Neural Computing and Applications (2022) 34:20715–20756
https://doi.org/10.1007/s00521-022-07543-4

ORIGINAL ARTICLE

To learn or not to learn? Evaluating autonomous, adaptive, automated traders in cryptocurrencies financial bubbles

Alfonso Guarino[1] · Luca Grilli[2] · Domenico Santoro[3] · Francesco Messina[4] · Rocco Zaccagnino[5]

Received: 23 February 2022 / Accepted: 13 June 2022 / Published online: 27 July 2022

Abstract

Financial bubbles represent a severe problem for investors. In particular, the cryptocurrency market has witnessed the bursting of different bubbles in the last decade, which in turn have had spillovers on all the markets and real economies of countries. These kinds of markets and their unique characteristics are of great interest to researchers. Generally, investors and financial operators study market trends to understand when bubbles might occur using technical analysis tools. Such tools, which have been historically used, resulted in being precious allies at the basis of more advanced systems. In this regard, different autonomous, adaptive and automated trading agents have been introduced in the literature to study several kinds of markets. Among these, we can distinguish between agents with *Zero/Minimal Intelligence (ZI/MI)* and *Computational Intelligence (CI)*-based agents. The first ones typically trade on the market without resorting to complex learning strategies; the second ones usually use (deep) reinforcement learning mechanisms. However, these trading agents have never been tested on the cryptocurrencies market and related financial bubbles, which are still mostly overlooked in the literature. It is unclear how these agents can make profits/losses before, during, and after a bubble to adjust their strategy and avoid critical situations. This paper compares a broad set of trading agents (between *ZI/MI* and *CI* ones) and evaluates them with well-known financial indicators (e.g., volatility, returns *Sharpe ratio*, drawdown, *Sortino* and *Omega ratio*). Among the experiment's outcomes, *ZI/MI* agents were more explainable than *CI* ones. Based on the results obtained above, we introduce *GGSMZ*, a trading agent relying on a neuro-fuzzy mechanism. The neuro-fuzzy system is able to learn from the trades performed by the agents adopted in the previous stage. *GGSMZ*'s performances overcome those of other tested agents. We argue that *GGSMZ* could be used by investors as a decision support tool.

Keywords Zero-intelligence trader · Reinforcement learning trader · Cryptocurrency · Financial bubbles

1 Introduction

Price prediction in financial and real markets is a problem that industry experts and scholars have always studied. Forecasting has become an increasingly complex process, especially today, where markets are fully connected, and information circulates easier and faster. However, in parallel with the increase in forecasting complexity, different tools have been developed to carry out machine-assisted forecasting. For example, various studies successfully forecast stock prices [8, 9, 11, 57, 78] (or more specifically daily close price of stocks [49]), stock market index performance [71], carbon emissions futures prices [6], the price of gold [45], the price of oil [10] and the price of various commodities [5] like coffee, cocoa, etc.

In recent years, Bitcoin has attracted considerable attention from investors, policy makers, and the media. This is not surprising since its price increased from a value of nearly zero in 2009 to almost $20,000 in December 2017. This was accompanied by a tremendous increase both in the number of Bitcoins in circulation and the Bitcoin market capitalization, being around 16.8 million Bitcoins and $300 billion, respectively. Policymakers around the world have raised concerns because Bitcoin is anonymous, decentralized and unregulated, and it could be a bubble, threatening the stability of the financial system [16, 18, 50]. Nonetheless, investors appear to be attracted by the potential to earn high returns, the introduction of Bitcoin derivatives, and the potential

Extended author information available on the last page of the article

diversification benefits. Thanks to these features, the focus has shifted to this market, and it is possible to find a lot of studies that develop and test prediction models for the Bitcoin market. For example, Shah and Zhang [67] propose a trading strategy based on a Bayesian regression model that allows them to earn substantial returns when tested on real data. In a similar context, Madan, Saluja, and Zhao [58] propose the use of binomial regressions, support vector machines and random forest algorithms to predict the sign of the Bitcoin price change. Using machine learning optimization, Greaves and Au [39] obtain an up-down Bitcoin price movement classification accuracy of roughly 55%. In more recent times, Atsalakis et al. [7] developed a neuro-fuzzy system for Bitcoin price prediction with root mean squared error (RMSE) of 0.0376. Lastly, Mudassir et al. [61] proposed a machine learning approach exploiting joint regressors forecasting for Bitcoin price prediction, which has proven to be effective also in medium-term predictions.

In step with the introduction of prediction methods and systems, a set of tools to study the financial market has been proposed. There is a long tradition of research to automatically discover, implement, and fine-tune strategies for autonomous adaptive automated trading in financial markets, with a sequence of research papers on this topic published at major artificial intelligence (AI) conferences and in prestigious journals. Among these, we can distinguish between two broader sets of automatic traders: *Zero/Minimal Intelligence (ZI/MI)* traders [19], and *Computational Intelligence (CI)* traders [55]. The first set concerns agents that trade on the market without resorting to complex learning strategies. The second set includes traders that usually exploit (deep) reinforcement learning mechanisms.

1.1 Overview

Objective In this paper, our goal is to study and evaluate the behavior of different agents in the Bitcoin market during financial bubbles (see a visual abstract in Fig. 1). We use two different types of trading agents to analyze their ability to identify the particular market phases (*before/during/after* the bubble) and their behavior in the investment phase. We want to analyze *ZI/MI* and *CI* agents in different scenarios, such as various stages of a financial bubble and compare these agents to understand which ones make more profit in anomalous situations. (Over the years, these bubbles in crypto seem to be more frequent.) Finally, based on what we have observed experimenting with such trading agents, we aim to develop our own automatic trader that operates during the bursting of a bubble. The ultimate goal is to define and introduce a trader outperforming state-of-the-art ones.

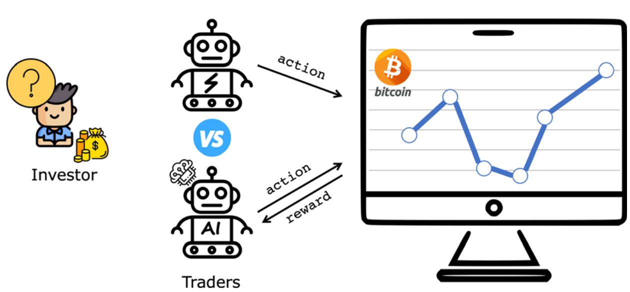


Fig. 1 Visual abstract of our proposal

Motivation Bitcoin has particular attributes that introduce additional challenges when building a model to forecast its price movements. For example, its volatility is considerably higher than that of gold, the US dollar or stock markets [13], and it is particularly susceptible to regulatory and market events [31]. Additionally, prices may be manipulated through suspicious trading activity [34]. Our interest is driven by the absence in the literature of a comparison between two types of agents, i.e., *ZI/MI* and *CI*. Moreover, since the crypto market is subjected to the effect of financial bubbles more and more frequently, it is interesting to study how these agents behave in the different market phases [18]. Furthermore, since the capitalization of these markets is always higher, studying which agent has the best behavior could allow human traders to benefit from its strategies (added value not to be underestimated, from an economic point of view). Finally, the analyses made on this market could be transferred to newbies that have the same characteristics (e.g., high volatility, high frequency of bubbles, ...).

The proposed approach In order to compare the *ZI/MI* and *CI* traders, we considered the ones in [19]—a broad collection of *ZI/MI* agents including *ZIC*, *ZIP*, *GDX*, *AA*, and *GVWY* (see Sect. 2.1)—, and the ones in [55]—a collection of *CI* traders including *A2C*, *DDPG*, *TD3*, *PPO* and *SAC* (see Sect. 2.2). We compared such agents on the Bitcoin market from 2015 to 2018, from 2019 to 2021, and Ethereum market from 2019 to 2021 showing how *ZI/MI* agents were more explainable than *CI* ones. Building upon the achieved results, we introduce a neuro-fuzzy system, which is trained on the basis of the experience made by the best agents found in the previous phase and whose aim is to suggest the best operation to perform on the market in a specific period. Neuro-fuzzy systems are hybrid models that combine the functionality of fuzzy systems with the learning abilities of neural networks [59]. Consequently, one of the main advantages of a neuro-fuzzy system is its ability to learn and use linguistic variables to model the input–output relationships of a given system. In addition, using neural network learning algorithms, the fuzzy subsystem can automatically adjust the parameters of the fuzzy rules, thereby producing a data-driving-based rule for more accurate forecasting.

