are defined as in work [5].

- Rule 5: Fixed exit levels (FEL).

This rule is characterized by three parameters:

18. fel_{flag} turns on and turns off the rule (boolean).
19. fel_{tp} is a threshold value entering in the evaluation of an exiting position to take profit.
20. fel_{sl} is a threshold value entering in the evaluation of an exiting position to stop loss.

The FEL rule closes a long position if the price is higher than, or equal to, fel_{tp} (exits a long position taking a profit) or smaller than, or equal to, fel_{sl} (exits a long position stopping the loss). Vice versa, the rule exits a short position if the price is smaller than, or equal to, fel_{tp} or is higher than fel_{sl} .

- Rule 6: Trailing exit levels (TEL).

This rule is characterized by four parameters:

21. tel_{flag} turns on and turns off the rule (boolean).
22. tel_{tp} is a threshold value entering in the evaluation of an exiting position to take profit.
23. tel_{sl} is a threshold value entering in the evaluation of an exiting position to stop loss.
24. tel_{tl} is a threshold value entering in the evaluation of an exiting position with $tel_{tl} < tel_{tp}$.

This rule exits a position as the FEL rule does, but it updates the values of tel_{tl} and tel_{tp} over time when no exiting condition is verified (see work [5] for more details).

Note that an exiting rule can be active only if a position is already opened, and at a given instant only one exiting rule can be active.

C. ORDERS AND MECHANISM OF PRICE FORMATION

As in the works [10] and [11] the trading mechanism is based on a realistic order book that sorts buy and sell orders in two lists and matches them allowing to fulfill compatible orders. Precisely buy and sell orders are sorted respectively in descending and ascending order with respect to the limit price, and orders with the same limit price are sorted in ascending order with respect to the order issue time. Note that the limit price models the price to which a trader desires to conclude his transaction. It depends on the current price, $p(t)$, and on the random variable, $N_i(\mu, \sigma_i)$. As in works [10] and [11], the limit prices, $p_{b,i}^l$ and $p_{s,i}^l$, related to buy and sell order, respectively, and the price, p_T , are defined as follows:

$$p_{b,i}^l(t) = p(t) * N_i(\mu, \sigma_i) \quad (7)$$

$$p_{s,i}^l(t) = \frac{p(t)}{N_i(\mu, \sigma_i)} \quad (8)$$

where

- $N_i(\mu, \sigma_i)$ is a random variable drawn from a Gaussian distribution with average $\mu \simeq 1$ and standard deviation σ_i .
- $p(t)$ is the current bitcoin price. It is strictly linked to the price to which a transaction is performed, p_T . The mechanism for the formation of the price p_T is the following:
 - if $p_{b,i}^l > 0$ and $p_{s,i}^l = 0$ then $p_T = \min(p_{b,i}^l, p(t))$;
 - if $p_{s,i}^l > 0$ and $p_{b,i}^l = 0$ then $p_T = \max(p_{s,i}^l, p(t))$;
 - if $p_{b,i}^l = 0$ and $p_{s,i}^l = 0$ then $p_T = p(t)$;
 - if $p_{b,i}^l > 0$ and $p_{s,i}^l > 0$ then $p_T = \frac{p_{b,i}^l + p_{s,i}^l}{2}$.

For more details about the order book, the amount of the issued orders, the definition of limit prices and the price clearing mechanism see work [11] to which this work refers.

IV. GENETIC ALGORITHM

Usually a GA searches for optimal solution of a problem, starting from a first population of solutions. Then it recombines them by introducing elements of disorder, thus creating new solutions in an attempt to converge to the optimal solution.

In this work we refer to the GA implementation proposed in [5]. The GA simulates a trading system in which each individual is represented by a set of technical analysis rules (called chromosomes in the GA theory). Each rule is characterized by two or more parameters (called genes in the GA theory).

More precisely the algorithm we used proceeds according to the following scheme.

- 1) The GA generates the first population of solutions (chromosomes) randomly. It generates 100 individuals in a random way, that is 100 sets of technical analysis rules choosing their parameters randomly. In our model a chromosome represents rules.
- 2) The GA evaluates each individual by computing the profit or loss that the individual gets over the training period.
- 3) The GA sorts the 100 individuals according to their profit or loss that represent the fitness function evaluated by the GA.
- 4) The GA creates the new generation proceeding by steps as follows.
 - The GA inserts into the new generation the individual who get the highest profit (elitism).
 - Each of the 100 individuals can become a parent for the new generation with a probability that

TABLE 1: Ranges in which the GA searches for the values of some parameters.

Subsamples Parameters	2014-2018 Range
2: $filter_{periods}$	[1, 15]
3: $filter_{increaseS}$	[0.02705, 0.079]
4: $filter_{decreaseS}$	[0.02705, 0.079]
8: rsi_n	[2, 10]
9: rsi_{os}	[15, 35]
10: rsi_{ob}	[65, 85]
12: ema_n	[2, 10]
14: $macd_{periodS}$	[5, 90]
15: $macd_{periodL}$	[10, 100]
16: $macd_{periodN}$	[5, 25]
19: fel_{tp}	[0.0074, 0.2]
20: fel_{sl}	[0.0025, 0.079]
22: tel_{tp}	[0.0074, 0.2]
23: tel_{sl}	[0.0025, 0.079]
24: tel_{tl}	[0.0025, 0.079]

depends on his profit, hence on his ranking. Higher is his profit, higher is his probability to be a parent. We divided the individuals into ten classes and assigned a probability to each class according to the best performances for profit⁴.

