JCBI

Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 09 Issue 01
2025

Research Article
https://doi.org/10.56979/901/2025

A Core Moving Average Indicator based Artificial Market Model for Studying the Bitcoin Trading

Arslan Baig[1], Kashmala Sahar[1], Muhammad Amjad[1], Naila Nawaz[1], Mueen ud Din[1], and Muhammad Arslan Rauf[1,2*]

[1]Riphah College of Computing, Riphah International University, Faisalabad, Pakistan.
[2]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China.
[*]Corresponding Author: Muhammad Arslan Rauf. Email: marslanrauf@hotmail.com

Received: April 05, 2025 **Accepted:** May 12, 2025

Abstract: The Bitcoin cryptocurrency market exhibits complex dynamics that challenge traditional strategy development. The market consists of two types of agents; Random Traders who submit buy or sell orders without any strategic rationale and Chartists trade the finest sets of trading strategies based on the Genetic Algorithms (GAs) theory modelling bitcoin price formation over a representative directive manuscript and recreating BTC price series that displays several formal features found in real-time price series. A key issue of algorithmic trading is the identification parameter configurations that consistently yield profitable outcomes across varying market conditions. To resolve this we simulate trading behavior within an agent-based artificial chartist. The proposed chartists operate using five key indicator including Filter, Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and Simple Moving Average (SMA). A subset of these agents employs a genetic algorithm to optimize the parameters of these indicators for maximum profitability during a training period, while others use randomly selected parameter configurations. In comparison to a buy and hold place in BTC. Our trading method provide high alpha, utility, sharpe ratio gains and significantly reduce the severity of drawdowns. Simulation results indicate that the genetically optimized chartists using SMA with a range between 7 and 53 achieving a mean value of 21.31 and a standard deviation of 17.13 outperform both randomly parameterized chartists and non-strategic random traders.

Keywords: Simulated Cryptocurrency Market; Trading Strategy Evaluation; Genetic Algorithm Optimization; Simple Moving Average (SMA); Algorithmic Trading; Market Simulation

1. Introduction

According to a study conducted in December 2017, the trading volume of cryptocurrency exceeded \$50 billion. One wonders which currency led to such a dramatic increase in cryptocurrencies. tudies indicate that in 2017. Bitcoin experienced a substantial appreciation of approximately 1500% against the US dollar accompanied by a significant surge in daily trading volume. With just two small orders making more than \$5 trillion a day from the cryptocurrency market [28]. This was because it was not possible to guarantee the verification of the account to the new customers due to the killing of new customers [7].

The market in which bitcoin is traded is called the digital exchange market or digital market [20]. The digital trading market facilitates online trading to its buyers and sellers. Bitcoin exchange has become an integral part of virtual currency exchanges in the global world. Bitcoin exchanges are the most trusted in the digital market as bitcoin trading volume is higher than all other currencies. Prominent online platforms facilitating Bitcoin trading include exchanges such as BFX, CBASE, KRK, BSTP, BNB, HBP, HBG and LBC. According [17] Stock returns are impacted by various factors including prior returns market capitalization book-to-market ratio and overall market valuation. As highlighted in studies [1, 2, 22] efficiency of financial markets is dynamic and changes over time.

The digital market has a complete channel for bitcoin and other digital currency exchanges. To avail this facility the trader signs up in the digital market and after complete verification becomes an account holder and can deposit his money in the digital account. There is usually a simple formula for making a profit in which the trader buys the currency at a low price and as the market rises he sells the currency and makes a good profit. In the digital market bitcoin exchange traders small investors and owners of large enterprises think of making a good profit by playing easily [3].

Genetic algorithms were initially developed to model the gradual evolutionary process capturing how organisms adapt and transform over successive generations [12, 18]. In natural evolution as an organism nurtures it struggles to utilize the full resources. Generation comes into being. As the commercial market is changing day by day the business model and methodology needs to be adapted to the new environment and law. Genetic algorithms are used to check the best points which is a comparison to the commercial process described by [5, 14].

In recent years Bitcoin has emerged as a volatile and decentralized financial instrument exhibiting unique market dynamics that differ significantly from traditional assets. Traditional econometric and statistical models often fall short in capturing the complexity of its price formation particularly the nonlinear interactions between various market participants. This limitation motivates the adoption of agent-based modeling (ABM),which allows the simulation of heterogeneous trader behaviors and the emergent properties of decentralized markets. Unlike conventional models ABMs can incorporate micro-level trading rules and decision-making processes making them well-suited to reproduce stylized facts observed in real-world Bitcoin price series such as fat-tailed return distributions volatility clustering and persistent autocorrelations. Therefore, this study adopts an ABM approach to explore how trading strategies optimized via GA influence price dynamics in an artificial Bitcoin market [15].

Previous agent-based models of cryptocurrency markets such as the one proposed in [11], have laid foundational work in simulating trading environments; howeve they often fall short in capturing the nuanced behaviors and statistical properties observed in actual bitcoin markets. These models typically rely on static or overly simplified trading strategies lacking mechanisms for dynamic adaptation or learning. Furthermore, many omit the use of realistic order book simulations which are essential for accurately reflecting market microstructure. As a result such models struggle to reproduce key stylized facts like heavy-tailed return distributions volatility clustering and the absence of autocorrelations in returns. These shortcomings highlight the need for more sophisticated modeling techniques that incorporate adaptive agents and data-driven rule evolution [18].

To address the limitations of existing models in replicating the complex nonlinear behaviors observed in the Bitcoin market this study proposes an enhanced agent-based artificial market. Unlike prior work which primarily focused on generic trading behaviors our model introduces a more dynamic framework by evolving trading strategies through genetic algorithm. A key innovation lies in the incorporation of technical indicators particularly the Simple Moving Average (SMA) whose parameters are optimized during the training phase. This allows Chartist agents to systematically learn and adapt trading rules that reflect the real-world stylized facts such as volatility clustering and fat-tailed return distributions more accurately than static rule-based or random-agent models. By simulating these behaviors within a realistic limit order book environment the model provides a more granular understanding of how micro-level agent strategies can drive macro-level price dynamics in cryptocurrency markets.

The rest of the paper is structured as follows: Section 2 provides a review of the related literature. The model of an artificial market where agents trade bitcoins is presented in Section 3 that shows the market agents, trading regulations and the orders and the price creation method. The GA that was used to choose the optimal sets of trade rules is presented in Section 4. The GA's performance and the outcomes of simulating the planned artificial financial market are shown and discussed in Section 5. Paper is finally concluded in Section 6.

2. Related Work

While numerous studies have addressed Bitcoin price prediction and trading strategies relatively little work has explored transaction simulations using agent-based models that reflect realistic trading dynamics. One influential contribution is the study by [13] which utilized moving averages to forecast Bitcoin prices and examined various trading strategies. Other approaches have employed machine learning techniques such as perceptrons and Bayesian Neural Networks (BNNs) to estimate trading values with varying levels of accuracy [19].

Several platforms including CoinTracking, BitcoinCharts, and Bitcoinity provide real-time trading environments where users can apply technical indicators to interpret market trends [21]. These platforms serve as the foundation for many empirical studies but often fail to model the underlying mechanics of order execution and market formation.