The adaptive network-based fuzzy inference system (ANFIS) used in the present study was proposed by [47]. It consists of five layers of adaptive networks with several inputs and one output. Such fuzzy system thus created is placed at the core of a new trading agent, namely *GGSMZ*, and was tested on the Bitcoin and Ethereum market also during the bubbles of 2018 and 2021. The results show how *GGSMZ* outperforms other agents under many indicators in various market situations, and it can be a great trading support tool.

Key points The main contributions of this paper are:

- The study of the efficacy of Zero/Minimal Intelligence and Computational Intelligence based agents in terms of economic return in trading cryptocurrencies. In particular, we have analyzed the behavior of such trading agents also during financial bubbles. To the best of our knowledge, the current study is one of the first that compares such a broad range of trading agents (*ZI/MI* and *CI*) in similar scenarios.
- The analyzes show that some *ZI* agents can identify the phase of a bubble based on volatility. While for *CI* agents, their behavior is always excellent at any stage of the market. However, despite the optimality of the investment phase, *CI* agents lose explainability due to their depth of training.
- In the light of the above, we have built a novel learning-based trading agent, namely *GGSMZ*, that is based on an adaptive neuro-fuzzy inference system (ANFIS) approach. We have tested such an agent and compared its performance against the most promising ones found in the previous step of the project. The results indicate that *GGSMZ* was able to learn from the best choices of the *CI* agents and to use them to put himself in a profitable position. Investors could use our neuro-fuzzy model as a decision support tool. In the literature, few agents perform actions based on RL trading agents, placing us among the first to develop models of this type (particularly with neuro-fuzzy rules).

Paper's organization This article is structured as follows: Sect. 2 presents the literature review, highlighting initiative and studies that show contact points with the presented paper; Sect. 3 elucidates the set of methods used and the dataset adopted; Sect. 4 provides details and a step-by-step description of the experiments carried out, ending with the discussion on the obtained results; Sect. 5 presents *GGSMZ* and the neuro-fuzzy system at its core. It also illustrates the methodology adopted to build the neuro-fuzzy system and the experiments made; Sect. 6 concludes the paper with final remarks and an overview of the work done, and it traces the path for future works. Finally, Appendix 1 repeats the experiments previously carried out with the same agents on a different market, that of currencies (FOREX).

2 Literature review

This section focuses on providing the key literature referring to *ZI/MI* trading agents (Sect. 2.1), *CI* agents (Sect. 2.2), and dwells on the works that compared different agents (Sect. 2.3).

2.1 Zero-intelligence and minimal intelligence trading agents

As many world's major financial markets have lived a shift from physical stock exchanges to electronic markets, many software agents with various degrees of artificial intelligence have started to replace human traders. One relevant example of these software agents is represented by the Zero/Minimal Intelligence trading agents, which we briefly sketch in the following.

The birth of the ZI is due to Becker [14], who developed a model thanks to which he was able to discover that the taking of the supply and demand curves is associated with a behavior of the agents (traders) without any individual rationality. These behaviors are due to a market mechanism. On this idea, the first to consider a market mechanism in continuous double auctions were Gode and Sunder [38]. In particular, they consider two types of markets, each consisting of twelve agents, divided into two groups: buyer and seller. Traders can submit shouts at any time for one unit at a time. The key feature is that buyers and sellers can modify the offer after submitting a price, e.g., buyers may submit a higher price and sellers a lower price than the bid. The subjects operating in this market are human agents, who can shout prices at any time and whose price choice is governed by strategy and ZI agents (that do not learn strategies): In particular, the ZI are classifiable in *ZI Unconstrained* that can shout prices at a loss compared to their booking prices; and *ZI Constrained* (ZIC), for which this mechanism is not allowed and the shouted price cannot allow losses. As a result, Gode and Sunder found that in markets populated by human traders and ZI Constrained there is a rapid convergence toward the equilibrium price, while in markets populated by ZI Uncontrained this convergence did not occur (measuring a higher profits dispersion). According to Gode, Spear and Sunder [37], the result of this analysis highlights how the dominant factor in auctions is not the strategy chosen by the trader, but the market mechanism. The effect of this mechanism produces a rational market behavior even in the presence of irrational agents, going against the classic economic theories according to which the perfect rationality of the agents allowed an optimal allocation in the markets.

There have been several extensions of this model. Friedman [32] and Wilson [77] introduced two behavioral models for ZI agents: (i) Bayesian Game Against Nature (BGAN) with bounded rationality, to explain the bid-ask spread; (ii) Waiting Game Double Auction (WGDA) with completely rationality, to check what happens in markets with an unequal number of traders. Jamal and Sunder [46] used ZI traders whose price limits use heuristic and Bayesian rules, demonstrating the achievement of Bayesian equilibrium.

Gjerstad and Dickhaut [35] developed a trading strategy called GD to achieve competitive equilibrium outcomes (prices and allocations) in a market where individual choices are made myopically using heuristic beliefs. Their model aims to strike a balance between the approach taken by Wilson and the one by Gode and Sunder, while it also boasts the merit of avoiding the positive autocorrelation of price changes found in Friedman's model. In addition, Gode and Sunder [36] examined the effect of unconstrained price controls, showing that traders do not adjust the strategy in the case of price controls.

Among various criticisms that have been put forward to the model of Gode and Sunder, most notably is that of Cliff and Bruten [24], which have shown that the accuracy with which the model captures the behavior of real markets is dependent on the supply and demand functions. The condition demonstrated by Gode and Sunder only occurs when these functions are symmetrical (a situation that does not occur in reality), making the ZI model weak in representing the results.

Cliff and Bruten [24] have developed an agent called *Zero Intelligence Plus* (ZIP) with a learning mechanism, through which the agent maintains a profit margin that reflects that individual's belief of the profit that can be obtained from a successful transaction, therefore function of the trader's reservation price. In this case, the authors demonstrated how ZIP behavior allows for better performance than ZI. One of the main features of the marketplace they used is that, at any given time, only one agent can announce a bid/offer. This agent is chosen at random by the market institution.

Inspired by this ZIP agent work, while considering unrealistic the marketplace bid/offer procedure, Priest and van Tol [62] have developed a new agent called PS-agent. The performance of ZIP agents and PS-agents has been compared in a marketplace characterized by a persistent shout double auction mechanism, where a trader's current bid or offer will persist until the trader makes another. As a result, PS-agents turn out as more rapid in converging to equilibrium than the ZIP agents. Then, ZIP and a modified version of GD, renamed as MGD, have been tested by Das et al. [25] in CDA markets, in order to study the interactions between human and artificial traders. Another extension of GD model, the GDX, has been developed by Tesauro and Bredin [72]. The GDX not only involves a belief function that an agent builds to indicate whether a particular shout is likely to be accepted in the market, but it also considers the time left before the auction closes.

Inspired by Das et al. [25], Grossklags and Schmidt [40, 41] have studied the effect of knowledge/ignorance of the presence of trader-agents on the behavior of human traders, highlighting a "knowledge effect" capable of altering market dynamics. The ZIP trader has been modified by Cliff [19] through genetic algorithms to study the evolution of strategies or by extending the parameters from 8 to 60—introducing the *ZIP60* [20]. In this paper, it has been observed that, thanks to a simple search/optimization process, is possible to found ZIP60 parameter-vectors that outperform ZIP8.

The introduction of ZI and ZIP agents marked an important step in trading strategies [51].