- According to their probability the GA chooses random pairs of two individuals to generate a new individual. The genes of the new individuals are chosen following the same approach proposed by Boboc and Dinica [5] (crossover). Specifically, being the genes 24, enumerated from 1 to 24 as in section III-B, the GA selects randomly a number n between 1 and 24, and assigns to the new individual the genes from 1 to n of one parent and the genes from $n+1$ to 24 of the other parent. With this mechanism the GA generates 80 individuals.
- The GA generates the remaining 19 individuals in a random way to increase the diversity.

5) Once the GA creates all the 100 individuals of the new generation, this generation becomes the actual generation, and the GA repeats the steps 2, 3 and 4.

6) The GA repeats the procedure from steps 2–5 until it reaches 80 such iterations given a higher number of iterations keep the results substantially unchanged⁵.

The GA implements all the rules described in section III-B. The values of the rules' parameters are set as in work [5]. It is worth emphasizing how we defined the values for the parameters 3 ($filter_{IncreaseS}$), 4 ($filter_{DecreaseS}$), 19 (fel_{tp}), 20 (fel_{sl}), 22 (tel_{tp}), 23 (tel_{sl}), and 24 (tel_{tl}). These parameters take values within a range comprised between a minimum and a maximum value (see Table 1). In the work [5] these minimum and maximum values are expressed in pip. In the work just quoted the pip is the very small measure of change

⁴We implement this procedure as in work [5], see this for more details and for the calibration of the algorithm.

⁵Refer to the work [5] for more details about the implementation of every step of the algorithm

in the EUR/USD currency pair and is equal to $\frac{1}{10000}$. Contrary to the EUR/USD currency pair, in the BTC/USD currency pair this change is much higher and much more variable over time. We set these minimum and maximum values taking into account the variation percentage of the price, hence the returns. We defined these values starting from the computation of the cumulative function of the returns of the EUR/USD daily series so that the probabilities correspond each other. We computed the return considering a time interval equal to one day given that our simulation proceeded by daily steps. Computed this function, the value of the probability to get the minimum or maximum value defined in [5] is known. For symmetry with the work just quoted, we computed the cumulative function of the returns of the BTC/USD daily series, considering a time interval equal to one day. Computed this function, we get for the minimum or maximum value of the parameters above mentioned, the values associated to the same probability to get the minimum, or the maximum value in the work [5] respectively.

The GA selects at each cycle of 80 iterations a set of rules, that includes an opening rule and an exiting rule, characterized both by specific values for their parameters. These parameters' values are those that guarantee the highest profitability. We run the algorithm 100 times extracting as a result 100 different sets of trading rules. These 100 sets of rules represent the sets of rules from which the Chartists select the set of rules to adopt to perform trading in the proposed artificial model.

Note that the GA implements a trading system in which each set of trading rules models a trader who issues buy and sell orders without any constraint of resources. The trader opens a position issuing a buy or a sell order depending on the signal provides by his/her opening rule. After having opened a position, the trader can close it only when his/her opening order is fully executed and his/her exiting rule gives his/her the signal to close the position taking the profit or stopping the loss.

It is worth underlining that buy and sell orders issued during the training and the testing period involve the purchase or the sale of only one bitcoin, and for each issued buy or sell order the system automatically generates a sell or buy order respectively, to fully execute all issued orders. These last two assumptions keep the algorithm as simple as possible allowing to compute the profit of each trader easily. In addition the last one guarantees that all issued orders are executed in a time acceptable compatibly with our training period, and above all it avoids continuous order imbalances in the book on the buy or sell side.

The database used to training and to test the GA is the daily series of the BTC/USD currency pair from January 1st, 2014 to December 31st, 2018. We consider the bitcoin closing prices and specifically 1826 values. This daily time series has been separated in two series with same length. The former, used in the training period, considers the closing prices of the first two and a half years and is used for finding the set of trading rules that achieves the highest performance.

The latter, used in the testing period, tests the performance of the sets of rules found in the training period considering the closing prices in the last two and a half years, that hence represent the testing period.

In order to evaluate the GA performance, we perform analysis with different training and testing sets. For example we partitioned the original sample into three subsamples, where each of them individuates a different market regime. The first subsample contains the bitcoin daily price from January 1st, 2014 to December 31st, 2016; the second contains the bitcoin daily price from January 1st, 2017 to December 31st, 2017; and finally, the third contains the bitcoin daily price from January 1st, 2018 to December 31st, 2018.

To give an idea of the different market regimes in the three subsamples taken into account let us illustrate for each subsample some statistical values, such as the average, the standard deviation, the minimum and the maximum value. The average is equal to \$455, \$3,970 and \$7,601 respectively for the first, the second and the third subsample; the standard deviation is equal to \$179, \$4,022 and \$2,470 respectively for the first, the second and the third subsample; the minimum value is equal to \$177, \$775 and \$3,236 respectively for the first, the second and the third subsample; and finally the maximum value is equal to \$975, \$19,475 and \$17,527 respectively for the first, the second and the third subsample. Each subsample was divided in two series of the same length. We used the former series for the training; and the latter series for testing the performance of the sets of rules found in the training period. The GA was repeated three times, one time for each subsample. As a result we obtained three best sets, each of them constituted by 100 sets of trading rules. GA's results shown lower performance than that obtained considering the whole series.

In addition running the proposed model under the assumption that Chartists adopt a set of rules choosing it between these three sets depending on the time in which they perform trading, we obtained results similar to those obtained running the model considering the only one best set of rules generated by GA considering the whole series. This is due to the substantial differences between the trading system implemented in the proposed model and that implemented through the GA as we are going to see next.