The model presented in this paper is primarily inspired by the agent-based framework of [11], which simulates BTC/USD trades using a realistic order book mechanism. While [11] incorporates essential stylized facts of the Bitcoin price series our work extends this model by integrating a more sophisticated mechanism for order matching incorporating order size and direction without requiring agents to explicitly optimize utility functions. In contrast to utility-maximizing strategies used in [9], [24], and [27], our model assumes bounded rationality where agents act based on heuristics rather than calculated expectations.

Additionally, chartist behavior in our model is guided by procedural indicators such as Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) and a genetic filter. The application of GA as outlined in [27] plays a pivotal role in optimizing the set of transaction rubrics utilized by chartist agents. However, focuses on rule optimization in isolation our model embeds these optimized strategies within a competitive artificial market allowing for dynamic interaction between rule-based and random traders. This allows us to assess not only the profitability of trading strategies but also their resilience and performance under varying market conditions.

By combining the realistic price formation mechanism of [11] with an evolutionary trading system inspired by [27] our approach offers a more comprehensive and empirically grounded simulation of Bitcoin market behavior.

3. Proposed Methodology

In this study two categories of agents Random and Chartists engage in the buying and selling of cryptocurrencies such as Bitcoin within an agent-based trading environment [6, 8]. Trader's issue arbitrary orders based on their experience. Chartists place orders following signals generated by sound trading rules.

3.1. The Agents

At the beginning of time t = 0, each trader in the market has a fiat amount in cash US dollars equal to c_i (0) and a cryptocurrency amount equal to b_i (0) here i denotes the index corresponding to the i-th individual trader. A trader trade in market at time when t_i^E greater than zero with fiat money in cash dollars equal to c_i (t_i^E) where the letter E stands for Entry.

As proposed by previous research [10] a percentage of the bitcoins is about 40% is non-tradable at the beginning equated to real-world limitations such as misplaced coins or storage for years. Subsequently only 60% of all the bitcoins are tradable at the beginning. Newly mined bitcoins enter the flow and according to market forces 60% of newly mined coins enter tradability every 90 days. To manage computational complexity the simulated market was scaled down to approximately 1/2500 of the actual Bitcoin market. Parameters a and b which govern the model dynamics were selected to reflect this scaling while maintaining balance in the simulation environment. Accordingly, both the average daily trading volume and the number of active Bitcoin traders were proportionally reduced by a factor of 2500. In order to avoid the impracticality of simulating transactions involving extremely small fractions of Bitcoin newly generated bitcoins were aggregated over a 90-day period before being distributed among traders. This approach ensured more realistic and computationally efficient transaction modeling within the agent-based framework.

We followed the principle of the force of preferential attachment which revealed that the rich get richer. By allocating new bitcoins to a randomly selected portion of traders they are given the number of bitcoins in proportion to those already in possession [16, 29].

In accordance with Zipf's law [23] the initial distribution of wealth among traders engaged in both cryptocurrency and fiat currency markets is determined prior to the commencement of simulations following the methodologies outlined in [10, 11].

To estimate the temporal evolution of the number of traders NT(t) active in the Bitcoin market we employed an exponential fitting methodology as originally introduced in [11]. Rather than listing all historical values in detail we based the curve fitting on selected data points that reflect key milestones in the growth of Bitcoin adoption. These include early-stage adoption figures (e.g., 1 user in January 2009 and ~280,000 by the end of 2013) as well as broader estimates of Bitcoin users in later years reaching over 10 million by the end of 2017 and approximately 2.1 billion by 2021. These values reflect both active traders and general Bitcoin holders assuming many early adopters were also engaged in mining and trading activities. Based on the available data points the growth trajectory of traders was modeled using a generalized exponential function defined as follows:

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

where t is the time in days starting from January 1, 2014 (i.e., t = 1824) a = 1.744×10^4 is the estimated initial number of traders at the start of our simulation b = 0.002465 is the exponential growth rate of the trader population over time. These parameters were derived through non-linear regression and represent the best fit for observed user growth trends allowing the model to simulate a dynamically evolving market population.

According to the methodology [10, 11] considering the number of traders entering the market at a specific point in time t. A chartist agent has probability percC = 0.6 and a random trader the probability of passing is percR = 0.4. The equation to find the population of chartists is percC + percR = 1.

3.1.1. Chartists

Technical analyst are people in marketplace who predict the current situation in the trading market and place buy and sell orders in an attempt to maximize their profits. Each chartist is guided by a predefined set of trading rules comprising an entry rule and an exit rule. Prior to participating in the market the chartist evaluates the entry condition and analyzes all signals generated by the rule set. Based on this assessment, the chartist submits either a buy or sell order.

According to the given instruction chartist awaits a suitable moment for closing the trade after it has been initiated for taking profit or cutting the loss on the occurrence of a bad loss. After exiting the market for trading chartist demonstrates his closing rules by placing an order. Keep in mind that chartists are only able to enter orders where their entered order fully meets the open/close requirements. When entering a trading market chartists select an finest set of trading rubrics from finest trading rubrics or from randomly selected set of trading rubrics. As $percC_b + percC_R = percC$, probability of applying the best rules $percC_B$ is 0.46, and the probability of applying the random rules $perc_{CR}$ is 0.12c. Here C_{bR} represents portion of Technical analyst obeying finest rules and C_{rR} represents portion obeying random rules.

As per the above principle the optimum set is selected by a GA that determines the parameters of the rules to optimize the profits of the traders. A set at random is not computed using a formula and the parameters are chosen randomly. We will therefore refer to the part of the chartist making use of the best practices as CbR and the rest as C_{rR}.

3.1.2. Random Traders

Random traders do not speculate. They enter the market to diversify their portfolio meet short term financing requirements.With equal likelihood and depending on their fiat and cryptocurrencies avai lable with them they order to buy/sell.

The overall proposed approach is illustrated in Figure 1.

3.2. Trading Rules

The current model has six rules by following the four rules with 24 parameters are used to open an order and two rules are used to close it.

3.2.1. Criteria for Initiating a Trading Position

Guidelines for Initiating Position are listed below.

1. Filter

A rule filter refers to a trading strategy based on detecting minor price fluctuations relative to previous highs and lows. Based on these five parameters it is defined as follows:

1. $filter_{flag}$, boolean variable that activates or deactivates the filter rule;
2. $filter_{periods}$ (n), number of periods considered for evaluating price changes;
3. $filter_{increase}S$ (p), threshold for detecting upward price movements;
4. $filter_{decrease}S$ (q), threshold for detecting downward price movements;
5. $filter_{boolean}S$, boolean indicator determining the direction of the trading signal.