A further step forward has been made by Vytelingum et al. [74] with the presentation of a dominating strategy called Adaptive Aggressive (AA), which has been widely considered to be the best performing strategy in the public domain. The crucial peculiarity of AA is having both a short and a long-term learning mechanism to adapt its behavior to changing market conditions. Later on, AA's supposed dominance has been tested against two novel algorithms known as GVWY and SHVR [21], which involve no AI or machine learning at all. The result is surprising: GVWY and SHVR can outperform AA and many of the other AI/ML-based trader-agent strategies.

2.2 Computational Intelligence traders

The increasingly strong use of neural networks, also in the financial field, has made it possible to combine the high ability to represent features with reinforcement learning. For example, Deng et al. [29], starting from the idea that computers can beat experienced traders, proposed a recurrent neural network (RNN) for sensing the dynamic market condition for feature learning and combined it with a RL framework that makes trading decisions. Almahdi and Yang [2] proposed a recurrent reinforcement learning (RRL) method for portfolio allocation, with a risk-adjusted performance objective function (Calmar ratio) to obtain signals and asset weights, showing how this method outperforms hedge fund benchmarks. Jiang, Xu and Liang [48] proposed a RL framework for asset allocation, consisting of a convolutional neural network (CNN), an RNN and a long short-term memory (LSTM) in a particular scheme with deep deterministic policy gradient, showing how, on a crypto market, this framework monopolize top positions in various experiments. Liu et al. [56] proposed an adaptive trading model, namely iRDPG, to develop trading strategies useful to balance exploration and exploitation combining RL techniques with GRU-based networks. Or again, on financial signal' study, Ye et al. [79] built a new RL framework, the State-Augmented RL framework (SARL), that augments asset information with their price movement prediction as additional states, to incorporate data heterogeneity and environment uncertainty of the market, testing it on the Bitcoin and stock markets, and demonstrating the importance of state augmentation. Wang et al. [75] proposed *AlphaStock*, a new type of strategy based on the *Attention Mechanism* to model the price relations for buying and selling strategy, testing it on the USA and Chinese markets and highlighting the robustness of their model. Wang et al. [76], considering the market conditions, proposed a Deep RL method to optimize the investment policy (DeepTrader); a model that considers macro-market conditions as an indicator and is able to capture the spatial and temporal dependencies between assets.

Recently, Yang, Gao and Wang [55] due to the difficulty of developing RL models under the programming aspect, created a new open-source framework (*FinRL*) to help quantitative traders. Several works have been developed on this framework, such as Guan and Liu [42] who used it to explain the trading strategies of a DRL agent for portfolio management in three steps; or Bau and Liu [12] who proposed a DRL multi-agent-based on FinRL, which capture high-level complexity, to optimize the process of selling a large number of stocks (called liquidation). Thanks to the ease of implementation and the number of agents included, in line with the previous authors, we also used *FinRL* for the subsequent analyzes.

2.3 Comparison and evaluation of different trading agents

Since Gode and Sunder developed the *ZIC* agent, several papers have addressed the topic of comparing bidding strategies and agents' behaviors. First, Cason and Friedman [17] evaluated the performances of Wilson's waiting game/Dutch auction (WGDA) model, Friedman Bayesian game against nature (BGAN) and Gode and Sunder *ZIC* agent in price formation in Double Action Markets. The results suggested that models which rely most heavily on trader rationality, as WGDA and BGAN, have less ability to describe markets behavior than *ZIC* agents, which requires very little trader rationality. Nevertheless, the authors suggest new experiments since the conditions of their experiment did not give a fair chance to WGDA model. In 2001, in their already mentioned work, Das et al. [25] applied the laboratory methods of experimental economics to compare Extended-GD agent and ZIP agent against human traders in a continuous double auction (CDA) mechanism. Ten years later, De Luca and Cliff [26] recreated the same experiment in a trading system called OpEx, obtaining the same results as Das et al. [25] in terms of comparison between robot traders and human traders, as ZIP and GDX agents had consistently outperformed human traders, and observing that GDX had outperformed ZIP. At the same time, in 2002 Tesauro and Bredin [72] pointed out that ZIP slightly outperformed EGD. In addition, another work by De Luca and Cliff [27] confirmed that "Adaptive Aggressive" (AA) algorithmic traders of Vytelingum [74] outperformed ZIP, GD, and GDX in agent vs agent experiments in CDA markets, as Vytelingum himself claimed. A few years later, Vach [73] questioned the dominance of AA over ZIP and GDX agents by designing symmetric agent–agent experiments with a variable composition of agent population. Surprisingly, GDX is a dominant strategy over AA in many experiments in this work in contrast to the previous literature. In 2019 Cliff [23] reaches a similar result: in markets with dynamically varying supply and demand, so market environments that are in various ways more realistic and closer to real-world financial markets, AA can be routinely outperformed by more straightforward trading strategies. On the other hand, AA remains dominant only in highly simplified market scenarios and maybe because AA was designed with exactly those simplified experimental markets in mind. In the same year, Snashall and Cliff [69] made another step forward by exhaustively testing AA across a sufficiently wide range of market scenarios against GDX. The outcome was that not only AA is outperformed by GDX in more realistic market environments, but also in the simple experiment conditions that were used in the original AA papers. So, the various results achieved in the previous years and well known in the literature could no longer be fully trusted. On this path, one year later, Rollins and Cliff [64], employing a new version of BSE called Threaded-BSE (TBSE) by Rollins [64], questioned the original benchmark dominance-hierarchy AA $>$ GDX $>$ ZIP $>$ ZIC, obtained in the BSE, and got a different result: The dominance-hierarchy is instead ZIP $>$ AA $>$ *ZIC* $>$ GDX. The authors also guess that this new achievement is probably due to the previous use of simplistic simulation methodologies.

Thus, several experiments have been conducted with autonomous, adaptive, automated traders, but to the best of our knowledge the following aspects have been overlooked:

– There is lack of throughout comparisons in the cryptocurrencies market, and, in particular, in the Bitcoin and Ethereum market;

- There is lack of experiments on how the trading agents behave during financial bubbles—except the study by Duffy and Unver [30] that successfully verified whether ZIC traders can generate asset price bubbles and crashes of the type observed in a series of laboratory asset market experiments[1].
- There is lack of comparison between *ZI/MI* traders and other traders adopting higher degree of AI techniques, such as *CI* ones.

We remark that this work aims to fill the gaps mentioned above offering a comparison between *ZI/MI* and *CI* trading agents on the crypto market over different phases, including during the bursting of a financial bubble. Furthermore, building upon the experiments carried out, we propose *GGSMZ*, a trading agent relying on a neuro-fuzzy system which outperforms other analyzed traders.

3 Methods and materials

In this section, we first present and provide details about the traders adopted, from *ZI/MI* ones (Sect. 3.1) to the *CI* ones (Sect. 3.2). Then, we sketch technical details on adaptive neuro-fuzzy systems at the basis of the proposed *GGSMZ* trader (Sect. 3.3). Lastly, we show the dataset employed for running the experiments (Sect. 3.4).