V. RESULTS

In this section we first describe the results of the GA, then we illustrate the results of the model's simulation.

A. GENETIC ALGORITHM RESULTS

Before starting the model's simulation we run the GA described in section IV. We implemented this algorithm in Smalltalk language and run it in order to find the optimal set of rules, hence the optimal values of the parameters of the set of rules.

The algorithm searches for the values of parameters that guarantee the higher profits as in work [5]. It is applied 100 times on the training period, in order to find the character-

TABLE 2: Descriptive statistics of the 100 sets of best solutions.

Parameters	2014 - 2018 Mean (Std)
2: $filter_{periods}$	8.08 (3.98)
3: $filter_{increaseS}$	0.054 (0.015)
4: $filter_{decreaseS}$	0.05 (0.013)
8: rsi_n	6.04 (2.08)
9: rsi_{os}	24.65 (5.32)
10: rsi_{ob}	74.31 (5.57)
12: ema_n	6.35 (2.54)
14: $macd_{periodS}$	49.58 (24.67)
15: $macd_{periodL}$	70.58 (25.17)
16: $macd_{periodN}$	14.46 (5.94)
19: fel_{tp}	0.096 (0.053)
20: fel_{sl}	0.042(0.021)
22: tel_{tp}	0.113 (0.051)
23: tel_{sl}	0.041 (0.023)
24: tel_{tl}	0.036 (0.022)

istics of the best 100 sets of rules. Then the performance of these 100 sets were assessed on the out-of-sample series, hence on the testing period.

Table 2 shows the descriptive statistics of the 100 sets of trading rules selected by the GA.

Note that some parameters take values out of the ranges described in Table 1. This is because they are updated during the simulation as described in the section III-B.

The 100 individuals, who apply the best rules selected by the GA, achieve on average a profit equal to \$4,531.5 (std. \$63.2) in the training period, and equal to \$82,374.8 (std. \$1,456.2) in the testing period.

To evaluate the goodness of the results we applied the same trading system to 100 random individual, that is to 100 individuals who apply the six rules described in section III-B selecting them, included their parameters' values, in a random way. As expected, these random individuals achieve an average lower profit. Precisely, they achieve on average a profit equal to \$2,271.51 (std. \$1,177.97) in the training period, and equal to \$43,679.1 (std. \$29,564.9) in the testing period, confirming the high performance of the individuals that adopt the rules selected by the GA⁶.

B. MODEL RESULTS

The proposed model was implemented in Smalltalk language and run over a period that ranges between January 1st, 2014 to December 31st, 2018. The simulation period was thus set to 1459 steps, with a step corresponding to one day.

We run the model 100 times with the same initial conditions, but different seeds of the random number generator in order to assess the robustness of our model and the validity of our statistical analysis. As regards the calibration of the model, we set the variables' values as in work [11] unless otherwise stated in Section III.

⁶ Note that as expected the standard deviation value is very lower when we take individuals who adopt the best sets of rules into account, than when we consider individuals who adopt the rules choosing them in a random way.

1) Traders' statistics, simulated bitcoin price, and volume of exchanged bitcoins

In the following Section we illustrate some statistics of the traders' populations, such as the crypto and the fiat cash, and the average total wealth per capita of the traders⁷ obtained running the *Base Run*.

Figures 1 and 2 show the average and the standard deviation across all Monte Carlo simulations respectively of the crypto and fiat cash per capita for the Chartists' population, that includes C_{bR} and C_{rR} , and for the Random Traders' population⁸⁹. Fig. 1(a) illustrates the average crypto cash over time and shows as C_{bR} are able to get cash higher than others traders, thanks to their ability of trading and to an intrinsic mechanism of the model.

The simulator assigns the new bitcoin mined to the traders in a way proportional to their crypto cash. As a result over time the simulator assigns a number of mined bitcoins to C_{bR} higher than those assigned to Random Traders and to C_{rR} , given C_{bR} own a higher and higher crypto cash (see fig. 1 in which the crypto cash of C_{bR} is higher than that of C_{rR} and the trend of the crypto cash of C_{bR} is increasing contrary to the trend of the crypto cash of Random Traders).

The decreasing trend of the Random Traders' crypto cash implies that these traders issue sell orders characterized by smaller and smaller amounts and buy orders characterized by larger amounts than those of the sell orders. In turn this mechanism creates an imbalance in the book, in which the amount of the bitcoins of the unsatisfied buy orders is much bigger than the amount of the bitcoins of the unsatisfied sell orders. Consequently, the buy orders of Random Traders are hardly executed due to this imbalance and to the sorting of the list of all the orders in the book. This explains the increasing trend of the fiat cash and the decreasing trend of the average total wealth per capita of the Random Traders (see figs. 2 and 3).

Fig. 3 shows the average total wealth per capita showing as C_{bR} are able to get higher profits than other traders'

⁷The average total wealth per capita is computed as the sum between the fiat cash and the crypto cash multiplied by the bitcoin price. In order to highlight only the trading profits it does not include the initial crypto and fiat cash owned by every trader at time t^E , that is the time in which every trader enters the market.

⁸Remember that C_{bR} denotes the part of Chartists who adopt the best rules and C_{rR} denotes the part of Chartists who adopt the random rules.

