Bitcoin Algorithm Trading using Genetic Programming

Monira Essa Aloud
Department of Management Information system, College of Business Administration, King Saud University

mealoud@ksu.edu.sa

Abstract

The author presents a simple data-driven intraday technical indicator trading approach based on Genetic Programming (GP) for return forecasting in the Bitcoin market. We use five trend-following technical indicators as input to GP for developing trading rules. Using data on daily Bitcoin historical prices from January 2017 to February 2020, our principal results show that the combination of technical analysis indicators and Artificial Intelligence (AI) techniques, primarily GP, is a potential forecasting tool for Bitcoin prices, even outperforming the buy-and-hold strategy. Sensitivity analysis is employed to adjust the number and values of variables, activation functions, and fitness functions of the GP-based system to verify our approach's robustness.

Keywords:

Bitcoin market; Artificial Intelligence; Genetic Programming; Technical analysis; Trading rules.

1 INTRODUCTION

Cryptocurrencies are now rapidly growing in use as a new type of financial tools. The Bitcoin market, the earliest decentralized cryptocurrency, pioneered in [24], has matured to a size of approximately 170 billion US dollars. Bitcoin has emerged as a potential investment opportunity [8, 14]. As a financial asset, Bitcoin has elements both of a commodity and a currency.

The efficient market hypothesis (EMH) provides a vital theoretical framework in the financial-economic domain [26]. Specifically, the EMH states that if the financial market is informationally efficient (i.e., prices fully reflect the available information that relates to the financial asset being traded), a profitable investment trading strategy (i.e., a buy-and-hold trading strategy) is not required to outperform the market taking into consideration the execution costs of the transaction. Therefore, it is not feasible to make price predictions by using the available information in that market.

The absence of evaluation methods for Bitcoin investments has driven the exploration and examination of the potential forecasting of different variables such as Bitcoin popularity [13], trading volume [4], stock market ambiguity [8], and other cryptocurrencies [9]. Bitcoin prices appear to exhibit a forecasted pattern [27], resulting from Bitcoin's short history in the financial market and the irrational trading behaviours of different orders in the Bitcoin market [9]. In conjunction with Bitcoin market growth, researchers have also explored the cryptocurrency investment domain, even though only a small number of papers have been published in that field [10, 12, 16, 19].

The price dynamics in the Bitcoin time series have been shown to exhibit some unique features. [3, 5, 6, 23] study the Bitcoin market's statistical properties. One of the most significant results they note is the Hurst exponent's inefficiency of the time series of returns. The study reported in [4] shows that the Bitcoin price returns exhibit heavier tails and are more volatile than other financial stocks. This tendency has been identified through the detection of power-law behavior and the study of the scaling laws and exponents. Furthermore, [10] shows that the Bitcoin market exhibits high price return and high volatility.

Several Artificial Intelligence (AI) and machine-learning methods have been employed effectively for price return forecasting. Among the most popular and effective methods have been artificial neural networks, genetic algorithms, and genetic programming (GP) [1, 2, 7, 18, 22, 23, 29]. In this study, we use GP to search for technical analysis indicators, as an alternative to applying pre-specified trading rules. AI techniques learn and identify patterns from a set of data to forecast price movements and thereby assist investors in their decision process. Several studies in the financial forecasting literature have used technical analysis indicators as inputs to form their predictors [11, 29].

Nevertheless, one limitation of the various studies on financial forecasting based on AI is that the technical analysis indicators they use as inputs have a pre-specified period. For illustration, for the moving average technical analysis indicator, they use a 12-day moving average time window. However, no one can assure that "12 days" is optimal for a specified period's technical indicator. As a result, in this study, we consider it vital to investigate the effects of evolving trading rules on GP to decide on the optimal indicators.

The contribution of this study is to apply AI techniques for algorithmic trading in the Bitcoin market. This study

Manuscript received July 5, 2023
https://doi.org/10.22937/IJCSNS.2023.23.7.24

investigates whether technical trading rules that evolve using GP can overtake a buy-and-hold (B&H) trading strategy in the Bitcoin market. We use five technical indicators and evaluate their trading performance by comparing them with a B&H strategy based on the return on investment (ROI) and Sharpe ratio calculated using the bootstrapping method. Our study offers Bitcoin traders a potential strategic trading method based on AI and machine-learning techniques, thereby showing AI's potential in evolving arbitrage trading rules and profitable investing decisions. Experimental results were promising, showing improved performance of the evolving trading rules using GP, although results were not always consistent across all datasets tested. This is the first study to employ a GP-based forecasting tool to Bitcoin trading.