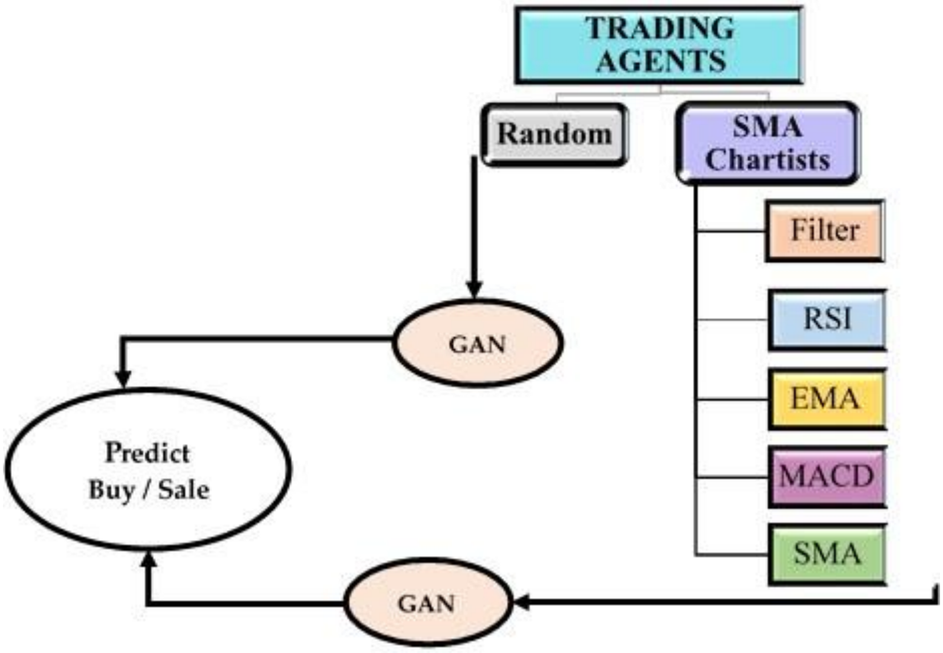

Figure 1. The overall proposed methodology

As defined by $filter_{flag}$ arguments is a boolean flag that turns the rule on or off represented by values between 0 and 1. The flag signifies whether the rule is considered during trading or not. The second parameter defines the initial value of $filter_{periods}$ specifying the historical period to be included in the analysis. The $filter_{increase}S$ and $filter_{decrease}S$ arguments are variables that represent the recent price action of the currency pair measured in pips a standard unit in foreign exchange trading that represents the smallest price movement of a currency quote.

The $filter_{boolean}S$ arguments is a Boolean variable with an equal probability of being 0 or 1. If $filter_{boolean}S$ = 0 then strategy generates buy indication if price moves up by more than p pips and a sell signal if the price moves down by more than q pips. Alternatively when value is 1 signals are inverted an increase by more than p pips creates a sell signal while a decrease by q pips creates a buy signal.

2. Relative Strength Index (RSI)

These steps are used to calculate RSI values:

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

To calculate RSI where n is number of periods and P is the last price of the period. The RSI price indicates buy and sell market orders. The RSI depends on a few parameters that indicate whether it is higher or lower than so called over bought and over sold indicators. The following features illustrate this principle:

6. rsi_{flag}; boolean variable that enables or disables the RSI-based trading rule;
7. rsi_n; number of periods used to compute the Relative Strength Index (RSI);
8. rsi_{os}; threshold indicating an over sold market condition;
9. rsi_{ob}; threshold indicating an over bought market condition;
10. $rsi_{boolean}$; boolean parameter that determines the interpretation of RSI signals for trade execution;

RSI_{flag} which is a boolean signal rsi_{flag} is turned on or off according to the rules. The rsi_{os} parameter number of periods to take into account when calculating average increases and reductions. The "oversold" and "overbought" signals are the eighth and ninth arguments respectively. The eighth and ninth arguments are constants and their values are provided by [5]. The $rsi_{boolean}$ argument is a Boolean variable that assumes a value of either 0 or 1 with equal probability and is used to determine the direction of the trading signal. When this variable equals 0 a buy signal is issued if the RSI falls below the rsi_{os} (oversold threshold) and a sell signal is generated if the RSI exceeds the rsi_{ob} (overbought threshold). Conversely, when the variable takes the value 1 logic is inverted: a sell signal is triggered if RSI is below rsi_{os} and buy signal is generated if is above rsi_{ob}.

3. Exponential Moving Average (EMA)

The Exponential Moving Average (EMA) is a type of weighted moving average that assigns greater significance to more recent price data. The EMA values are calculated as follows:

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

Where EMA(n) and close are the price of the indicator in the previous moment number of periods and closing price of period respectively. EMA has two main arguments that make up the rule

11. ema_{flag}, boolean variable used to activate or deactivate the exponential moving average (EMA) rule;

12. ema_n, defines period n used in the calculation of the EMA.

The ema_n, argument proves that for average calculation last n day closing prices need to be taken into account. If close price is higher than EMA(n) then this rule generates a buy signal whereas if the close price is lower than EMA(n) this rule generates a sell signal.

4. Moving Average Convergence Divergence (MACD)

Calculation of several moving averages results in the definition of the MACD indicator. It described as:

$$\mathrm{MACDt}\ (p, q, m) = \mathrm{MACDt}\ (p, q) - \mathrm{Signalt}\ (m) \quad (4)$$

where:

$$\mathrm{MACDline}_t\ (p, q) = \mathrm{EMA}_t\ (p, \mathrm{close}) - \mathrm{EMA}_t\ (q, \mathrm{Close}) \quad (5)$$

$$\mathrm{SignalLine}_t\ (m) = \mathrm{EMA}_t\ (m, \mathrm{MACD}_t\ (p, q)) \quad (6)$$

In this context p and q represent time spans for short term and long term exponential moving averages respectively while m denotes number of periods used to compute MACD line indicator. This term close refers to closing price of asset for given time retro.

MACD indicator employs following setting parameters:

13. macdflag: Turning on the MACD rule when 1 and off when 0.

14. macdperiodsS (as p): Defines the length of the short EMA period.

15. macdperiodsL (as q): Defines the length of the long EMA period which should be more than p.

16. macdperiodsN (as m): To mark the short-long EMAs difference period of the EMA.

17. macdbooleanS: A random binary variable with uniform probability and assumes a value of either 0 or 1.

$\mathrm{Macd_{boolean}S}$ parameter value-based trading signals depend upon value of $\mathrm{macd_{boolean}S}$ parameter. When value of the $\mathrm{macd_{boolean}S}$ parameter is 0 gives buy signal. When value of the $\mathrm{macd_{boolean}S}$ lies above the long moving average it gives sell signal otherwise. Conversely if value of $\mathrm{macd_{boolean}S}$ increases to one it generates sell indication where the gap between the long and short moving averages is not greater than signal value.

If a trading position is already active corresponding opening rule is disregarded. Additionally when all rule activation flags are set to zero or when multiple flags are simultaneously set to 1 one flag is randomly selected and assigned a value of one while the remaining flags are reset to 0. Each flag has an equal likelihood of being chosen during this randomization process.

5. Simple Moving Average (SMA)

In order to investigate best entry in market by using SMA indicator code on BINANCE data. The calculation of simple moving averages results in the definition of the SMA indicator. SMA is described as:

$$\mathrm{SMA_n} = \frac{1}{n}\sum_{i=0}^{n-1} y_{t-i} \quad (7)$$

Where n is window size i is total value minus window size plus one n total number of observed values and y_{t-i} is single observed value.