⁹In addition to the *Base Run* we planned several simulation sets, in order to study the sensitivity of the model to the percentage, $percC_B$, of Chartists C_{bR} , and to the probability, $probNoLimit$, to issue market orders. This is because through the analysis of these variables we can analyse the substantial differences between the trading system implemented in the proposed model and that implemented through the GA. The first variable implies the presence of different traders with different trading strategies in the proposed model. The second variable implies the presence of limit orders and market orders in the proposed model. On the contrary the trading system implemented through the GA includes only a limited number of Chartists and orders with unitary amount and without limit price. In addition it generates automatically sell/buy orders to fully execute all issued orders just they are issued. Results do not highlight any significant pattern for increasing values of $percC_B$ and $probNoLimit$. All simulation results show that Chartists, C_{bR} , are able to get higher profits than other traders' populations over time.

populations.

Fig. 4(a) and (b) shows the average and the standard deviation of the simulated bitcoin price. The average value of prices steadily increases with time in contrast with what happens in reality, where bitcoin price has abrupt falls. This is because, as shown in works [10] and [11], in the proposed model there exists an intrinsic mechanism that does not allow to reproduce the peak and the abrupt fall of the real bitcoin price: the average price tends to the ratio of total available cash to total available bitcoins, since new traders bring in more cash the price tends to increase.

Finally fig. 5 shows the average and the error bar (standard deviation) of the volume of bitcoins exchanged. Specifically, fig. 5 (a) describes the simulated volume of bitcoins exchanged over time. Fig. 5(b) illustrates the real volume extracted by blockchain.info, that provides an average of the trading volume on major bitcoin exchanges. The average value over time is slightly higher than the real one (see fig. 5(a) and (b)). It is equal to $2.6 * 10^5$ in the simulated market and equal to $0.7 * 10^5$ in the real one.

Fig. 6 describes the volume of the bitcoins exchanged in simulated market (multiplied by 2500 that is the resize we applied to the real market), in three simulation runs. Also in the simulated market the volume of the exchanged bitcoins exhibits several peaks which deviate from the average value more than the peaks present in the real volume (see fig. 5(b)). Results highlight that our model is able to reproduce the real volume of bitcoin exchanged, after having multiplied the simulated volume by the scaling factor of our simulation (2500).

All results described until now refer to the *Base Run* that implements the model described in Section III considering the bitcoin price as an endogenous variable implemented through the mechanism described in Section III-C. We also run another simulation set considering the bitcoin price as an exogenous variable. We set it equal to the real bitcoin price and analysed the total wealth per capita for the different kinds of traders, as in the *Base Run*.

Simulation results showed that C_{bR} get higher profits than Random Traders but not higher than C_{rR} . This is due to two reasons. The first is the substantial difference between the order book implemented in the GA and that implemented in the proposed model. The second is the different trend of the real and simulated bitcoin price. The latter is steadily increasing, and the former is not.

In the case of exogenous price C_{bR} gets higher profits than C_{rR} if we define not optimal ranges for the initial solutions from which the GA starts.¹⁰

2) Stylized facts in the real and simulated Bitcoin market

We studied the real and simulated bitcoin price series between January 1st, 2014 to December 31st, 2018. Specifically

¹⁰Remember that in the case of endogenous price the ranges in which the GA selects the initial solutions are the optimal ranges as described in section IV.

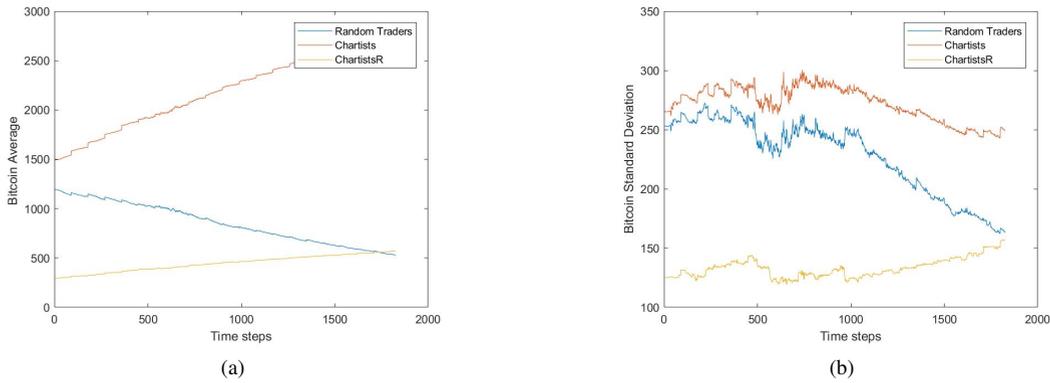


FIGURE 1: Average (a) and standard deviation (b) across all Monte Carlo simulations of the crypto cash in the *Base Run*. In the legend the label ‘Chartists’ refers to C_{bR} . Instead the label ‘ChartistsR’ refers to C_{rR} , and the label ‘Random Traders’ refers to Random Traders.

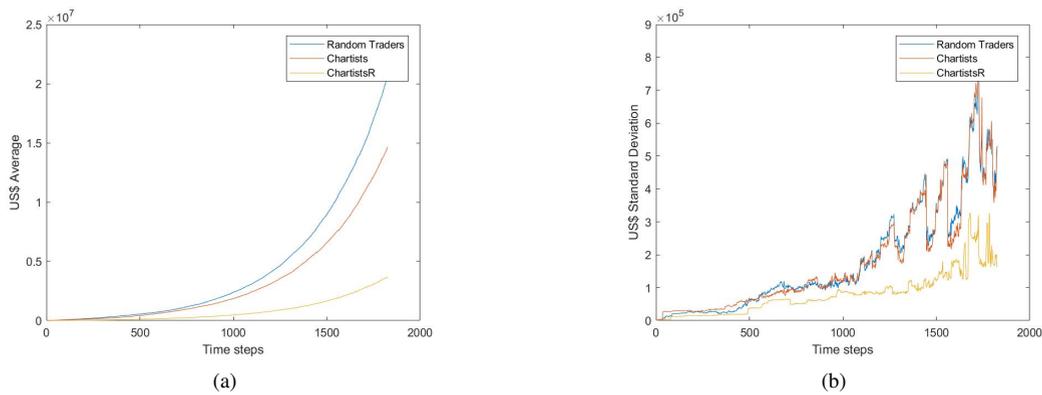


FIGURE 2: Average (a) and standard deviation (b) across all Monte Carlo simulations of the fiat cash in the *Base Run*. In the legend the label ‘Chartists’ refers to C_{bR} . Instead the label ‘ChartistsR’ refers to C_{bR} , and the label ‘Random Traders’ refers to Random Traders.