3.1 Zero/Minimal Intelligence traders

We used the following *ZI/MI* traders:

- Zero Intelligence Constrained (**ZIC**), the *ZIC* trader generates random bids or offers (depending on whether it is a buyer or a seller) distributed independently, identically and uniformly over the entire feasible range of trading prices from 1 to 200. The trader has no memory of past market activity, and each trader has an equal probability of being the next trader to make a bid or an ask. The assumption by Gode and Sunder [38]: (i) each ask, bid, and transaction is valid for a single unit; (ii) a transaction cancels any unaccepted bids and offers; (iii) when a bid and ask crosses, the transaction price is equal to the earlier of the two. Buyer's profit from buying the ith unit is given by the difference between the redemption value of the unit i, v_i, and its price p_i: $\pi_i^B = v_i - p_i$ Seller's profit from selling the ith is given by the difference between the price of the unit i, p_i, and its cost to the seller c_i: $\pi_i^S = p_i - c_i$. Every trader has to sell the unit i before selling the unit $i+1$. The agents are subject to budget constraints: If they generate a bid (to buy) above their redemption value or an offer (to sell) below their cost, such actions are considered invalid and are ignored by the market. So, the market forbids traders to buy or sell at a loss. Therefore, the support of the distribution that generated the uniform random bids was restricted between 1 and the redemption value of the bidder, while the uniform distribution of random ask was restricted to the range between the seller's cost and 200.
- Zero Intelligence Plus (**ZIP**), it is an evolution of ZIC. Individual traders adjust their profit margins using market price information thanks to simple adaptive mechanisms. More specifically, they adjust the profit margins up or down based on the prices of bids and offers made by other traders and whether these shouts are accepted, leading to deals or ignored. As a result, the performances of these agents sensibly increase. The adjustments depend on four factors. The first is whether the trader is active or inactive – in other words, if it is still able to make transactions or not. The other three factors are connected to the most recent shout: its price q, whether it was a bid or an offer and whether it was accepted or rejected. At a given time t, an individual ZIP trader i calculates the shout price $s_i(t)$ for a unit j by multiplying the trader's real-valued profit margin $\mu_i(t)$ by the limit price $\lambda_{i,j}$ of the unit: $s_i(t) = \lambda_{i,j}[1 + \mu_i(t)]$. *Sellers*: $\mu_i(t) \in [0, \infty)\forall t$, so that s_i is raised by increasing μ_i or lowered by decreasing μ_i; *Buyers*: $\mu_i(t) \in [-1, 0]\forall t$, so that s_i is raised by decreasing μ_i or lowered by increasing μ_i. In principle, a ZIP buyer will buy from any seller that makes an offer less than the buyer's current bid shout price; similarly, a ZIP seller sells to any buyer making a bid greater than the seller's current offer shout price. The aim is that the value of μ_i for each trader should alter dynamically, in response to the actions of other traders in the market, increasing or decreasing to maintain a competitive match between that trader's shout-price and the shouts of the other traders.
- Gjerstad-Dickhaut (**GDX**), the GDX agent is the result of an improvement process that begins from Gjerstad and Dickhaut [35] with their GD trader and ends up with Tesauro and Bredin [72]. As ZIP trader, GD agent can trade profitably by adapting its behavior over time, in response to market events. In contrast to the ZIP work, Gjerstad's trading algorithm uses frequentist statistics, gradually constructing and refining a belief function that estimates the likelihood for a bid or offer to be accepted in the market at any particular time, mapping from price of the order to its probability of success. The original GD agent was developed for a market where there was no queue, so old bids or asks

[1] Their work employed populations of ZIC placed in the various laboratory market environments that have given rise to price bubbles.

were erased as soon as there was a more favorable bid/ask or a trade. In Das et al. [25] version of the CDA market, unmatched orders can be retained in a queue, and therefore the notion of an unaccepted bid or ask becomes ill-defined. In their version of GD agent, called Modified-GD (MGD), they overcome this problem by introducing into the GD algorithm a "grace period" t_g. Another modification to GD addressed the need to handle a vector of limit prices since the original algorithm assumed a single tradeable unit. Finally, an extension of MGD was reported by IBM researchers Tesauro and Bredin in 2002 and took the name of GDX [72]. In their work, Tesauro and Bredin combine the belief function with a forecast of how it changes over time. The result is an optimization of cumulative long-term discounted profitability, whereas GD traders merely optimize immediate profits.

- Adaptive Aggressiveness (**AA**), the AA agent has both a short- and long-term learning mechanism to adapt its behavior to changing market. In particular, in the static case, the agent can be effective by assuming that the competitive equilibrium does not change significantly, whereas in the dynamic case, it can make no such assumption and must learn, assuming that this competitive equilibrium may change. The focus is on the bidding aggressiveness shown by the agent because it describes how the agent manages the trade-off between profit and probability of transaction. Whenever the agent submits a bid or an ask, a short-term learning mechanism is employed to adjust agent's level of aggressiveness $r \in [-1, 1]$. For $r < 0$, the agent adopts an aggressive strategy, which trades-off profit to improve its probability of transacting in the market. For $r > 0$, the agent adopts a passive strategy, waiting for more profitable transactions than at and willing to trade-off its chance of transacting for a higher expected profit. If $r = 0$, the agent is neutral and submits offers at what it believes is the competitive equilibrium price, which is the expected transaction price. How the level of aggressiveness influences an agent's choice of which bids or asks to submit in the market depends on a long-term learning strategy, based on market information observed after every transaction. In a few words, an AA agent has two principal decision-making components: *(i)* a bidding layer that specifies what bid or ask should be submitted based on its current degree of aggressiveness; *(ii)* an adaptive layer to update its behavior according to the prevailing market conditions. Given a target price τ and a set of bidding rules, the first layer determines which bids or asks to submit. The aggressiveness model gives a mapping function to τ employing the agent's current degree of aggressiveness, its limit price $\hat{p}^*$ and an intrinsic parameter θ.
- Giveaway (**GVWY**), the GVWY agent simply sets its quote price equal to its limit price, giving away any chance of surplus. GVWY seller: $P_{sq(GVWY)}(t) = \lambda^S$ GVWY buyer: $P_{bq(GVWY)}(t) = \lambda^B$ where S and B are, respectively, the seller's limit price and buyer's limit price. Anyway, the GVWY trader can enter in a surplus-generating transaction: If a GVWY buyer quotes its limit price λ^B and the current best ask $p^*_{ask} < \lambda^B$, the GVWY buyer is matched with whichever seller issued that best ask and the transaction goes through at price p^*_{ask} yielding a $\lambda^B - p^*_{ask}$ surplus for the GVWY buyer.

3.2 Computational Intelligence-based traders

On the other side, as previously said we used *FinRL* [55] as a reinforcement learning (RL) framework. This framework, consisting of 3 layers, encapsulates historical trading data in training environments and provides useful demonstrative trading tasks to users for develop their strategies. The first layer, Application, is used to transform the trading strategy into deep reinforcement learning (DRL) by defining: the state space $\mathcal{S}$ (that describes how the agent perceives the environment), the action space $\mathcal{A}$ (that describes the allowed actions for an agent) and the reward function (as an incentive for the agent to learn better policy, Sharpe ratio in this case). The second layer, Agent, allows the user to play with the standard DRL algorithms like *Stable Baseline 3* [63], *RLlib* [52] and *ElegantRL* [54]. Finally, the third layer, Environment, simulates real world markets to learn a new strategy. Here the agent updates iteratively and obtains a trading strategy to maximize the expected return. The methods used in FinRL framework for representing the training agents are:

- Asynchronous Advantage Actor Critic (**A2C**) [60], a policy optimization method that performs gradient ascent to maximize performance. Defining a state $\mathbf{s}_t$ in which an actor selects an action $\mathbf{a}_t$ according to the policy π, r_t the scalar rewards such that $R_t = \sum_{k=0}^{+\infty} \gamma^k r_{t+k}$ is the total accumulated return from t with discount factor γ, $Q^\pi(\mathbf{s}, \mathbf{a})) = \mathbb{E}[R_t | \mathbf{s}_t = \mathbf{s}, \mathbf{a}]$ the action value following the policy π, $Q^*(\mathbf{s}, \mathbf{a}) = \max_\pi Q^\pi(\mathbf{s}, \mathbf{a})$ the maximum action value for the state s, $V^\pi(\mathbf{s}) = \mathbb{E}[R_t | \mathbf{s}_t = \mathbf{s}]$ the value of state s under the policy π and $Q(\mathbf{s}, \mathbf{a}; \theta)$ an approximate action-value function, Mnih et al. [60] starting from the one step Q-Learning loss

$$J(\theta_i) = \mathbb{E}[r + \gamma \max_{a'} Q(\mathbf{s}', \mathbf{a}'; \theta_{i-1} - Q(\mathbf{s}, \mathbf{a}; \theta_i)]^2$$

have designed new methods to find a RL method that is trainable through neural networks without excessive use of resources. In this vein, the authors have introduced a property modification of Asynchronous one-step Q-Learning (in which each thread interacts with its own copy of the environment and computes a gradient of the loss), Asynchronous n-step Q-Learning and Asynchronous advantage actor-critic (called A3C) that maintains a policy $\pi(\mathbf{a}_t|\mathbf{s}_t;\theta)$ and an estimate of the value function $V(\mathbf{s}_t;\theta_v)$.