The parameters of the SMA rule are:

18. sma_{flag}, accepts 0 or 1 rule is switched on and off using values 0 and 1 respectively.

19. $\mathrm{sma_{periods}AVG}$, noted p.

20. $sma_{periods}L$, q is defined with the constraint that q > p, ensuring the long-term period exceeds the short-term period.

21. $sma_{periods}N$, the parameter m represents the moving average calculated from the difference between the short-term and long-term exponential moving averages.

22. $sma_{boolean}S$ takes either 0 or 1 with the same likelihood.

where y_{t-i} represents the asset's closing price at time $t-i$ and n is the number of periods over which the average is calculated. Within a trading context SMA generates a buy signal when the current price or a short-term SMA such as the 10-day crosses above a longer-term SMA (such as the 50-day). This crossover typically signals the emergence of a potential upward trend or bullish market sentiment. Conversely a sell signal is generated when the price or short-term SMA crosses below the long-term SMA suggesting the onset of a downward trend or bearish market condition. These crossover strategies help traders identify entry and exit points based on trend confirmation reducing the impact of short-term market noise.

3.2.2. Criteria for Closing a Trading Position

As already indicated after opening a spot traders look for the finest time to close it to minimize profit or loss. The same requirements that apply to starting a post also apply to leaving one.

6. Fixed Exit Levels (FEL)

This exit strategy is defined by the following three parameters:

23. fel_{flag}, boolean variable that enables or disables the FEL rule;

24. fel_{tp}, take-profit threshold used to evaluate when an open position should be closed to secure gains;

25. fel_{sl}, stop-loss threshold used to determine when an open position should be exited to limit losses;

If price is greater than or equal to fel_{tp} or less than or equal to fel_{sl} the FEL rule closes a long position and takes profit exits stop loss long position. If price is less than equal to or greater than fel_{tp} or less than fel_{sl} the rule exits the short position.

7. Trailing Exit Levels (TEL)

This rule is characterized by four parameters:

26. tel_{flag} is a Boolean variable that activates or deactivates the trailing exit rule.

27. tel_{tp} represents profit-taking threshold used to evaluate an open position should be closed to secure gains.

28. tel_{sl} denotes stop-loss threshold to assess whether position should be exited to prevent further losses.

26. teltl: A threshold value to be used for active position management and it must be smaller than teltp.

If any specific exit condition fails rule closes the position like the FEL rule but with a difference that teltp and teltl values are dynamically modified over time.

3.3. Orders and mechanism of price formation

The trading system outlined in studies [10], [11] is structured around a realistic limit order book mechanism. This order book organizes buy and sell orders into separate queues and matches them based on predefined criteria. Orders with identical limit prices are prioritized chronologically earlier orders are executed first while those with varying limit prices are ranked in descending order according to the limit price itself. The limit price reflects value at which a trader intends to execute a transaction and is influenced by both a random variable N_i (μ, σ_i) and the prevailing market price p(t). As demonstrated in [10], [11], the threshold prices associated with buy and sell orders denoted as P_b^l, i and P_s^l, i respectively are determined in relation to the current market price p(t), as shown below:

$$P_{b,i}^l(t) = p(t) * N_i(\mu, \sigma_i) \tag{8}$$

$$P_{s,i}^l(t) = \frac{p(t)}{N_i(\mu, \sigma_i)} \tag{9}$$

Where N_i (μ, σ_i^c) is a random variable with mean $\mu \simeq 1$ and a standard deviation of σ_i drawn from a Gaussian distribution. The price of Bitcoin at this time is p(t). It is directly related to the price p_T at which the transaction is completed the procedure for price formation p_T is given below:

- if $p_{s,\ i}^l > 0$ and $p_{s,\ i}^l = 0$
  then $p_T = \min(p_{s,\ i}^l, p(t))$;
- if $p_{s,\ i}^l > 0$ and $p_{b,\ i}^l = 0$

then $p_T = \max(p^l_{s,\ i}, p(t))$;

- if $p^l_{b,\ i} = 0$ and $p^l_{s,\ i} = 0$
  then $p_T = p(t)$;
- if $p^l_{b,\ i} > 0$ and $p^l_{s,\ i} > 0$
  then $p_T = \frac{p^l_{b,\ i} + p^l_{s,\ i}}{2}$

4. Genetic Algorithm

Genetic Algorithm (GA) begins by generating a population of potential solutions from an initial search space and iteratively refines these solutions to identify the most optimal one. This process involves the intentional introduction of variability such as mutation and crossover to explore new solution paths and promote convergence toward an optimal outcome.

In this research we employed the GA model as presented in [5]. In our simulation of a trading system context each solution is encoded in the form of a chromosome which is a list of technical analysis rules. Each of these rules has a number of defining parameters which are referred to as genes in GA.

The following is the algorithm that we implemented:

1) A random initial population of 100 candidate solutions (chromosomes) is created. Each chromosome is a unique set of technical trading rules acquired by randomly selecting parameter values from 100 different setups. In this model one chromosome is associated with one trading rule.A GA calculates each individual gain or loss during training to estimate them.
2) A GA calculates each individual gain or loss during training to estimate them.
3) A GA groups 100 people based on their gain or loss. By GA evaluates what represents a fitness function.
4) GA advances a new generation after passing through the following stages.

- GA introduces the highest profit earner in the new generation.
- Based on its profitability and therefore its ranking each of the 100 individuals may have a chance to become a parent in the new generation. The probability of becoming a parent will depend on income with the higher the income the higher the probability of becoming a parent. We separated people into 10 classes and gave each class a probability based on how well they performed in terms of profitability. We follow the steps outlined in work [5] which you may refer to for further information as well as the algorithm's calibration.
- GA selects random pairs of individuals from the population to generate new offspring based on predefined probabilities. The crossover process follows the strategy outlined in [5]. Specifically the GA randomly selects an integer n between one to tenty four and constructs new distinct by inheriting genes 1 through n from one parent and genes n+1 through 24 from the other. Each individual consists of 24 genes as detailed in Section 3.2. Using this method GA produces a total of 80 new individuals in each generation.
- To increase diversity the genetic algorithm randomly generates the remaining 19 individuals.

5) After the Genetic Algorithm produces 100 new individuals this collection is used as the next generation. Then the algorithm repeats from selection and reproduction (steps 2 and 3) then creating a new generation step 4.
6) Because making the iterations larger than a threshold does not provide significantly improved results the program is set to loop steps 2 through 5 up to 80 generations.

Table 1. Parameter search intervals explored by GA.