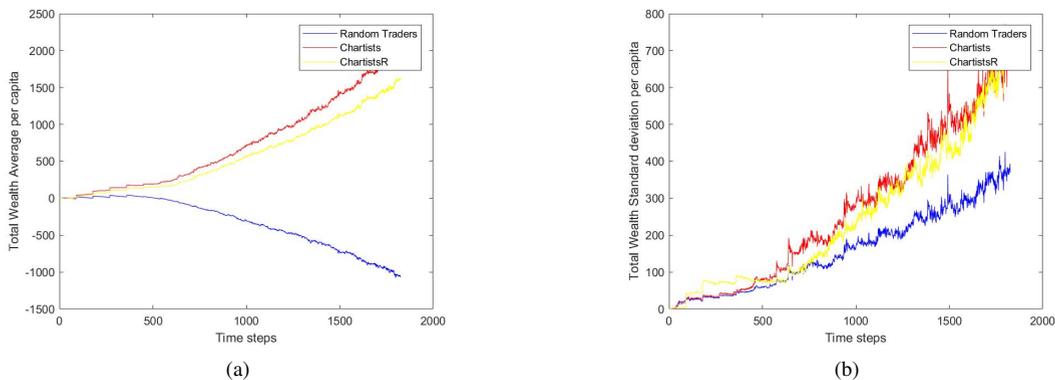


FIGURE 3: Average and standard deviation across all Monte Carlo simulations of the total wealth per capita in the *Base Run*.

we analysed and studied some stylized facts of the real and simulated bitcoin time series.

As regards the studied stylized facts, they are the same as in

works [10], [11]. We studied the unit-root property, the fat tail phenomenon, and the Volatility Clustering ([4], [25], [26]).

In order to study the unit-root property we applied the

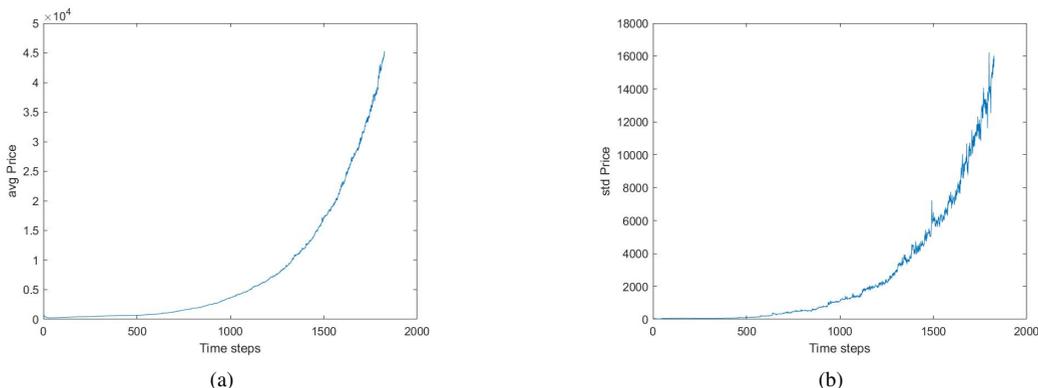


FIGURE 4: Average and standard deviation across all Monte Carlo simulations of the simulated bitcoin price in the *Base Run*.

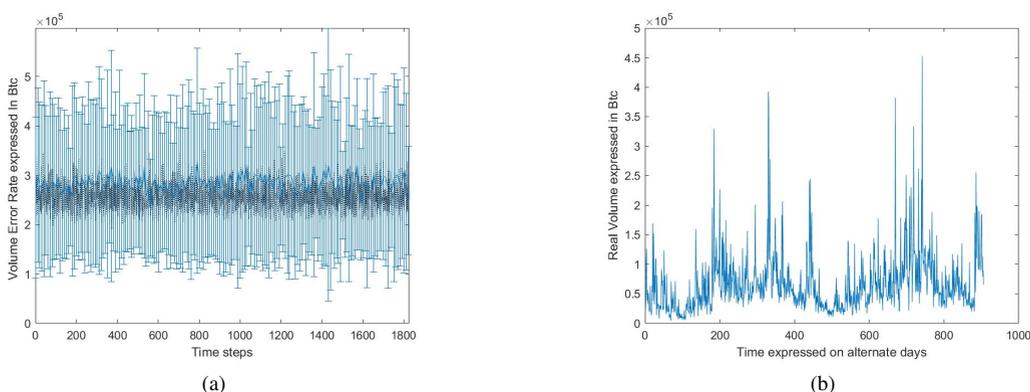


FIGURE 5: Average across all Monte Carlo simulations of the volume of btc (a) in simulated market (multiplied by 2500 that is the resize applied by us to the real market), and (b) in real market.