- Deep Deterministic Policy Gradient (**DDPG**) [53], a first type of mixed method between Q-Learning and Policy Optimization that use each to improve the other. In this situation, since it is not possible to apply Q-Learning to continuous action spaces, Lillicrap et al. [53] use an approach based on the deterministic policy gradient (DPG). Considering $\mathbf{s}_t$ the state in which an agent takes an action $\mathbf{a}_t$ and obtain the reward r_t, ρ^π the discounted state visitation distribution for a policy π, Q the off-policy, $\mu(s) = argmax_a Q(\mathbf{s}, \mathbf{a})$ a greedy policy, γ the discount factor and β a stochastic behavior policy, it is possible to start from Q-Learning loss

$$J(\theta^Q) = \mathbb{E}_{\mathbf{s}_t \sim \rho^\beta, \mathbf{a}_t \sim \beta, r_t \sim E}[(Q(\mathbf{s}_t, \mathbf{a}_t|\theta^Q) - y_t)^2],$$

where $y_t = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma Q(\mathbf{s}_{t+1}, \mu(\mathbf{s}_{t+1})|\theta^Q)$. The author, to make the DPG deeper and implement it through neural networks, has made several changes, e.g., to the replay buffer making it larger, improving the learning algorithm to avoid divergence, using the batch normalization technique and adopting a new policy $\mu'(\mathbf{s}_t) = \mu(\mathbf{s}_t|\theta_t^\mu) + \mathcal{N}$ built by introducing a noisy process $\mathcal{N}$.

- Twin-Delayed DDPG (**TD3**) [33], an evolution of DDPG method that solve the problem of reducing overestimation bias by introducing a novel clipped variant of Double Q-Learning and reduce high variance estimates minimizing error at each update. In this case, Fujimoto et al. [33] maintain the loss of the DDPG model but introduce a novelty in updating the pair of critics of the actions selected by the target policy, defining

$$y = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \min_{i=1,2} Q_{\theta_i'}(\mathbf{s}', \pi_{\phi'}(\mathbf{s}') + \epsilon)$$

with $\epsilon \sim clip(\mathcal{N}(0,\sigma), -c, c)$, where c is a constant and $clip(\mathcal{N}(0,\sigma), -c, c)$ clip the probability. These changes made it possible to increase the stability and performance with consideration of function approximation error.

- Proximal Policy Optimization (**PPO**) [66], another policy optimization method that maximize a surrogate objective function which indicates the variations of the $J(\pi_\theta)$ function at each update. In particular, Schulman et al. [66] develop a loss function that combines policy surrogate and value function error term. Starting from the clipped loss

$$J^{CLIP}(\theta) = \mathbb{E}[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t]$$

where π_θ is the stochastic policy, $\hat{A}_t$ is an estimator of the advantage function at time t, $r_t(\theta) = \frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\theta \text{old}}(\mathbf{a}_t|\mathbf{s}_t)}$, ϵ is an hyperparameter and $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t$ modifies the surrogate objective by clipping the probability ratio; the authors combined it with entropy bonus obtaining the following objective

$$J_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[J_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](\mathbf{s}_t)]$$

where c_1 and c_2 are coefficients, S is the entropy bonus and L_t^{VF} is a squared-error loss $(V_\theta(\mathbf{s}_t) - V_t^{targ})^2$ between state-value functions.

- Soft Actor-Critic (**SAC**) [44], another mixed method between Q-Learning and Policy Optimization that uses stochastic policies and entropy regularization to stabilize learning than DDPG. In this case, the Soft Actor-Critic algorithm start from a maximum entropy variant of the policy iteration method. According to [44], we know that $\mathbf{s}_t \in \mathcal{S}$ is the current state, $\mathbf{a}_t \in \mathcal{A}$ is an action, $V_\psi(\mathbf{s}_t)$ is the parameterized state value function, $Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$ is the soft Q-function and $\pi_\phi(\mathbf{s}_t, \mathbf{a}_t)$ is the tractable policy. The parameters are: ψ learned by minimizing the square residual error

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\frac{1}{2}(V_\phi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_{psi}}[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\psi(\mathbf{a}_t|\mathbf{s}_t)])^2\right],$$

where $\mathcal{D}$ is the distribution of previously sampled states and actions; θ learned by minimizing the soft Bellman residual

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}}\left[\frac{1}{2}(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t))^2\right],$$

with $\hat{Q}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p}[V_{\bar{\psi}}(\mathbf{s}_{t+1})]$ and $\bar{\psi}$ the exponentially moving average of the value network weights; and finally ϕ learned by minimizing the expected KL-divergence

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[D_{KL}\left(\pi_\psi(\cdot|\mathbf{s}_t)\middle\|\frac{\exp(Q_\theta(\mathbf{s}_t,\cdot))}{Z_\theta(\mathbf{s}_t)}\right)\right].$$

3.3 Neuro-fuzzy systems: ANFIS technical details

In this section, we provide basic technical details on adaptive neural fuzzy inference system (ANFIS) which is at the basis of our *GGSMZ* trader (see Sect. 5).

ANFIS was first proposed by Jang [47]. ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are derived from training examples. As a matter of example, we assume a FIS with two inputs x and y with two associated membership functions (MFs), and one output (z). For a typical first-order Takagi–Sugeno fuzzy model [70], a common rule set, with two fuzzy if-then rules, is presented as follows:

- Rule 1: if x is A_1 and y is B_1, then $f_1 = a_1x + b_1y + c_1$,
- Rule 2: if x is A_2 and y is B_2, then $f_2 = a_2x + b_2y + c_2$,

where A_1, A_2, B_1 and B_2 are the linguistic labels of the inputs x and y, respectively, and a_i, b_i, c_i $(i = 1, 2)$ are the parameters [47]. Figure 2a, b illustrate the reasoning mechanism and the corresponding ANFIS architecture, respectively [47].

As shown in Fig. 2b, ANFIS is a multilayer network. The operation of ANFIS model from layer 1 to layer 5 is briefly presented below [47].

- *Layer 1*: all the nodes in this layer are adaptive nodes, which indicate that the shape of membership function can be modified through training. Taking Gaussian MFs as an example, the generalized MFs are defined as follows:

$$O_i^1 = \mu_{Ai}(x) = e^{-\frac{(x-c_i)^2}{2\sigma_i^2}}$$

where x is crisp input to node i and Ai is the linguistic label, such as low, medium and high. O_i^1 is the membership grade of fuzzy-set Ai, which can be trapezoidal, Gaussian, bell-shaped and sigmoid functions or others. The variables (σ_i, c_i) are the parameters of the MF governing the Gaussian function.

- *Layer 2*: The nodes in this layer are gray circle nodes labeled Π, indicating that they perform as a simple multiplier. Each node output represents the firing strength of each rule.

$$O_i^2 = w_i = \mu_{Ai}(x) \times \mu_{Bi}(x),\ \ i = 1, 2$$

- *Layer 3*: the nodes in this layer are also gray circle nodes labeled N. The ith node is the ratio of the ith rule's firing strength to the sum of all rules' firing strengths. The outputs of this layer can be given by

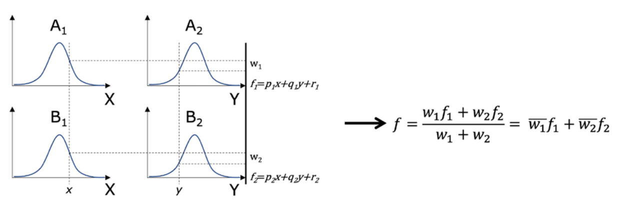
$$O_i^3 = \bar{w_i} = \frac{w_i}{w_1 + w_2},\ \ i = 1, 2$$

- *Layer 4*: each node i in this layer is adaptive. Parameters in this layer are considered as consequent parameters. The outputs of this layer can be represented as