The paper is structured in four sections. Section 2 defines the dataset and forecasting method. Section 3 defines the GP-based forecasting system. Section 4 offers the results. Section 5 concludes the paper.

2 Data and Method

This section describes how we use AI techniques, specifically GP, and technical analysis indicators, to develop trading rules to forecast Bitcoin prices. Section 2.1 presents the historical price data. Section 2.2 illustrates and defines the technical analysis indicators used in this study. Section 2.3 describes the GP core working mechanism.

2.1 Dataset

We use the daily price and volume Bitcoin historical data from Yahoo Finance. We used the following three cryptocurrency pairs: BTC-USD, ETH-USD, and BCH-USD. The time period for BTC-USD and ETH-USD is from January 1, 2017 to March 1, 2020; while for BCH-USD is from July 23, 2017, to March 1, 2020. For the price data, we use the following for each trading day: opening po, closing pc, high ph, and low pl prices.

In our study, we use the price return data rt, which is calculated from the closing prices in a daily time-series, as follows:

$$R_t = \frac{p_t}{p_{t-1} - 1} - 1 \qquad (1)$$

where t is a time index in the time series data. The data record for each Bitcoin are illustrated in Figure 1.

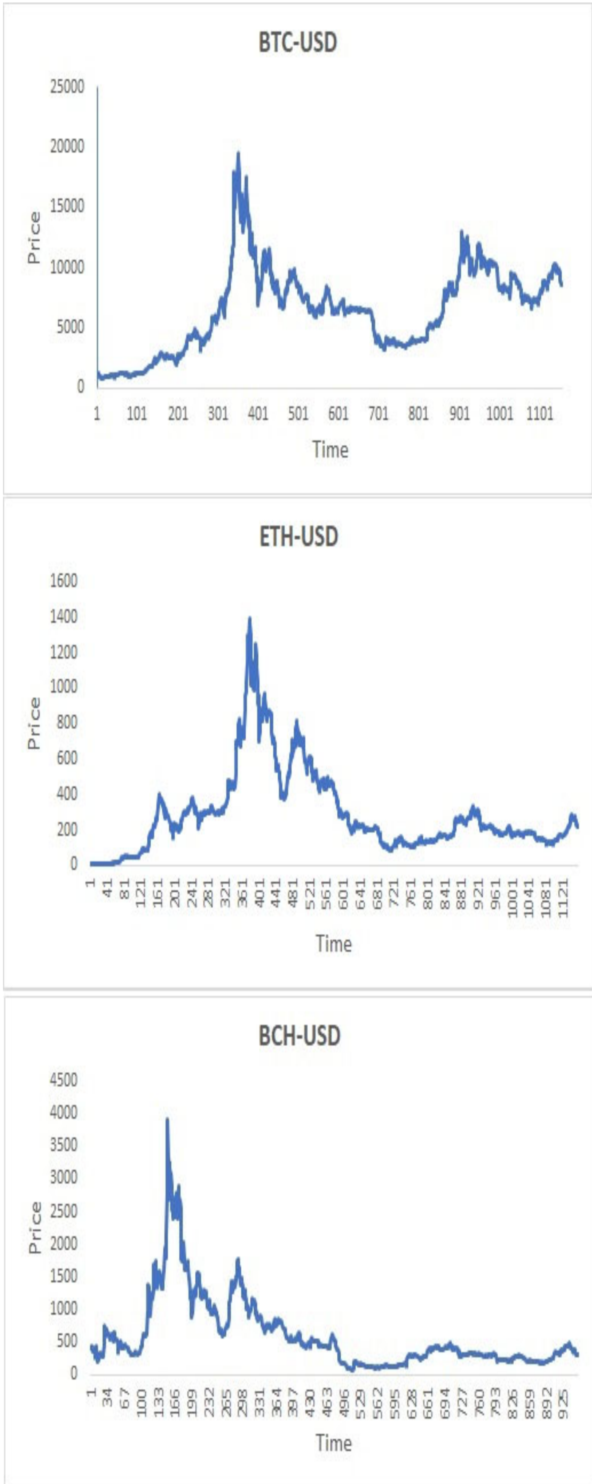


Figure 1. Daily closing price for the following cryptocurrency pairs: BTC-USD, ETH-USD, and BCH-USD.

2.2 *Technical Indicator*

In this section, we describe the five technical analysis indicators along with their formulas. These technical indicators have been used to generate trading rules using the training datasets.