Augmented Dickey-Fuller test, under the null hypothesis of random walk without drift, to the real and simulated series of bitcoin daily prices and to the real and simulated series of bitcoin daily price logarithms we considered.

For both the series, real and simulated, at levels 1, 5, and 10% we cannot reject the null hypothesis since the τ_1 statistic, and the percentiles of the τ_1 statistic, for real price series (real price logarithm series), and for simulated price series (simulated price logarithm series) respectively, at each level are always higher than the corresponding critical values. The critical values are equal to -2.58 , -1.95 and -1.61 at levels 1, 5, and 10%, respectively, with 1,826 observations. For real price series and real price logarithm series τ_1 statistic are -1.44 , and -1.12 , respectively. For the simulated series the percentiles of the τ_1 statistic across all Monte Carlo simulations are described in Table 3.

The second studied property is the fat-tail phenomenon. The distribution of real and simulated returns is a leptokurtic distribution, so that we can infer a "fat tail". The Kurtosis value of the real price returns and that of the real price absolute returns are equal to 8.97 and 12.62 respectively, consequently the distribution of returns is more outlier-prone

TABLE 3: Percentile Values of the τ_1 statistic for the null hypothesis of random walk without drift across all Monte Carlo simulations. Statistics of price logarithm series are in brackets.

Percentile Value			
.25	.50	.75	.975
1.3 (2.1)	1.6 (2.3)	1.9 (2.5)	2.8 (2.9)

than the normal distribution. Table 4 describes the 25th, 50th, 75th and 97.5th percentiles pertaining to kurtosis of the price returns across all Monte Carlo simulations. The simulated kurtosis is a bit higher than the real case. Its median is equal to 28 for price returns, which is a value not too distant from the real one.

The third property studied is the Volatility Clustering. Fig. 7, (a) and (b), show the autocorrelation functions of the real price returns and of the absolute returns, respectively, at time lags between zero and 20. Table 5 shows the 25th, 50th, 75th and 97.5th percentiles pertaining to average and standard deviation of the autocorrelation of simulated raw

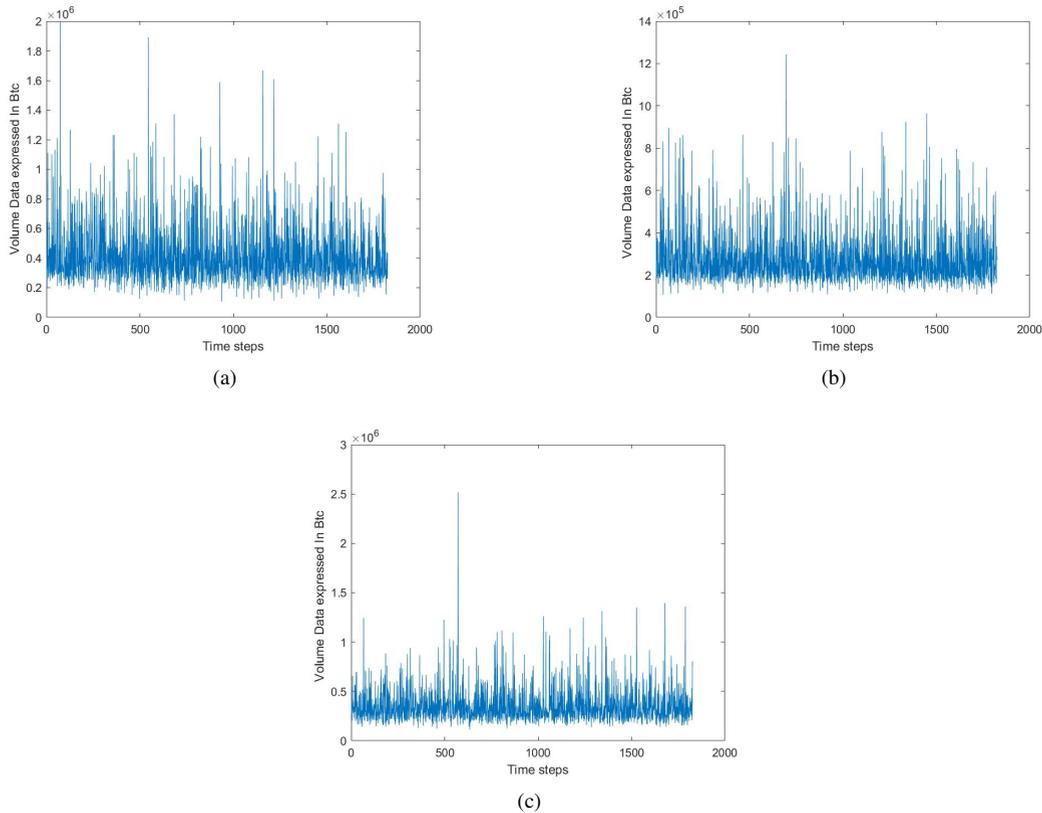


FIGURE 6: Volume of btc in simulated market (multiplied by 2500 that is the resize we applied to the real market), in three simulation runs.

TABLE 4: Percentile Values of the Kurtosis value of the price returns and of the price absolute returns (in brackets) across all Monte Carlo simulations.

	Percentile Value			
	.25	.50	.75	.975
	19 (24)	28 (37)	45 (55)	377 (460)

TABLE 5: Percentile Values of average and standard deviation of the autocorrelation of raw returns ($avgRet_{raw}$ and $stdRet_{raw}$, respectively) and those of absolute returns ($avgRet_{abs}$ and $stdRet_{abs}$, respectively) across all Monte Carlo simulations.