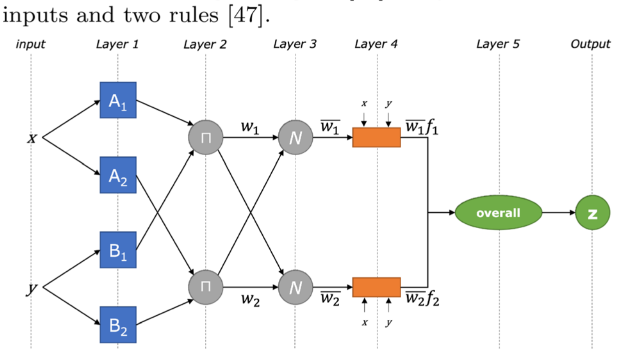
$$O_i^4 = \bar{w_i}f_i = \bar{w_i}(p_ix + q_iy + r_i),\ \ i = 1, 2$$

- *Layer 5*: the node in the last layer computes the overall output as the summation of all incoming signals. The overall output is given as

$$O_i^5 = z$$
$$= \sum_i i = 1^2 \bar{w_i}f_i = \frac{w_1(p_1x + q_1y + r_1) + w_2(p_2x + q_2y + r_2)}{w_1 + w_2}$$



(a) A first order Tagaki-Sugeno [70] fuzzy model with two inputs and two rules [47].



(b) The equivalent ANFIS architecture [47].

Fig. 2 ANFIS details

In the ANFIS architecture, the major task of the training process is to make the ANFIS output fit with the training data by optimizing the fuzzy rules and parameters of membership functions. The hybrid-learning algorithm incorporating gradient method and the least-squares are used in ANFIS to estimate the initial parameters and quantify the mathematical relationship between input and output. Further details are in [47, 70].

3.4 Crypto datasets

This work uses datasets that describe the evolution of the price of some of the most famous cryptocurrencies, Bitcoin and Ethereum, in different time frames.

BTC-USD 2018 dataset Regarding the Bitcoin price and its time-division, we have chosen as a ticker the BTC-USD price recorded by CoinMarketCap through Yahoo!Finance[2], and we have split it into 3 time frames. The entire dataset contains the prices from 01/01/2015 to 12/31/2018 (the first big bubble on this crypto) for a total of 1,460 days and consists of the classic OHLCV features used in financial sector: Open, High, Low, Close, and Volume. In Table 1, we show an extract of how the dataset is composed.

We know that the bubble in this crypto market, started on December 17, 2017, with the price of 1 BTC reaching around $20,000 and reaching its peak toward the end of January, marking the explosion point and causing the Bitcoin's price settlement in the following months. Based on these events, we created the first time frame[3], called **Before**, with training from 1/1/2015 to 2/28/2017 (543 days) and tests from 3/1/2017 to 12/15/2017 (221 days); the second time frame, called **During**, with training from 1/1/2015 to 12/15/2017 (745 days) and tests from 12/16/2017 to 5/31/2018 (112 days); and the third time frame, called **After**, with training from 01/01/2015 to 5/31/2018 (858 days) and tests from 6/1/2018 to 12/31/2018 (146 days). In Fig. 3, we plot the close prices of BTC during the period of interest.

BTC-USD 2021 dataset We consider again the BTC-USD price recorded by CoinMarketCap through Yahoo!Finance, but in the famous bubble of 2021. This situation became famous thanks to the incredible advancement of the Bitcoin price up to $60,000, which has brought many other cryptos to the fore. We consider the OHLCV dataset and perform the temporal division in the three intervals. However, here, a particular situation arises of two consecutive bubbles. We can create the first time frame, **Before**, with training from 3/1/2019 to 4/30/2020 (294 days) and tests from 5/1/2020 to 1/31/2021 (189 days); the second time frame, **During**, with training from 3/1/2019 to 1/31/2021 (484 days) and tests from 2/1/2021 to 7/31/2021 (126 days); and the third time frame, **After** (characterized by a new bubble), with training from 3/1/2019 to 7/31/2021 (610 days) and tests from 8/1/2021 to 12/31/2021 (106 days). In particular, in this situation, we have decided to use a much smaller dataset than the previous one (with much fewer days) to verify the capabilities of the different agents. In Fig. 4, we plot the close prices of BTC during the period of interest.

ETH-USD 2021 dataset We also consider the trend of the Ether cryptocurrency (differentiated from the previous one because it is based on the Ethereum Blockchain). In particular, it is known how the trend of Bitcoin also affects the other cryptocurrencies (including Ethereum), so we decided to consider the same situation as the previous dataset and analyze the two bubbles that occurred in 2021. Ether peaked at a price and broke the $4000 per ETH barrier. The ticker is ETH-USD, again from CoinMarketCap by Yahoo!Finance, and is the classic OHLCV dataset (as in the previous cases) for 1037 days. The time frames are constructed in this way by dividing the days as for the previous crypto: **Before** with training from 3/1/2019 to 4/30/2020 and tests from 5/1/2020 to 1/31/2021; **During** with training from 3/1/2019 to 1/31/2021 and tests from 2/1/2021 to 7/31/2021; and **After** with training from 3/1/2019 to 7/31/2021 and tests from 8/1/2021 to 12/31/2021. In Fig. 5, we plot the close prices of ETH during the period of interest.

Table 1 Extract of the BTC-USD price dataset

Date	Open	High	Low	Close	Volume
1/1/2015	320.43	320.43	314.00	314.24	8036550
1/2/2015	314.07	315.83	313.56	315.03	7860650
1/3/2015	314.84	315.14	281.08	281.08	33054400
1/4/2015	281.14	287.23	257.61	264.19	55629100
1/5/2015	265.08	278.34	265.08	274.47	43962800
⋮	⋮	⋮	⋮	⋮	⋮

4 Experiment

In this section, we analyze the behavior of agents with our setup (Sect. 4.1) in the different time phases, showing which are the best (Sect. 4.2) and proposing some recommendations to investors.

4.1 Experiment setup

From the *ZI/MI* agents side, referring to what Cliff introduced, we used 5 buyer agents and 5 sellers for each type (to have 10 agents every day). The simulation tool used to test the predictive power of the various agents is the Bristol Stock Exchange (BSE) [22]. In this limit-order-book financial exchange written in Python, agents are free to make their own trading strategies based on their intrinsic functioning. To make the operation more realistic has been developed a Python's multi-threading version, which allows traders to operate asynchronously of each other: the Threaded Bristol Stock Exchange (TBSE)[4]. In this TBSE that we have used, some parameters are extracted from the time series of the Bitcoin/Ethereum price (OHLCV

[2] https://finance.yahoo.com/quote/BTC-USD/.

[3] All dates are in US format.

[4] https://github.com/MichaelRol/Threaded-Bristol-Stock-Exchange.

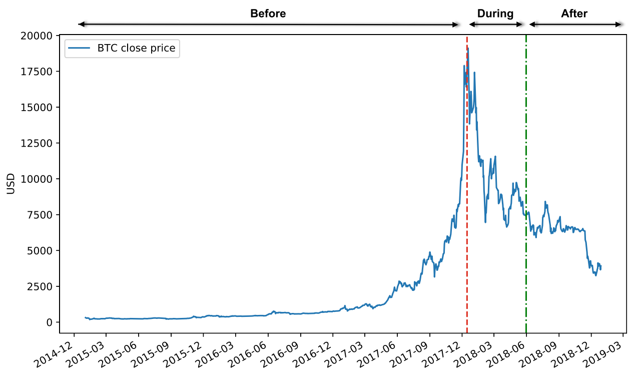


Fig. 3 BTC close prices over the period of interest (bubble 2018). **Before**: from start to the red dashed line. **During**: from the red dashed line to the green dot-and-dash line. **After**: from the green dot-and-dash line to the end