The first rule we used is the price moving average (MA), which the financial literature considers the most popular technical rule [11]. The MA trading rules imply a signal for a buy (sell) order once the current price, or latest average price, go above (is less) a longer-term price average of the specified stock asset. In this study, we use the 10-day, 15-day, and 30-day price averages. The MA formula is as follows:

$$MA(L,t) = \frac{p(t) - \left(\frac{1}{L}\sum_{i=1}^{L} p(t-i)\right)}{\frac{1}{L}\sum_{i=1}^{L} p(t-i)} \quad (2)$$

A second technical trading rule among the most widely used financial indicators [11] is the trading range breakout (TRB). The TRB is also well known for representing the support and resistance levels. The TRB issues minimum and maximum prices, correspondingly, for which a financial asset has transacted throughout the preceding *n* days. In this study, for *n* days, we use 50, 150, and 200 days, as [11] used. The TRB indicator is calculated as follows:

$$TRB(L,t) = \frac{p(t) - max\{p(t-1), \dots, p(t-L)\}}{max\{p(t-1), \dots, p(t-L)\}} \quad (3)$$

The third technical trading rule used is the rate of change (ROC). The ROC rule correlated the current price to the price *n* days in the past. A standard time period used in the financial literature for ROC is ten trading days, which we used in this study. The ROC is calculated as follows:

$$ROC = \left(\frac{p_c - p_{c-n}}{p_{c-n}}\right) * 100 \quad (4)$$

where p_c is the closing price of the most recent time period, while p_{c-n} is the closing price for the previous *n* periods.

The fourth technical trading rule used is the on-balance volume (OBV) technical momentum trading indicator, which is considered the top-known technical indicator established on trading volume to make price predictions. The OBV indicator uses the volume flow of an asset to predict changes in the price time series. The OBV formula is as follows:

$$OBV = f(x) = \begin{cases} volume, if\ close > close_{prev} \\ 0, if\ close = clsoe_{prev} \\ -volume, if\ close < close_{prev} \end{cases} \quad (5)$$

We use the relative strength index (RSI) for the fifth technical trading indicator rule, which measure the adjustment and rate of price movements. The authors in [28] state that the RSI is the highly employed countertrend technical indicator. The RSI fluctuates between zero and 100. An RSI value greater than 70 typically indicates that Bitcoin has increased in value but is now overbought (i.e., one should sell) and oversold when below 30. These two traditional levels can also be modified if required to improve the fit of the asset price time series. The RSI is calculated using the following formula:

$$RSI = 100 - \left[\frac{100}{1 + \frac{Average\ gain}{Average\ loss}}\right] \quad (6)$$

2.3 *GP*

GP, introduced in [20], is an extension of the GAs pioneered by Holland [15]. The principal difference between GAs and GP is in their representation structures for potential solutions to a problem. In GAs, the individual members of the population are represented by fixed-size strings that represent candidate solutions. In contrast, in GP they are structured in programs that offer the applicant several solutions.

GP routinely generates decision trees that are executable by computation and have various structures and length representations. Decision trees are developed by mixing different functions and terminals. The function set comprises the basics employed to structure the models of a decision tree and create the branches. These can be relational and arithmetic operators, domain-specific functions, Boolean values, or operator pre-specified functions. The terminal set comprises the variables and constants that lie in the closing nodes of a branch on the tree. These terminal values are the independent variables of the GP space search, or as temporary constants produced as random constant values generated in the initialization population of the decision tree. In addition, terminals can operate as the parameters to the defined functions. As a way of illustration, a classic technical trading rule in financial markets, "buy order once the asset's price develops greater than the MA of the asset's price for the previous 10 days," is exemplified by a tree structure demonstrated in Figure 2. This tree structure can be written in infix notation as follows: "IF asset_price > MA (price, 10) THEN buy."

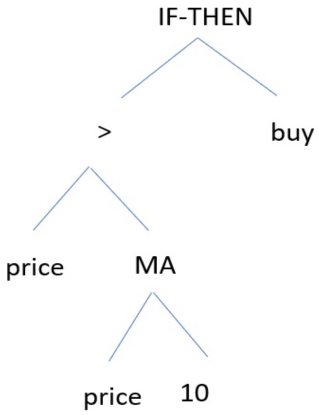


Figure 2. Decision-tree representation of a trading rule.

The basic GP steps for creating candidate solutions to a problem are (Poli et al. 2008):

(1) Create an initial population pop_i of n trading rules (programs) from the available data randomly.

(2) Repeat for each population pop_i:

(2.1) Execute each trading rule (program) using a training dataset and calculate its fitness value using the defined fitness function.