Descriptive statistics	Percentile Value			
	.25	.50	.75	.975
$avgRet_{raw}$	0.007	0.009	0.01	0.013
$avgRet_{abs}$	0.06	0.08	0.1	0.1
$stdRet_{raw}$	0.03	0.03	0.033	0.04
$stdRet_{abs}$	0.03	0.03	0.033	0.05

returns, and those of simulated absolute returns, at time lags between 1 and 20, across all Monte Carlo simulations. Fig. 7 and Table 5 show that the autocorrelation of raw returns acquires smaller values than those of the absolute returns, demonstrating the presence of volatility clustering both for

the real and the simulated price series.

VI. CONCLUSIONS

In this work, we propose a heterogeneous agent model of the Bitcoin market with the aim to study and analyze the trading of the currency pair BTC/USD from January 1st, 2014 to December 31st, 2018.

The proposed model simulates the bitcoin trading, by implementing different trading strategies and a price clearing mechanism based on a realistic order book. In this market Chartists and Random Traders perform trading.

Chartists trade applying trading rules choosing them from two sets – one constitutes by rules whose parameters are chosen by a genetic algorithm that selects the trading rules that maximize profits and the other constitutes by rules whose parameters are chosen in a random way. Random Traders trade without applying any trading rules.

The main result of the model is the fact that some key stylized facts of bitcoin real price series and of Bitcoin market are very well reproduced. Specifically, the model reproduces quite well the unit-root property of the price series, the fat tail phenomenon of the price returns, the volatility clustering of the price returns, and the volume of bitcoins exchanged. Further simulation results show that the trading rules selected by the genetic algorithm guarantee higher profits than those

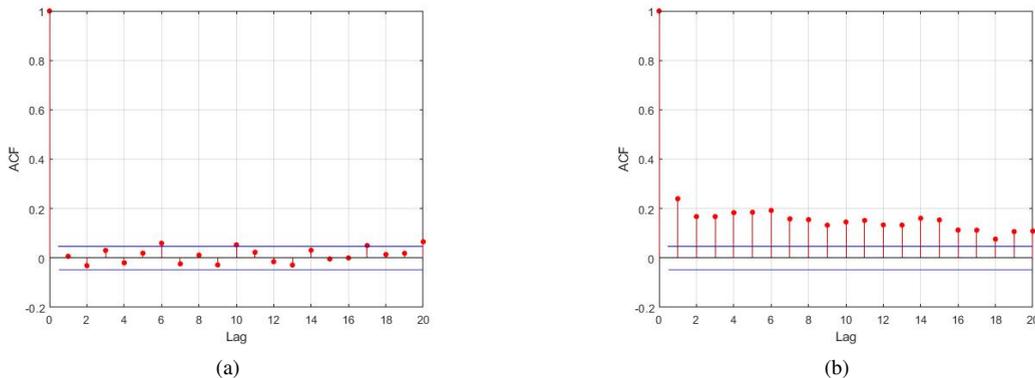


FIGURE 7: Autocorrelation of (a) raw returns, and (b) absolute returns of bitcoin prices.

obtained by Random Traders and by Chartists, C_{rR} , who adopt the trading rules choosing their parameters in a random way.

Furthermore the genetic algorithm’s results show a good performance both in the training and testing period.

To our knowledge this work is the first one that aims to study the BTC/USD market through an agent based model including realistic trading strategies. This first work, due to the computational complexity of the trading system, implemented through the genetic algorithm, considers only buy and sell orders with unitary amount and for each issued buy or sell order the system automatically generates a sell or buy order respectively. In future research we will investigate the possibility to implement a more complex algorithm.

REFERENCES

[1] Abounoori E, Shahrazi M, Rasekhi S, An investigation of Forex market efficiency based on detrended fluctuation analysis: A case study for Iran. *Physica A* 391: 3170–3179, (2012).

[2] Alvarez–Ramirez J, Rodriguez E, Espinosa-Paredes G, Is the US stock market becoming weakly efficient over time? Evidence from 80-year-long data, *Physica A* 391, 5643–5647 (2012).

[3] Anonymous, How many people touched Bitcoin up to 2017 and what is the current adoption pace?, [https://steemit.com\(2017\)](https://steemit.com(2017)).

[4] Begusić S., Kostanjcar Z., Stanley H. E., Podobnik B., Scaling Properties of Extreme Price Fluctuations in Bitcoin Markets, *Physica A* 510, 400-406 (2018)

[5] Boboc I-A, Dinica M.-C., An algorithm for testing the efficient market hypothesis, *PLoS ONE* 8 (10) 2013.

[6] Chakraborti A, Toke I.M, Patriarca M, Abergel F. Econophysics review: II. Agent-based models. *Quantitative Finance*, 11(7), 1013–1041 (2011).

[7] Chaparro F., Some of the biggest crypto exchanges are shutting out new users because they can’t keep up with demand, <http://www.businessinsider.com/crypto-exchanges-are-shutting-out-new-users-because-they-cant-keep-up-with-demand-2017-12?IR=T> (December 27th, 2017).

[8] Chen S.H, Chang C. L, Du Y. R. Agent-Based Economic Models and Econometrics. *The Knowledge Engineering Review*, 27, 187–219 (2012).

[9] Chiarella C., Iori G. Perello J. The impact of heterogeneous trading rules on the limit order book and order flows. *Journal of Economic Dynamics and Control*, 33(3), 525-537(0) (2009).

[10] Cocco L, Concas G, Marchesi M. Using an Artificial Financial Market for studying a Cryptocurrency Market. *Journal of Economic Interaction and Coordination*, Springer, DOI:10.1007/s11403-015-0168-2 (2015).