Subsamples Strictures	2014 – 2018 Range
2: filterperiods	[1, 15]
3: filterincreaseS	[0.02705, 0.079]
4: filterdecreaseS	[0.02705, 0.079]
8: rsin	[2, 10]
9: rsios	[15, 35]
10: rsiob	[65, 85]

12: eman	[2, 10]
14: macdperiodS	[5, 90]
15: macdperiodL	[10, 100]
16: macdperiodN	[5, 25]
19: smaperiodAVG	[7, 53]
24: feltp	[0.0074, 0.2]
25: felsl	[0.0025, 0.079]
27: teltp	[0.0074, 0.2]
28: telsl	[0.0025, 0.079]
29: teltl	[0.0025, 0.079]

The parameters of the rules are set according to the work [5]. The definition of parameters 3 ($filter_{Increase}S$), 4 ($filter_{Decrease}S$), 19 ($sma_{period}AVG$), 24 (fel_{tp}), 25 (fel_{sl}), 27 (tel_{tp}), 28 (tel_{sl}) and 29 (tel_{tl}) should be emphasized. These variables accept values from a certain range between the minimum value and the maximum value as shown in Table 1. Lowest and highest values are provided in [5]. A pip equal to 1 / 10000 dollar in the work just described is a very small unit of conversion for the BTC.USD currency pair.

In contrast to EUR/USD currency pair the BTC/USD pair exhibits significantly higher volatility and a more pronounced rate of change over time. The minimum and maximum thresholds were established based on observed returns and price variations. To align the probability distributions the initial values were defined by calculating collective distribution function (CDF) of daily proceeds from EUR/USD series. Given that simulation operates on a daily time step returns were computed using one-day intervals.

To ensure consistency with the methodology outlined in [5] the cumulative return function for the BTC/USD series was also derived using daily return data. The probability of observing the specified minimum or maximum values can be estimated from this function. Accordingly values corresponding to the lowest and highest probabilities were obtained by evaluating the CDF at the minimum and maximum return thresholds respectively.

After every 80 cycles of the GA iteration it selects a set of guidelines with initial statute and exit statute both of which have absolute values for their parameter values. The settings of these factors provide the optimum level of profit. The algorithm is executed 100 times producing 100 distinct sets of trading rules. From this pool of rule sets chartist agents select a specific set to guide their trading behavior within the proposed synthetic market model.

Every rule set generated by the GA is an automated trader that will automatically enter and exit buy and sell orders in response to signals produced by the entry rule. A trader will enter a position when the conditions of the entry rule are satisfied and will only close the position once the initial order is fully filled and the exit rule produces a take-profit or stop-loss condition.

It is important to emphasize that throughout both the training and testing phases each buy or sell order corresponds to the purchase or sale of exactly one Bitcoin. For every buy order placed a corresponding sell order is automatically generated by the system and vice versa. All orders are submitted with the requirement that they be fully executed. These two simplifying assumptions serve to streamline the model allowing for rapid computation of individual trader profits. Moreover they ensure that all transactions are promptly executed within the designated training period thereby maintaining consistency and preventing the accumulation of unmatched orders in the order book.

The GA's performance was evaluated using closing prices from the BTC/USD currency pair on a daily basis across five years January 1, 2014 to December 31, 2018 for a total of 1826 daily observations. To simulate a realistic backtesting scenario and avoid overfitting the dataset was divided into two equal halves each spanning 2.5 years. The first half January 1, 2014 to June 30, 2016 was used as the training period during which the GA identified the optimal sets of trading rules. These rule sets were then evaluated during the testing period July 1, 2016 to December 31, 2018 enabling an unbiased assessment of their generalizability and performance on unseen data.

To further explore how market dynamics influence strategy effectiveness we segmented the full sample into three distinct subsamples each representing a different market regime:

- Subsample 1: January 1, 2014 – December 31, 2016 low volatility early adoption phase.
- Subsample 2: January 1, 2017 – December 31, 2017 high volatility speculative boom.
- Subsample 3: January 1, 2018 – December 31, 2018 market correction phase.

Descriptive statistics reveal the evolution of the market: Subsample 1 had a mean price of $455, SD = $179 Subsample 2 averaged $3970, SD = $4022 and Subsample 3 showed a mean of $7601, SD = $2470. Minimum prices across these subsamples were $177, $775 and $3236 respectively while maximum prices were $975, $19475 and $17527. Each subsample was split into equal training and testing windows to evaluate the GA's robustness across varying market conditions.

The GA was run separately on each subsample allowing us to extract three optimized rule sets (100 rules per set). Interestingly results indicated that rule sets derived from subsampled training data underperformed compared to those developed using the full 5-year dataset suggesting that broader market exposure leads to more generalized and effective trading strategies.

Moreover when the projected model was executed under the statement of chartist agents select one of three available rule sets for trading the outcomes were found to be comparable to those obtained when using a single optimal rule set identified by the GA based on the evaluation of the entire time series.

5. Results and Discussion

In this section we begin with a description of genetic algorithm (GA) results and end with an example of model simulation results.

5.1. Genetic Algorithm Results

We implement GA described in section 4 before we start the model simulation determine best set of rules and best values for resulting set's parameter values we constructed and ran this procedure using the SmallTalk programming language.

Genetic Algorithm finds out parameter settings that will maximize labor income [5]. To find characteristics of the top 100 sets of rules it is executed 100 times for training process and then test performance of 100 sets on the out-of-sample series of test period.

It is important to note that certain parameters may produce values outside the specified ranges in Table 1. This occurs because the parameters are dynamically updated throughout the simulation as detailed in Section 3.2.

During the training phase the 100 traders who adhered to the optimal rule sets identified by the GA achieved an average profit of $4531.5 mean per trader: $63.2. In the testing phase their performance improved substantially yielding an average total profit of $82374.8 mean per trader: $1456.2

Evaluate robustness of these results same trading strategy was applied to 100 randomly selected individuals each subjected to six experimental conditions outlined in Section 3.2 with parameters randomly assigned. As expected these randomly configured traders generated lower average profits. Specifically they earned $2,271.51 during training phase with standard deviation of $1177.97 and $43679.1 during the testing phase with a standard deviation of $29564.9. Notably standard deviation was significantly smaller for traders following the GA optimized rule sets indicating more consistent and reliable performance compared to those using randomly selected strategies.

5.2. Model Results

Between January 1, 2014 and December 31, 2018 proposed model was implemented in SmallTalk programming language. Thus 1459 steps or one day was chosen as the simulation time length. We test the flexibility of the model and the accuracy of our statistical analysis by running it 100 times under the same initial conditions but with different random number generator seeds. Unless otherwise stated in section 3. We used the values found in the work [11] for the variables when calibrating the model.

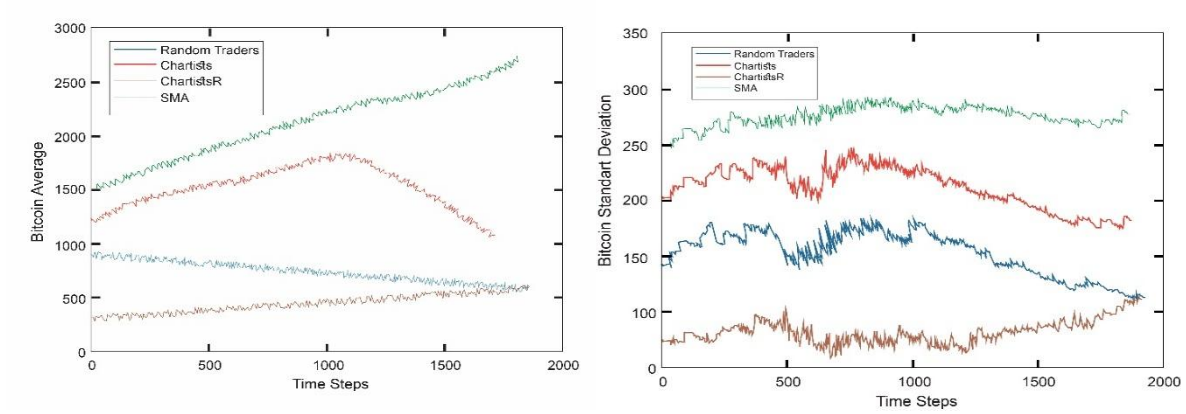

Figure 2. The Avg and Std of cryptocurrency holdings were calculated across all Monte Carlo simulations conducted Base Run.