(2.2) Select one or two trading rule(s) (program(s)) as of the population pop_i where a proportionate fitness probability of participating in the genetic operations (crossover, mutation, and reproduction).

(2.3) Create new individual trading rule(s) by employing genetic operators based on the defined probabilities.

(3) Repeat the previous steps until a satisfactory solution is realized or a specific other termination condition is satisfied (e.g., a maximum number of generations is reached).

(4) Return the best trading rule (program) with the highest fitness value.

The Fitness function measures the GP performance of candidate solutions through the evolution process. For each generation, parents (candidate solutions) are selected with regard to their fitness value to participate in the creation of a next-generation, with the result that parents with high fitness value get a greater possibility of replicating in the next generation. The crossover operator combines the content of two parents to create an offspring solution. On the other hand, Mutation is the second genetic operator used in both GAs and GP to create slight genetic variations of the solution for a minor fraction of the population. Reproduction is employed as the third genetic operator to avoid missing the most refined individuals. Therefore, the reproduction operator entails selecting various individuals in a population-based on their fitness value. Afterwards, the addition of a version of these individuals interested in the next generation without adjustments. In this study, the finest observed individuals are replicated to the new generation based on a fixed probability threshold value.

3 GP-Based System

The approach that the proposed GP-based system in this paper, and most studies of GP-based systems, have used in financial forecasting works as follows. The operator starts by providing a set of: historical price data, technical analysis indicators (described in section 2.2), and decision signals (buy, sell, and hold); the GP-based system afterwards uses the historical data and, over a GP learning process, generates and evolves a population of genetic decision trees (GDTs), that provide a decision support framework by offering recommendation signals to buy (denoted by 1), sell (denoted by -1), or hold (not-to-buy) (represented by 0). Figure 3 describes the Backus-Naur form (BNF) grammar of the GDTs for the GP-based system. Subsequently, the GP-based system evaluates the GDTs performance, throughout a training dataset, for every generation. Therefore, the GDT with the highest fitness value on the final generation is added to the defined testing historical dataset.

<Tree> ::= If-then-else <Condition> <Tree> <Tree> | Decision

<Condition> ::= <Condition> AND <Condition> | <Condition> OR <Condition> | NOT <Condition> | Indicator <Relation_Operation> Threshold

<Indicator> ::= MA_timePeriod | TRB_timePeriod | ROC_timePeriod | OBV_timePeriod | RSI_timePeriod

<Relation_Operation> := > | < | =

Fig. 3. BNF grammar of the GDTs for the GP-based system.

Look at the BNF of the GDTs: the root of each GDT is an if-then-else statement based on one or more technical analysis indicators. Consequently, the first branch of the GDT can be on either (a) a Boolean (examining whether an indicator is larger than, fewer than, or equivalent to a defined threshold value) or (b) a logic operator (and, or, not), hence be capable of taking numerous Boolean conditions. Thus, the if-then-else branches are viewed as a new GDT or a trading decision to place buy, sell, or hold orders.

Each GDT's performance is assessed using a fitness function based on a confusion matrix presented in Table 1 for the classification of two classes. A confusion matrix contains data regarding actual and predicted trading classifications performed by the GP-based system. Thus, the confusion matrix is an evidence outline of the GP-based system performance of the trading signals classifier; hence, it indicates both the accuracy of the predicted trading class and the likely errors. Consequently, there is a confusion matrix M_i for each trading rule $GDT_i \in T$.

Table 1. Confusion matrix for GP forecasting evaluation.

	Actual Positive	**Actual Negative**
Positive Prediction	True Positive (TP)	False Positive (FP)
Negative Prediction	False Negative (FN)	True Negative (TN)

Accordingly, in this study, we use three evaluation metrics: Rate of Correctness (RC), Rate of Missed Chances (RMC), and Rate of Failure (RF), presented in Equations (7), (8), and (9):

$$RC = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$RMC = \frac{FN}{FN + TP} \quad (8)$$

$$RF = \frac{FP}{FP + TP} \quad (9)$$

In this study, we use the above three evaluation metrics and define a fitness function based on an assigned weight value for each of the three-evaluation metrics. The fitness function is given in Equation (10):

$$f = (RC * w_1) - (RMC * w_2) - (RF * w_3) \quad (10)$$

where w_1, w_2, and w_3 are the weight values for each of RC, RMC, and RF, respectively. [18] and [21] report that these three weights are provided to signify investor preferences and risk tolerance. As an illustration, a high-risk trader would choose to avert investment decision failure. Consequently, a higher weight value for RF have to be used in this case. Nevertheless, [21] shows that adjusting the values of these weights appears not to influence the trading performance of the GP. For this study, we incorporate the role of weight value given our focus on improving the correctness of the predictions while reducing the chance of prediction failure. As a result, these weights have been set to w_{RC}= 0.5, w_{RMC}= 0.2, and w_{RF}= 0.3 correspondingly, and are provided in this approach to signify the importance of each measure to the GP-based system. The optimal value of the fitness function is one that allows the GP-based system to achieve high RC and lower RF.