[11] Cocco, L., Marchesi, M., Modeling and simulation of the economics of mining in the bitcoin market, *PLOS ONE* 11 (10), 1–31 (October 2016).

[12] Dempster MAH, Jones CM, A real-time adaptive trading system using genetic programming. *Quantitative Finance* 1, 397–413 (2001).

[13] Detzel, Andrew L. and Liu, Hong and Strauss, Jack and Zhou, Guofu and Zhu, Yingzi, Bitcoin: Learning and Predictability via Technical Analysis. Paris December 2018 Finance Meeting EUROFIDAI - AFFI. Available at SSRN: <https://ssrn.com/abstract=3115846> or <http://dx.doi.org/10.2139/ssrn.3115846> (2019)

[14] Fama E, Efficient capital markets: a review of theory and empirical work, *Journal of Finance* 25(2), 383–417 (1970).

[15] Garcia D, Schweitzer F. Social signals and algorithmic trading of Bitcoin.R. *Soc. open sci.*2: 150288. <http://dx.doi.org/10.1098/rsos.150288> (2015)

[16] Gibrat, R., *Les inégalités économiques*, Paris, France: Librairie du Recueil Sirey (1931).

[17] Hasan MZ, Kamil AA, Mustafa A, Baten MA (2012) Stochastic Frontier Model Approach for Measuring Stock Market Efficiency with Different Distributions, *PLoS ONE* 7(5), e37047 (2012).

[18] Holland JH, Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery* 3, 297–314 (1962).

[19] Indera N.I., Yassin I.M., Zabidi A., Rizman Z.I. Non-linear Autoregressive with Exogeneous input (narx) bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators. *Journal of Fundamental and Applied Sciences*, 9, No 3S (2017)

[20] Iori G., Porter J. Agent based Modelling for Financial Markets, Chapter for *Handbook on Computational Economics and Finance*. Eds. Shu-Heng Chen and Mak Kaboudan, OUP Oxford University Pres (2014).

[21] Jang H., Lee J., An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. *IEEE Access*, 6, 5427-5437, (2018).

[22] Kim JH, Shamsuddin A, Lim KP (2011) Stock return predictability and the adaptive market hypothesis: Evidence from century-long U.S. data. *Journal of Empirical Finance* 18: 868–879.

[23] Levy, M., Solomon, S., New evidence in the power-law distribution of wealth, *Physica A* 242 (1-2), 90–94 (1997).

[24] LiCalzi, M., Pellizzari, P. Fundamentalists clashing over the book: a study of order-driven stock markets. *Quantitative finance*, 3(6), 470–480 (2003).

[25] Lux T, Marchesi M. Volatility Clustering in Financial Markets: a Microsimulation of Interacting Agents. *International Journal of Theoretical and Applied Finance*. 3(4): 675-702. doi: 10.1142/S0219024900000826 (2000)

[26] Pagan A. The econometrics of financial markets. *J. Empirical Finance*. 3: 15-102. doi: 10.1016/0927-5398(95)00020-8 (1996)

[27] Tedeschi G., Iori G. Gallegati M. Herding effects in order driven markets: The rise and fall of gurus. *Journal of Economic Behavior and Organization*, 81(1), 82–96 (2012).

[28] Williams-Grut O., The cryptocurrency market is now doing the same daily volume as the New York Stock Exchange, <http://uk.businessinsider.com/daily-cryptocurrency-volumes-vs-stock-market-volumes-2017-12?IR=T>, December 20 (2017).

[29] Yule, G., “A Mathematical Theory of Evolution Based on the Conclusions of Dr. J. C. Willis, F.R.S.,” *Philosophical Transactions B*, CXCIII, 21–87 (1924).



LUISANNA COCCO received her PhD in Electronic and Computer Engineering in 2013 from the University of Cagliari. From June 2013 to date she is a member, as a postdoctoral researcher fellow, of the Agile Group, which is a research group in the field of software engineering at the Università degli Studi di Cagliari, Italy. Her research interests are in the area of the modeling of the complex systems, in particular economic and financial systems with heterogeneous agents. Since 2014, she

has extended her research interest to the modeling and simulation of the cryptocurrencies systems publishing papers of great significance for the scientific community.



ROBERTO TONELLI is temporary researcher and professor at University of Cagliari – Italy. He obtained the Abilitazione Scientifica Nazionale Universitaria of MIUR as Professore Associato for the scientific sector 09/H1, sistemi di elaborazione delle informazioni, and has a PhD in Computer Engineering and another PhD in Physics. His research interests are widespread and multidisciplinary. The main topic of his research has been the study of power laws in software systems within

the perspective of describing software quality. Since 2014, he has extended his research interest to the Blockchain technology.



MICHELE MARCHESI has been Full Professor since 1994 at the Faculty of Engineering of the University of Cagliari. Since 2016 he is Full Professor at the Department of Mathematics and Computer Science of the University of Cagliari where he teaches the Software Engineering course. He is the author of over 200 international publications, including over 70 in the magazine. He was one of the first in Italy to deal with POO, since 1986. He was founding member

of TABOO, the Italian association on object-oriented techniques. He has also worked on object analysis and design, UML language and metrics for object-oriented systems since the introduction of these research themes. In 1998 he was the first in Italy to deal with “Extreme Programming” (XP) and agile methodologies for software production. He organized the first and most important world conferences on “Extreme Programming and Agile Processes in Software Engineering” in Sardinia in 2000, 2001 and 2002. Since 2014, being among the first in Italy, he has extended his research interest to blockchain technologies, obtaining significant results in the scientific community.

• • •