Table 2. Descriptive statistics for the 100 best-performing sets.

Parameters	2014 – 2018 Avg (Std)
2: filterperiods	7.07 (2.78)
3: filterincreaseS	0.034 (0.013)
4: filterdecreaseS	0.07 (0.019)
8: rsin	7.03 (1.07)
9: rsios	42.54 (2.43)
10: rsiob	54.23 (4.87)
12: eman	4.25 (3.89)
14: macdperiodS	53.78 (14.98)
15: macdperiodL	64.74 (21.19)
16: macdperiodN	32.45 (4.89)
19: smaperiodAVG	21.31(17.13)
24: feltp	0.098 (0.052)
25: felsl	0.045 (0.019)
27: teltp	0.215 (0.039)
28: telsl	0.031 (0.021)
29: teltl	0.031 (0.024)

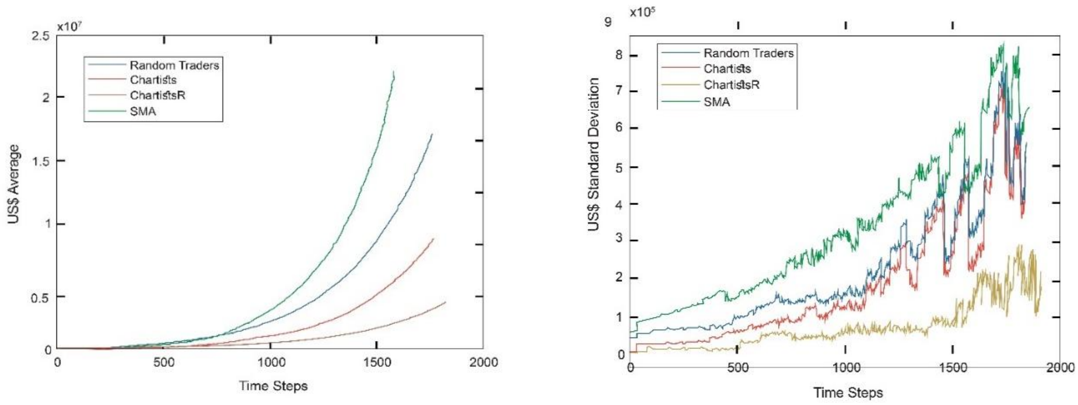

Figure 3. The Avg and Std of fiat currency holdings were calculated across all Monte Carlo simulations conducted Base Run.

In accompanying legend label Chartists corresponds to C_{bR} representing the segment of chartist traders employing the best-performing rules. The label ChartistsR also refers to C_{bR} indicating chartists using random rule sets while Random Traders denotes agents who engage in trading without relying on technical strategies.

5.2.1. Trader Statistics, Simulated Bitcoin Prices and Bitcoin Trading Volume

The average total wealth of traders defined as the sum of fiat currency and cryptocurrency holdings with the latter multiplied by the current bitcoin price is used as a key performance indicator in this analysis. To isolate profits earned strictly through trading activity initial fiat and cryptocurrency holdings of each trader at their entry time tE are excluded from these calculations. The outcomes from the Base Run and population level statistics on fiat and crypto assets are presented in the following section.

Figures 2 and 3 display the results for two trader groups: the Chartists divided into CbR who follow GA optimized rules and CrR who use randomly selected rules and Random Traders. These figures report the mean and standard deviation of both fiat and cryptocurrency holdings per individual based on multiple Monte Carlo simulations.

To explore the sensitivity of the model additional simulations were conducted by varying two key parameters: the proportion of chartists using best rules percCbR and the probability of submitting market orders. These variations allow a deeper investigation into the differences between the trading dynamics of the proposed synthetic model and those governed purely by the GA.

The first parameter percCbR reflects the diversity in trading strategies among participants in proposed model while the second captures the mix between market and limit orders. In contrast GA driven trading model involves only a limited number of chartists uses fixed order quantities and executes all trades through market orders automatically pairing buy and sell orders for immediate execution.

Simulation results show no consistent pattern in how variations in percCbR influence performance. However across all scenarios chartists following the best rules CbR consistently outperform both random chartists CrR and random traders in terms of profitability.

As shown in Figure 2 CbR group accumulates more cryptocurrency holdings over time due to superior trading strategies and built-in model advantages. Figure 4 further illustrates that the average total wealth per trader is highest for the CbR population confirming their greater effectiveness and profitability compared to other trader types.

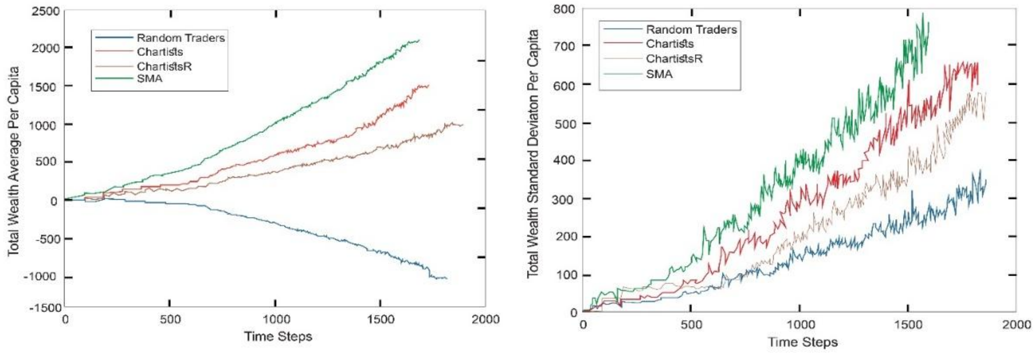

Figure 4. Average and standard deviation of the total wealth per capita in the *Base Run* across all Monte Carlo simulations.

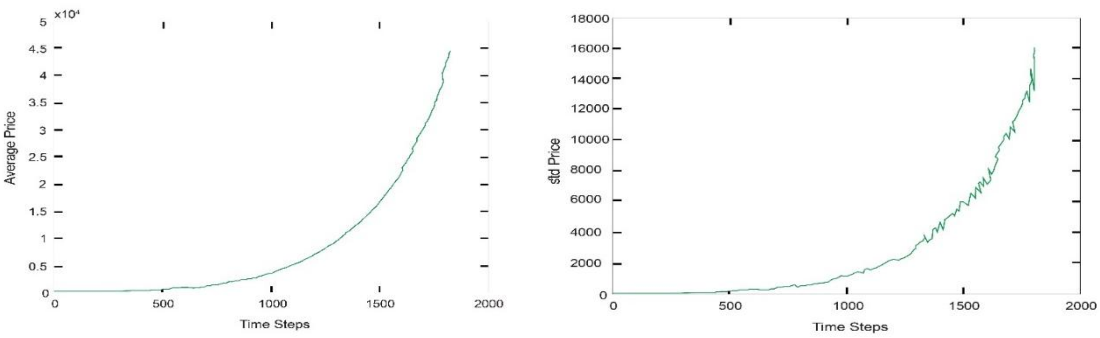

Figure 5. The *Base Run's* simulated bitcoin price avg and std for all monte carlo runs.