Throughout the evolutionary process of the GP-based system, we use the three genetic operators: crossover, mutation, and reproduction. Once the last generation of GDTs is reached, the best GDT is selected based on fitness value, where the GP-based system adds the GDT to the testing dataset.

One of the major challenges of evolving GP trading rules is over-fitting to the training dataset. Therefore, the evolved rules should get comparable trading performance when examined with a new dataset to confirm the effectiveness and efficiency of the evolved trading rules. Thus, in this study, we used a random sampling method developed in [7]. The idea is intended as an alternative to evaluating the fitness value over the entire dataset; the calculation is performed on randomly selected data segments which show improved prediction accuracy and out-of-sample outcomes contrasted to benchmark GP and volatility-adjusted fitness as well.

To compare the performance of the GP-based system with other classical trading strategies such as B&H, we look at another standard fitness function. There is a large selection of evaluation measurements for measuring the performance of trading in the financial market, the most employed being the Sharpe ratio, the Sterling ratio, and ROI [17]. The Sharpe ratio presents a risk-adjusted measure of investment performance and has been applied in several studies in the financial literature. The Sharpe ratio is calculated as follows:

$$\text{SharpeRatio} = \frac{\mu - RF}{\sigma} \sqrt{n} \quad (11)$$

where μ is the mean of the investment returns, RF is the risk-free rate, σ is the standard deviation of investment returns, and n the quantity of observation points.

4 Experimental Results

This section presents the experimental results of running the GP-based system on historical daily closing prices for Bitcoin using three datasets over 3.2 years. We begin by observing how the GP-based system affects the fitness values of the created population during the training period. This is because we are interested in seeing whether the trading rules based on technical indicators evolved using GP are providing GP with an advantage in financial forecasting and, if this is the case, how quickly to find the optimal trading rule throughout the evolutionary process. We then continue by presenting a summary statistics comparison of the results between the GP-based system and B&H strategy. At this time, we point out that all the results of fitness values have been normalized to a scale of [0,1]. The other evaluation measures (RC, RMC, RF) are already presented in this scale, and thus no normalization is required.

4.1 GP Parameter Settings

The most satisfactory GP variables settings were decided over initial experiments. Even though the higher GP population sizes result in superior performance, we set the population size to 100 for complications concerning the running time and search space. Crossover, mutation, and reproduction rates were chosen through systematic experimentations. The total number of generations was programmed to 60 because through our examinations, and no substantial improvement was noticed and reported following generation 50. Based on the sensitivity analysis, the values of GP variables are demonstrated in Table 2. For risk evaluation of the generated trading rules, the confidence level was set to 95%.

Table 2. GP parameters setting for the experiment.

Parameter	Value
Number of generations	60
Population size	100
Initialized method	Ramped half and half
Max initial tree depth	5
Selection method	Roulette wheel
Crossover rate	0.6
Mutation rate	0.3
Reproduction rate	0.1
CVaR confidence level	95%

4.2 Experiment Settings

Many applications of machine learning in the financial forecasting literature split the dataset into training and testing datasets. This is well-known as a static methodology because it applies the same dataset throughout the entire testing cycle with no reforming the dataset. Splitting the dataset to two datasets training and testing affects the precision for learning and, hence, the forecasting outcomes' accuracy. One more possible issue with the static methodology is that the forecasting model will depend on the dataset distribution employed in the training and testing datasets. We use the dynamic sliding-window cross-validation methodology to train and test the GP-based forecasting system in this work. Using this methodology, both learning and testing can be performed several times using small amounts of sequentially different datasets.

4.3 Results

This section represents the experiments' results and then compares these results between the GP-based system and B&H strategy. As we have indicated, we are interested in studying the behavior of the GP-based system, and accordingly, the trading rules generated; this requires examining the search space for GP. In other words, does the GP system find potential trading rules from the beginning of the evolutionary procedure since the GP search space is wide, and there is a variety of data input to consider? Or does the GP begin with low trading performance (low fitness values for generated trading rules) owing to the wide search space? These are simply two cases of behavioral forecasting using questions regarding AI techniques we might be asking.