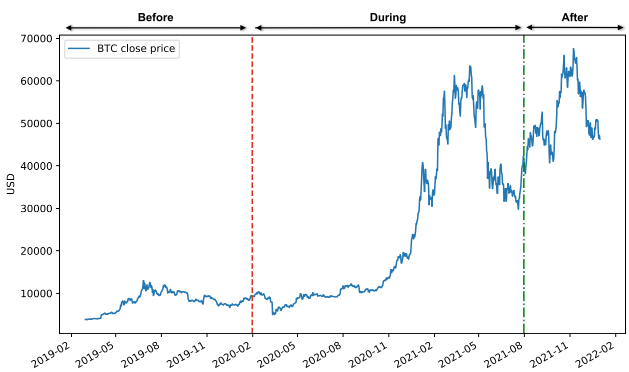


Fig. 4 BTC close prices over the period of interest (bubble 2021). **Before**: from start to the red dashed line. **During**: from the red dashed line to the green dot-and-dash line. **After**: from the green dot-and-dash line to the end

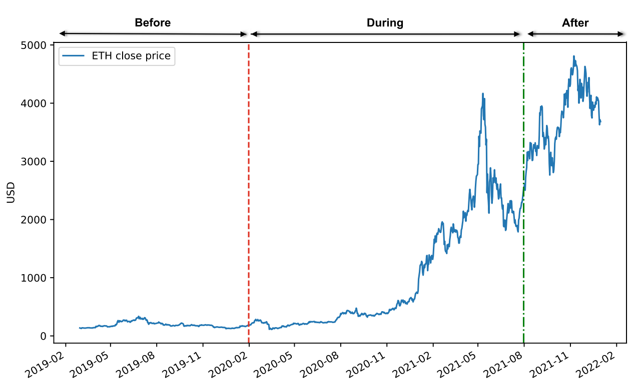


Fig. 5 ETH close prices over the period of interest (bubble 2021). **Before**: from start to the red dashed line. **During**: from the red dashed line to the green dot-and-dash line. **After**: from the green dot-and-dash line to the end

features in BTC-USD/ETH-USD datasets) that will serve to direct the exchanges between different 5 chosen agents: ZIC, ZIP, GDX, AA and GVWY. The key feature is that the returns obtained by the agents do not follow the actual price of cryptos but undergo variations according to the different situations in which the market finds itself (e.g., being inside a bubble or outside). Furthermore, these agents (by definition of the TBSE) can trade only one type of instrument (e.g., for each execution, they can trade only Bitcoin, or only Ethereum, and so on) and can only trade contracts of size 1.

On the *CI* side, on the other hand, some features representative of the traditional indicators use by financial analysts have been added to the dataset, e.g., moving average, convergence/divergence (MACD), relative strength index (RSI), smoothed moving average on the closing price at 30 and 60 days, commodity channel index, directional movement index and the Bollinger Band. Such indicators are reported in Table 2.

The *CI* agents were endowed with an initial capital equal to 20,000 price units for the BTC-USD 2018 dataset, 30,000 price units for the BTC-USD 2021 dataset (50,000 in the last two time frames given the exponential growth of the price) and 10,000 price units for the ETH-USD 2021 dataset; and based on this sum, they were able to manage it in the best possible way according to their rewards function. We have set the parameters of the different agents as shown in Table 3. These configurations are given by the authors in [55] and are those achieving the best results.

The comparison between the behaviors of the *ZI/MI* and *CI* agents takes place based on: (i) the cumulative returns on a daily basis, (ii)*the volatility of these, (iii) the Sharpe ratio, (iv) the max drawdown, (v) the Sortino ratio and (vi) the Omega ratio*. For cumulative returns, we can first consider the simple return r_t for one period as:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}},$$

where P_t and P_{t-1} represent the price value (of cryptos in these cases), respectively, at time t and $t-1$. Then, the cumulative return (or multiperiod) for n days is calculated as:

$$R_t(n) = \Pi_{i=1}^{n}(r_i + 1) = \\ (r_t + 1) \times (r_{t-1} + 1) \times \ldots \times (r_{t-n+1} + 1) - 1.$$

The *Sharpe ratio*, is defined as:

$$SR = \frac{r - r_f}{\sigma},$$

where r and σ indicate asset return and volatility respectively while r_f indicate the risk-free interest rate (set in pyfolio[5] $r_f = 0$). The Maximum Drawdown (MDD) represents the maximum loss of a trading capital for a certain

[5] pyfolio is a Python library for performance and risk analysis of financial portfolios developed by Quantopian Inc. At the core of pyfolio is a so-called tear sheet that consists of various individual plots that provide a comprehensive image of the performance of a trading algorithm. See more at https://github.com/quantopian/pyfolio.

Table 2 Features employed

Feature	Description	Formula
`Open`	Opening price recorded on the current day	n/a
`High`	Highest price recorded on the current day, up to the closing time	n/a
`Low`	Lowest price recorded on the current day, up to the time of closing	n/a
`Close`	Closing price recorded on the previous day	n/a
`Volume`	Volume of trades made on the current day	n/a
`Macd`	*Moving average convergence/divergence:* indicator based on the convergence and divergence of two exponential moving averages of closing prices, computed at 12 days ($\mathtt{EMA}_{12}$) and at 26 days ($\mathtt{EMA}_{26}$).	$\mathtt{EMA}_{12}$-$\mathtt{EMA}_{26}$
$\mathtt{boll}_{up}$	*Superior bollinger band:* indicator to represent the price and volatility of an **instrument**, using a 20-day *moving average* (`MA`) and the standard deviation (σ)	$\mathtt{MA}_{20} + \mathtt{K}\,\sigma$
$\mathtt{boll}_{dn}$	*Inferior bollinger band*: indicator that completes the previous one, using a 20-day (`MA`) and the standard deviation (σ).	$\mathtt{MA}_{20}$ - $\mathtt{K}\,\sigma$
$\mathtt{rsi}_{30}$	*Relative strength index*: indicator used to identify the oversold and overbought areas, highlighting the ideal timing to enter and exit the market; based on the `EMA` of the upward closing differences U over 30 days and the `EMA` of the closing downward differences D over 30 days.	$100\text{-}\frac{100}{1+\frac{\mathtt{EMA}_{30}(U)}{\mathtt{EMA}_{30}(D)}}$
$\mathtt{cci}_{30}$	*Crypto currencies index*: indicator that measure the growth and movement of the blockchain sector, tracking the 30 largest cryptocurrencies (called "stable coin"); based on the weight of the jth crypto W_j and the price of that jth crypto P_j as function of time.	$\sum_{j=1}^{30} W_j \frac{P_j(t)}{P_j(0)}$
$\mathtt{dx}_{30}$	*Directional movement index*: indicator that identifies in which direction the price of an asset is moving, in a period range of 30 days; based on the highest price H_t recorded in the tth day, the lowest price L_t, and two directional indicators, $DI_+ = \frac{\mathtt{MA}(H_t - H_{t-1}) - (H_t - H_{t-1})}{\text{Avg price}} \times 100$ and $DI_- = \frac{\mathtt{MA}(L_t - L_{t-1}) - (L_t - L_{t-1})}{Avg\,price} \times 100$.	$\frac{\lvert DI_+ - DI_- \rvert}{\lvert DI_+ + DI_- \rvert} \times 100$
$\mathtt{sma}_{30}$	*30-day simple moving average:* indicator calculated as a moving average over the period considered (in this case 30 days)	$\frac{\sum_{j=1}^{30}(\text{BTCprice}_j)}{30}$
$\mathtt{sma}_{60}$	*60-day simple moving average:* indicator calculated as a moving average over the period considered (in this case 60 days)	$\frac{\sum_{j=1}^{60}(\text{BTCprice}_j)}{60}$
`vix`	*Chicago board options exchange's volatility index*: it represents the stock's market expectation of 30-day volatility, based on the prices for puts $P(K)$ and calls $C(K)$ with strike K and $\tau = 30$ days to maturity, and the 30-day forward price on the S &P500	$\sqrt{\frac{2er\tau}{\tau}\left(\int_0^F \frac{P(K)}{K^2}\mathrm{d}K + \int_F^\infty \frac{C(K)}{K^2}\mathrm{d}K\right)}$
`turb`	*Turbulence*: indicator that uses 28 samples of `sma` prices from $\mathtt{sma}_{12}$ to $\mathtt{sma}_{120}$ to reveal unseen structure[1]	n/a