In model of simulation bitcoins freshly mined get distributed among traders according to the size of their current holdings of cryptocurrency. This leads to the C_{bR} group Chartists employing GA optimized rules increasingly holding a greater proportion of mined bitcoins at cost of C_{rR} Chartists employing random rules and Random Traders. As indicated by Figure 2 crypto-cash of C_{bR} increases consistently over time overtaking that of C_{rR} and particularly Random Traders whose holdings of cryptocurrency are in decreasing trend.

The decreasing crypto holdings of Random Traders are mainly as a result of their trading activities they issue more buy orders than sell orders and their sell orders are generally smaller in quantity. This provides a mismatch in the order book resulting in an excess of unexecuted buy orders over sell orders. As a result most buy orders from Random Traders go unexecuted adding to increasing fiat balance but decreasing average net worth per capita as their crypto positions do not increase efficiently.

Figure 5 shows the mean and standard deviation of the simulated Bitcoin price through time. In contrast to actual trends where Bitcoin plunges dramatically after peaks the virtual model features a smooth upward trend in average prices. The trend is shaped by the nature of the model as noted in [10, 11] where there is a mechanism that enables the model to mimic dramatic peaks and corrections in prices. Yet the mean in simulations still drifts upward over time since it's endogenously determined based on the proportion of total circulating fiat money to total circulating bitcoin. As new entrants to the market bring in more fiat money the price rises commensurately.

In Base Run bitcoin price is assumed to be an endogenous variable that is determined by the process outlined in Section 3.3 which applies the more general market model of Section 3. A further set of simulations was also performed where the bitcoin price was assumed constant and was an exogenous variable being fixed equal to the current market value of one bitcoin. In both instances we analyzed the combined wealth per trader for all trader types. Once we had scaled the simulated trading volume by a factor of 2500 to approximate true market conditions model accurately recreated the actual volume of bitcoin transactions.

Simulation performance shows that CbR traders reliably make greater profits than Random Traders although in certain instances they don't outperform CrR traders. Two interrelated factors account for this result:

The order book structure employed in the GA-based trading system is different from that employed in the synthetic model proposed here with implications for trade profitability and execution. There are discrepancies between real (exogenous) and simulated (endogenous) bitcoin price movements. The endogenous prices in the model slowly rise whereas actual bitcoin prices fluctuate more.

Yet under a scenario where no limit is placed on the initial space for the GA i.e. where the GA is not limited to explore freely outside the optimized initial range of Section 4 the CbR group performs better than CrR even when the simulation is based on exogenous prices. In the case where the model is based on endogenous pricing the GA limits its initial solutions within the specified optimal range leading to the persistent dominance of CbR strategies.

Table 3. Percentile values of the 1 statistic for all monte carlo simulations under the null hypothesis of a random walk without drift. In brackets are statistics for price logarithm series.

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

5.2.2. Stylized Characteristics of the Real and Simulated Bitcoin Markets

We looked at Bitcoin price history between January 1, 2014 and December 31, 2018 both in real time and simulated. We specifically tested and investigated various stylized data from real and virtual BTC time series.

Regarding the stylized facts that were analyzed they correspond to the facts in the publications [10], [11]. We investigated the phenomena of fat tails volatility clustering and unit root characteristic [4], [25] and [26].

The Augmented Dickey-Fuller test was used for the real and simulated series of daily Bitcoin prices as well as the real and simulated series of logarithms of the daily Bitcoin price that we investigated to test for the unit root property. flow under the null hypothesis of a random walk to be studied. Since the data of 1 and its percentiles for the original price series also known as the real price logarithm series and the simulated price series also known as the virtual price logarithm series respectively each surface there are always more relative than important. values we are unable to reject the null hypothesis for any series at the 1, 5, or 10% level. With 1826 observations significant values are -2.38, -1.92 and -1.41 at 2, 4 and 12% levels individually.

1 figure are -1.42 for the tangible value sequence -1.12 for the real value logarithm series respectively. Table 3 details the percentages of the 1 statistic of the simulation series over all Monte Carlo iterations.

The thick tail phenomenon is another investigated feature. The pronounced leptokurtosis observed in both empirical and simulated return distributions confirms the presence of fat tails. The presence of fat tails in the return distribution is evidenced by significant leptokurtosis observed in both empirical and simulated datasets. Specifically kurtosis values for actual returns and total returns are calculated as 7.64 and 11.52 respectively indicating a higher probability of extreme market movements compared to a normal distribution.

Table 4. All Monte Carlo simulations centile values for the Kurtosis value of the price returns and the price absolute returns are shown (in parentheses).

Centile Value			
.25	**.50**	**.75**	**.975**
21 (22)	25 (34)	46 (57)	382 (480)

Table 5. Avg_{Retraw} and std_{Retraw} average and standard deviation of the autocorrelation of raw returns, and avg_{Retabs} and std_{Retabs} the average and standard deviation of the autocorrelation of absolute returns throughout all monte carlo simulations.

Eloquent Statistics	**Centile Value**			
	.25	**.50**	**.75**	**.975**
$avgRet_{raw}$	.009	.008	.02	.012
$avgRet_{abs}$	.05	.07	.2	.2
$stdRet_{raw}$	.02	.01	.031	.03
$stdRet_{abs}$	.02	.02	.031	.04

As a result return distribution is more prone to outliers than the normal distribution. The .25th, .50th, .75th, and .97.5th centiles of the kurtosis price returns over all MC runs are shown in Table 4. Compared to the real scenario the synthetic kurtosis is slightly larger. For price returns median is equal to 28 which is close to the original number.

The third factor under investigation is volatility clustering. Table 5 shows the mean and standard deviation of the autocorrelation of simulated raw returns and absolute returns for time lags 1 to 20 along with the .25th, .50th, .75th, and .97.5th percentiles for each.

The results presented in Tables 3 and Table 4 provide statistical validation of the model's ability to replicate key stylized facts observed in real-world financial markets. Specifically Table 3 highlights the distribution of the l statistic across Monte Carlo simulations under the assumption of a random walk without drift. The corresponding values for the price logarithm series fall well within these percentiles indicating that the model preserves the non-trivial statistical properties of actual BTC/USD price movements including stationarity and moderate deviation from pure randomness.

Furthermore Table 4 reports the centile values of kurtosis for both price and absolute returns. The presence of high kurtosis values particularly at the 0.975 percentile 377 for returns and 460 for absolute returns strongly supports the presence of fat tails in the return distribution. This aligns with the empirical evidence from real Bitcoin markets where extreme price changes occur more frequently than predicted by normal distributions. Such results also indicate volatility clustering a hallmark of financial time series where large changes tend to be followed by large changes of either sign and small changes by small changes.