We perform our analysis of the experimental results in two rounds, following the experimental design in [18], in which they applied it in the stock market. In the first round, we compare the training fitness values based on the entire GP population. Therefore, we calculate the average fitness value for the entire GP population of trading rules (GDTs) in which this process is conducted and repeated for each GP generation. Let us denote this average by AvgFitPop. Consequently, we can examine how the GDTs' AvgFitPop changes throughout the 60 generations of each simulation run. We complete calculations of the average fitness value over these 60 simulations runs (number of generations) of AvgFitGen. Figure 4 shows the results of the AvgFitPop for the tested dataset in which each line in the graph represents the average AvgFitPop for a different tested dataset of the Bitcoin market. As we can see from Figure 4, the population of a GP-based system launches at first generation with an average fitness value between 0.1 and 0.2, for all tested datasets. This grows rapidly to 0.3–0.5 and stabilizes around 0.75, with half of the Bitcoin marginally exceeding this fitness value. As we see in Figure 4, the AVgFitPop for ETC-USD is the lowest among the Bitcoins, which is in line as well with the investment performance results in Table 3.

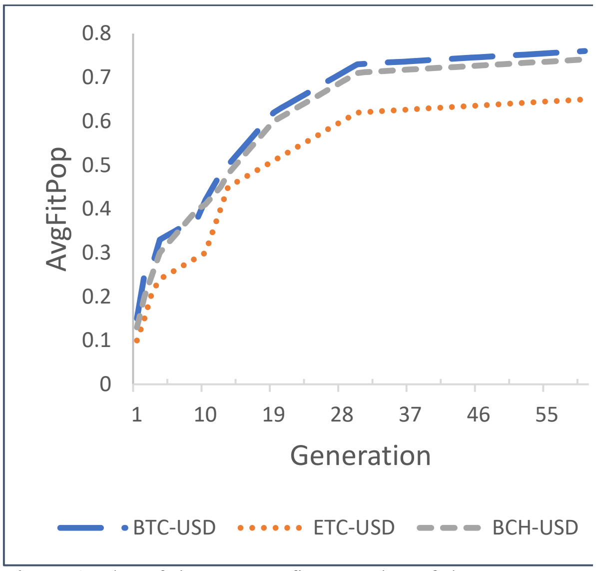
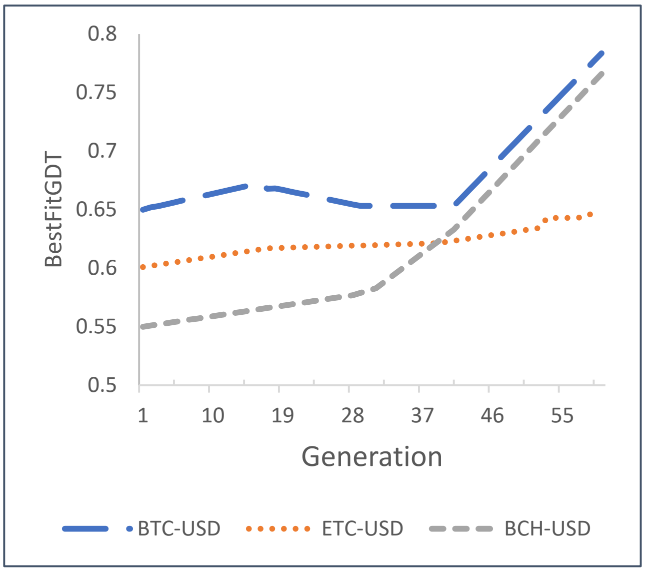


Figure 4. Plot of the average fitness value of the GP generations, calculated using the average fitness value of the GDT population for BTC-USD, ETC-USD, and BCH-USD. This implies that we initially find the average fitness value of the entire population of GDTs, for each of the 60 generations. Then, we find the average fitness value of this number over the 60 simulation runs.

For the second round of our experimental analysis, we compare the fitness value of the best individual GDT (highest fitness value) per generation; this fitness is called BestFitGDT. Thus, as an alternative to calculating the average fitness value of the entire population for each generation, we report only the GDT with the highest fitness value in each generation. We can thus represent by what means the highest fitness value changes over the 60 generations of a single run. We completely define the average fitness value, over these 60 runs for creating GP generations, of BestFitGDT. Figure 5 shows the results of the average best individual GDTs (highest fitness value) per generation. We see that results vary per Bitcoin, even though they appear to follow the same pattern. It is interesting to note from the results that although BestFitGDT on average performs AvgFitPop, this superiority is not exhibited in the average values of Fitness, RC, RF, and RMC.