6. Conclusion

To evaluate and simulate the trading of the BTC/USD currency pair over the period from January 1, 2014 to December 31, 2018 we developed a diverse agent based model of the BTC market. The proposed framework integrates a realistic order book-based price-clearing mechanism and incorporates two distinct types of market participants: chartists and random traders. Chartists apply trading rules derived from two sources one set optimized using a genetic algorithm to maximize profits during the training phase and another generated by selecting rule parameters randomly. Random traders by contrast submit buy or sell orders without following any trading strategy.

The model successfully reproduces several key stylized facts observed in real-world Bitcoin markets including the presence of unit roots in price series fat-tailed return distributions, volatility clustering and realistic trading volumes. While the model does not claim perfect accuracy simulation outputs represent a reasonable approximation of actual market behavior and demonstrate moderate improvements over existing models. Notably trading rules optimized by the genetic algorithm consistently outperformed randomly chosen strategies in both the training and testing phases indicating robustness and generalizability. A notable enhancement in this study is the incorporation of the Simple Moving Average (SMA) as an additional technical indicator within the genetic algorithm’s rule set. This inclusion expands the diversity of strategies that chartists can adopt and contributes to the model's ability to capture market dynamics more effectively.

To the best of our knowledge this work represents one of the first applications of agent-based modeling combined with realistic trading mechanisms to simulate BTC/USD market dynamics. For computational simplicity this initial model considers unit-sized buy and sell orders and generates matched counter-orders automatically. In future work we aim to incorporate more complex agent behavior, variable order sizes and advanced evolutionary optimization techniques to enhance the realism and scalability of the simulated trading environment.

Funding: This research received no external funding.
Data Availability Statement: Data will be made available on request.
Acknowledgments: Not Applicable.
Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Abounoori, E., Shahrazi, M., & Rasekhi, S. (2012). An investigation of Forex market efficiency based on detrended fluctuation analysis: A case study for Iran. *Physica A: Statistical Mechanics and its Applications*, *391*(11), 3170-3179.
2. Alvarez-Ramirez, J., Rodriguez, E., & Espinosa-Paredes, G. (2012). Is the US stock market becoming weakly efficient over time? Evidence from 80-year-long data. Physica A: Statistical Mechanics and its Applications, 391(22), 5643-5647.
3. Seetharaman, A., Saravanan, A. S., Patwa, N., & Mehta, J. (2017). Impact of Bitcoin as a world currency. Accounting and Finance Research, 6(2), 230-246.
4. Begušić, S., Kostanjčar, Z., Stanley, H. E., & Podobnik, B. (2018). Scaling properties of extreme price fluctuations in Bitcoin markets. Physica A: Statistical Mechanics and its Applications, 510, 400-406.
5. Boboc, I. A., & Dinică, M. C. (2013). An algorithm for testing the efficient market hypothesis. PloS one, 8(10), e78177.
6. Chakraborti, A., Toke, I. M., Patriarca, M., & Abergel, F. (2011). Econophysics review: II. Agent-based models. Quantitative Finance, 11(7), 1013-1041.
7. Chaparro, F. (2017). Some of the Biggest Crypto Exchanges are Shutting Out New Users Because They Can't Keep up With Demand. Mich. Tech. L. Rev., 25, 1.
8. Chen, S. H., Chang, C. L., & Du, Y. R. (2012). Agent-based economic models and econometrics. The Knowledge Engineering Review, 27(2), 187-219.
9. Chiarella, C., Iori, G., & Perelló, J. (2009). The impact of heterogeneous trading rules on the limit order book and order flows. Journal of Economic Dynamics and Control, 33(3), 525-537.
10. Cocco, L., Concas, G., & Marchesi, M. (2017). Using an artificial financial market for studying a cryptocurrency market. Journal of Economic Interaction and Coordination, 12, 345-365.
11. Cocco, L., & Marchesi, M. (2016). Modeling and Simulation of the Economics of Mining in the Bitcoin Market. PloS one, 11(10), e0164603.
12. Dempster, M. A. H., & Jones, C. M. (2001). A real-time adaptive trading system using genetic programming. Quantitative Finance, 1(4), 397.
13. Detzel, A. L., Liu, H., Strauss, J., Zhou, G., & Zhu, Y. (2019). Bitcoin: Learning and predictability via technical analysis. In Paris December 2018 Finance Meeting EUROFIDAI-AFFI (Vol. 1, p. 314).
14. Fama, E. F. (1970). Efficient capital markets. Journal of finance, 25(2), 383-417.
15. Garcia, D., & Schweitzer, F. (2015). Social signals and algorithmic trading of Bitcoin. Royal Society open science, 2(9), 150288.
16. Akhundjanov, S. B., & Toda, A. A. (2020). Is Gibrat's "economic inequality" lognormal?. Empirical Economics, 59, 2071-2091.
17. Hasan, M. Z., Kamil, A. A., Mustafa, A., & Baten, M. A. (2012). Stochastic frontier model approach for measuring stock market efficiency with different distributions. PloS one, 7(5), e37047.
18. Holland, J. H. (1962). Outline for a logical theory of adaptive systems. Journal of the ACM (JACM), 9(3), 297-314.
19. Indera, N. I., Yassin, I. M., Zabidi, A., & Rizman, Z. I. (2017). Non-linear autoregressive with exogeneous input (NARX) Bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators. Journal of fundamental and applied sciences, 9(3S), 791-808.
20. Chen, S. H., Kaboudan, M., & Du, Y. R. (Eds.). (2018). The Oxford handbook of computational economics and finance. Oxford University Press. vol. 2, pp. 27–32.
21. Jang, H., & Lee, J. (2017). An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE access, 6, 5427-5437.
22. Kim, J. H., Shamsuddin, A., & Lim, K. P. (2011). Stock return predictability and the adaptive markets hypothesis: Evidence from century-long US data. Journal of empirical finance, 18(5), 868-879.
23. Levy, M., & Solomon, S. (2017). New evidence for the power-law distribution of wealth. Physica A: Statistical Mechanics and its Applications, 242(1-2), 90-94.
24. LiCalzi, M., & Pellizzari, P. (2003). Fundamentalists clashing over thebook: a study of order-driven stock markets. Quantitative finance, 3(6), 470.
25. Lux, T., & Marchesi, M. (2000). Volatility clustering in financial markets: a microsimulation of interacting agents. International journal of theoretical and applied finance, 3(04), 675-702.
26. Pagan, A. (2018). The econometrics of financial markets. Journal of empirical finance, 3(1), 15-102.
27. Tedeschi, G., Iori, G., & Gallegati, M. (2012). Herding effects in order driven markets: The rise and fall of gurus. Journal of Economic Behavior & Organization, 81(1), 82-96.

28. Williams-Grut, O. (2017). The cryptocurrency market is now doing the same daily volume as the New York Stock Exchange. Business Insider UK.
29. Yule, G. U. (2022). II.—A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FR S. Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character, 213(402-410), 21-87.