Figure 5. Plot of the average best fitness value of the GP generations, calculated by considering the best fitness value of the GDTs in each population for BTC-USD, ETC-USD, and BCH-USD. This implies that we initially find the best fitness value for each population of GDTs for each of the 60 generations. Then, we find the average fitness value of this number over the 60 simulation runs.



The results of implementing the GP-based system are shown in Table 3. This table presents ROI, standard deviation, and Sharpe ratio of the B&H strategy and GP-based system. Although GP-based trading rules were profitable in the two cases (positive ROIs from 6.35% to 37.62%), they could not outperform the B&H in ETC-USD, with negative excess returns. However, the overall average ROI is positive (i.e., 10.85%), implying that the generated GP-based trading rules outperform the B&H strategy.

Table 3. Trading performance results: ROI, standard deviation (SD), and Sharpe ratio for B&H strategy and GP-based system for each dataset: BTC-USD, ETC-USD, and BCH-USD.

Ratio	GP	B&H
	BTC-USD	
ROI	37.62%	12.74%
SD	0.0061	0.0135
Sharpe ratio	0.041	0.224
	ETC-USD	
ROI	6.35%	9.54%
SD	0.0152	0.0254
Sharpe ratio	0.145	0.374
	BCH-USD	
ROI	35.47%	9.26%
SD	0.0075	0.0186
Sharpe ratio	0.035	0.265

5 Conclusion

We introduce a GP-based system for discovering profitable trading rules in the Bitcoin market. The examined AI technique was used to decide possible buy-and-sell conditions for Bitcoin to generate potential financial investment results to survive risky investment conditions while generating superior investment returns at minimum risk.

The trading rule base proposes the best combination for trading in the Bitcoin market based on the price data and technical indicators. The proposed GP-based system has some benefits. First, the system creates different decision-trading rules based on various technical indicators and the period used, reflecting correlations between these technical indicators. Second, applying the GP-based system, traders can benefit from the slight price fluctuations in the Bitcoin market. This benefit is associated with the dynamic environment of the Bitcoin market.

Nevertheless, the Bitcoin market transaction cost influences the trading performance of different orders due to repeated trading in the market. Hence, the transaction cost is reflected in the GP-based system for evaluation. Furthermore, the conditional Sharpe ratio was chosen for the fitness function of trading performance to compare it with a B&H trading strategy. To summarize, the GP-based system offers a trading rule base and a forecasting model that is simple, clear, and interpretable for investors in the Bitcoin market.

A more profound improvement of the GP-based system's configuration may produce improved results regarding a more specific dataset, analytical functions, and learning trading rules for further studies. Furthermore, an extended analysis of the impact of the chosen fitness measure might be conducted, examining its consequences in the whole performance of the GP-based system.

Acknowledgements

The authors thank the Deanship of Scientific Research and RSSU at King Saud University for their technical support.

Declaration

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

References

1. Aloud, M. 2016. Investment opportunities forecasting: A genetic programming-based dynamic portfolio trading system under directional-change framework. *J. Comp. Fin* 22(1): 1–35, doi: 10.21314/JCF.2018.346.
2. Aloud, M. 2017. Adaptive GP agent-based trading system under intraday seasonality model. *Intell. Decis. Technol. Int. J* 11(2): 235–251, doi: 10.3233/IDT-170291.
3. Alvarez-Ramirez, J.; Rodriguez, E.; and Ibarra-Valdez, C. 2018. Long-range correlations and asymmetry in the bitcoin market, *Phys. A* 492: 948–955, doi: 10.1016/j.physa.2017.11.025.
4. Balcilar, M.; Bouri, E.; Gupta, R.; and Roubaud, D. 2017. Can volume predict Bitcoin returns and volatility? A quantiles-based approach. *Econ. Modell.* 64: 74–81, doi: 10.1016/j.econmod.2017.03.019.
5. Bariviera, A. 2017. The inefficiency of bitcoin revisited: A dynamic approach. *Econ. Lett.* 161: 1–4, doi: 10.1016/j.econlet.2017.09.013.
6. Bariviera, A.; Basgall, M.; Hasperué, W.; and Naiouf, M. 2017. Some stylized facts of the bitcoin market. *Phys. A* 484 (2017): 82–90, doi: 10.1016/j.physa.2017.04.159.
7. Berutich, J.; López, F.; Luna, F.; and Quintana, D. 2016. Robust technical trading strategies using GP for algorithmic portfolio selection. *Expert Syst. Appl.* 46: 307–315, doi: 10.1016/j.eswa.2015.10.040.
8. Bouri, E.; Gupta, R.; Tiwari, A.K.; and Roubaud, D. 2017. Does Bitcoin hedge global uncertainty? Evidence from wavelet-based quantile-in-quantile regressions. *Fin. Res. Lett.* 23: 87–95 doi: 10.1016/j.frl.2017.02.009.
9. Bouri, E.; Shahzad, S.J.H.; and Roubaud, D. 2019. Co-explosivity in the cryptocurrency market. *Fin. Res. Lett.* 29: 178–183, doi: 10.1016/j.frl.2018.07.005.
10. Brière, M.; Oosterlinck, K.; and Szafarz, A. 2015. Virtual currency, tangible return: Portfolio diversification with bitcoin. *J. Asset Manag.* 16(6): 365–373, doi: 10.2139/ssrn.2324780.
11. Brock, W.; Lakonishok, J.; and LeBaron, B. 1992. Simple technical trading rules and the stochastic properties of stock returns. *J. Fin.* 47(5): 1731–1764, doi: 10.2307/2328994.
12. Chuen Lee, D.; Guo, L.; and Wang, Y. (2017). Cryptocurrency: A new investment opportunity? *J. Altern. Invest.* 20(3): 16–40, doi: 10.3905/jai.2018.20.3.016.
13. Dastgir, S.; Demir, E.; Downing, G.; Gozgor, G.; and Lau, C.K.M. 2019. The causal relationship between Bitcoin attention and Bitcoin returns: Evidence from the copulabased Granger causality test. *Fin. Res. Lett.* 28: 160–164, doi: 10.1016/j.frl.2018.04.019.

14. Dyhrberg, A.H. 2016. Bitcoin, gold and the dollar—A GARCH volatility analysis. *Finance. Res. Lett.* 16: 85–92, doi: 10.1016/j.frl.2015.10.008.
15. Holland, J.H. 1975. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press.
16. KiHoon Hong 2017. Bitcoin as an alternative investment vehicle. *Inf. Technol. Manag.* 18(4): 265–275, doi: 10.1007/s10799-016-0264-6.
17. Iba, H. and Aranha, C.C. 2012. Introduction to Genetic Algorithms. *In Practical Applications of Evolutionary Computation to Financial Engineering. Adaptation, Learning, and Optimization*, vol. 11. Berlin, Heidelberg: Springer.
18. Kampouridis, M. and Tsang, E. 2012. Investment opportunities forecasting: Extending the grammar of a GP-based tool. *Int. J. Comp. Intell. Syst*. 5(3): 530–541, doi: 10.1080/18756891.2012.696918.
19. Kim, Y.B.; Kim, J.G.; Kim, W.; Im, J.H.; Kim, T.H.; and Kang, S.J. 2016. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLOS ONE* 11(8): e0161197, doi: 10.1371/journal.pone.0161197.
20. Koza, J.R. 1992. Genetic programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press.
21. Li, J. 2001. FGP: A Genetic Programming-based Financial Forecasting Tool. PhD thesis, Department of Computer Science, University of Essex.
22. Lohpetch, D. and Corne, D. 2009. Discovering effective technical trading rules with genetic programming: towards robustly outperforming buy-and-hold, 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (2009), pp. 439–444
23. Mousavi, S.; Esfahanipour, A.; and Zarandi, M. 2014. A novel approach to dynamic portfolio trading system using multitree genetic programming. *Knowl. Based Syst.* 66: 68–81, doi: 10.1016/j.knosys.2014.04.018.
24. Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com.
25. Poli, R.; Langdon, W.B.; and McPhee, N.F. 2008. A Field Guide to Genetic Programming.
26. Schwert, G. 2003. Anomalies and market efficiency. *Handb. Econ. Fin.* 1: 939–974, doi: 10.2139/ssrn.338080.
27. Tiwari, A.K.; Jana, R.K.; Das, D.; and Roubaud, D. 2018. Informational efficiency of Bitcoin—an extension. *Econ. Lett.* 163: 106–109 doi: 10.1016/j.econlet.2017.12.006.
28. Wong, W.; Manzur, M.; and Chew, B. 2003. How rewarding is technical analysis? Evidence from Singapore stock market. *Appl. Financ. Econ.* 13(7): 543–551, doi: 10.1080/0960310022000020906.
29. Yan, W. and Clack, C.D. 2010. Evolving robust GP solutions for hedge fund stock selection in emerging markets. *Soft Comput.* 15(1): 37–50, doi: 10.1145/1276958.1277384.