[1] https://www.tradingview.com/script/ZEYUY4gy-Turbulence/

Table 3 Parameters set for each *CI* agent. Parameters are set as in [55]

Agent	Parameters
A2C	$n\_steps = 5$, $ent\_coef = 1 \times 10^{-2}$, $learning\_rate = 7 \times 10^{-5}$, $total\_timesteps = 5 \times 10^4$
DDPG	$batch\_size = 128$, $buffer\_size = 5 \times 10^4$, $learning\_rate = 1 \times 10^{-4}$, $total\_timesteps = 3 \times 10^4$
PPO	$batch\_size = 128$, $n\_steps = 2048$, $ent\_coef = 1 \times 10^{-2}$, $learning\_rate = 2.5 \times 10^{-4}$, $total\_timesteps = 5 \times 10^4$
TD3	$batch\_size = 10^2$, $buffer\_size = 10^6$, $learning\_rate = 1 \times 10^{-3}$, $total\_timesteps = 3 \times 10^4$
SAC	$batch\_size = 128$, $buffer\_size = 10^6$, $learning\_rate = 1 \times 10^{-4}$, $learning\_starts = 10^2$, ent_coef = auto_0.1, $total\_timesteps = 6 \times 10^4$

period, from a peak to a trough of a portfolio value. It is calculated as:

$$MDD = \frac{\text{Maximum value} - \text{Minimum value}}{\text{Maximum value}}.$$

The *Sortino ratio* is a financial risk indicator. It uses the *Downside Risk* (DSR) to highlight how investors feel under pressure when they perform inadequately compared to the minimum acceptable. First, we can define the DSR as a measure of the downward deviation (similar to the standard deviation) of the yield from the minimum acceptable yield. In this way, the *Sortino ratio* is calculated as:

$$\text{Sortino} = \frac{R_p - r_f}{DSR},$$

where also, in this case, $r_f = 0$ represents the risk-free rate, while R_p is the expected return. Finally, the *Omega ratio* is a risk-return performance measure, is an alternative to the *Sharpe ratio*, and is calculated by creating a partition in the cumulative return distribution in order to create an area of losses and an area for gains, so that:

$$\Omega(\theta) = \frac{\int_{\theta}^{\infty}[1 - F(x)]\mathrm{d}x}{\int_{-\infty}^{\theta} F(x)\mathrm{d}x},$$

where $F(x)$ is the cumulative probability distribution function of returns and θ the target return. FinRL automatically returns all these indicators, and to choose the best agent we select the one with the highest *Sharpe* and the lowest *Drawdown*, *Sortino*, and *Omega*.

4.2 Experiment results

Based on the indicators defined above, we can compare the agents in the different situations and concerning various instruments. Our goal is to understand how they behave in particular market situations, i.e., just before, during and after a bubble. As a benchmark (also indicated in graphics as *daily_return*), we consider the same indicators calculated on the price series extracted from the dataset in the same reference period. In the following, we report the result of the experiments conducted on BTC2018 bubble (Sect. 4.2.1), BTC2021 bubble (Sect. 4.2.2), and ETH2021 bubble (Sect. 4.2.3). For each report, we first sketch the reference values for the benchmark during the test period (in a box fashion), then we show the results obtained by the *ZI/MI* and *CI* agents and analyze them. All the figures mentioned are available in 2. For further graphics, we refer the reader to https://bit.ly/3wrkwi7.

4.2.1 Bitcoin bubble 2018

In this section, we offer details on the results achieved by *ZI/MI* and *CI* agents on the Bitcoin bubble of 2018 in three market situations, i.e., before, during and after the bubble.

Before

In this phase, it is clear how volatility was so high because the Bitcoin price underwent a sharp price jump in the test period (caused by the bubble's bursting).

Reference values BTC2018 *Before*

In the first time period, the reference values of benchmark during the test period were:

- Annual returns: 1574.78%;
- Volatility: 76.12%;
- Sharpe ratio: 3.33;
- Max Drawdown: −35.50%;
- Sortino ratio: 5.89;
- Omega ratio: 1.82.

In Table 4, we show how the different agents behaved in the same period, reporting the assumed values of the parameters used as a comparison.

We start the analysis of the results from the *ZI/MI* agents. We know these agents do not follow the actual price trend of Bitcoin but absorb some parameters from the reference market; for this reason, the returns are entirely distant from the benchmark. In particular, a value that we can use to understand the market phase is volatility: the ZIC, for example, which does not pursue a specific trading strategy but only trades, has an high volatility value, but comparing it with the same value assumed in the different periods can be used as a tool to identify whether the market is in an expansion phase (e.g., before/during a bubble) or recessive (after a bubble). Furthermore, by comparing the different indicators, we can see that, in these circumstances, the best agent was the AA, which achieved the highest *Sharpe ratio* and the lowest *Sortino ratio* and *Omega ratio*. *CI* agents, on the other hand, are comparable to the benchmark since they follow the same price level. In particular, the agent carrying out the best trading strategy in this time frame was the A2C that managed to attain a return equal to 1557%, the highest *Sharpe ratio* from which follows the lowest *Omega ratio*, and *Sortino ratio*. At the same time, the Drawdown remained reasonably constant for the various agents. Furthermore, the A2C was the only agent to achieve a close return to the benchmark. In terms of volatility, the strategies of the different agents were quite similar, with average volatility $\bar{\sigma} = 84\%$. Also here, to test a market situation, we can compare the average volatility of this time frame with the following ones. In Fig. 11, the volatilities of the most representative agents for both categories are represented. The green line (indicated in the legend as Backtest) represents the trend of the agent's volatility while the gray line represents the volatility followed by the BTC-USD ticker. The explosion's point of volatility for *ZIC* in Fig. 12a occurs when the bubble on the Bitcoin market explodes, near October, so in subsequent periods prices will remain stable on average. *We could assume that these explosion points*

Table 4 Comparison between the different agents in the **Before** period for Bitcoin bubble of 2018

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	−64.635	27697.151	2.40	−99.631	102.82	11.20
ZIP	104.434	3357.188	2.27	−98.981	17.26	2.77
GDX	−46.210	1609.329	3.97	−92.765	13.35	2.39
AA	28.481	1478.105	4.26	−92.475	13.32	2.37
GVWY	−23.152	1762.643	3.88	−94.913	14.85	2.58

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	1557.399	88.146	4.09	−34.436	7.28	2.04
DDPG	1278.55	85.746	3.72	−35.888	7.90	2.12
PPO	1047.734	83.104	3.78	−35.054	7.98	2.13
TD3	1334.90	84.761	3.51	−35.675	7.92	2.12
SAC	1195.451	89.574	3.12	−36.386	8.01	2.15

(b) *CI* agents

indicate the end of a specific market phase, while the volatility value what this phase is: whether in or out of a bubble. Making an intra-category comparison, we can say that on the side of the *ZI/MI* agents the most significant are the ZIC, in terms of the explainability of the volatility (necessary to understand if the market is on the bubble phase or not) and AA; on the *CI* side, on the other hand, the agent with the best behavior was the A2C that outperforms the benchmark, followed by PPO and DDPG. Figure 12 shows the cumulative returns of the leading agents for this time phase.

During Here, we analyze the second time period. The reference values of benchmark during the test period are available in the following box.

Reference values BTC2018 *During*
– Annual returns: −61.25%;
– Volatility: 87.40%;
– Sharpe ratio: −1.23;
– Max Drawdown: −65.28%;
– Sortino ratio: −1.64;
– Omega ratio: 0.82.

As in the previous case, Table 5 shows the behavior of different agents in this time period.

Here, likewise the previous time period, we can study volatility to understand what the market phase is. In particular, it is again the *ZIC* (for *ZI/MI*) that is the most explanatory of volatility; this time, however, the volatility value is halved compared to the previous situation (**Before**), suggesting that something has happened on the markets. During this time frame, the different agents were trained taking into account the strong price increases that occurred during the first phase of the bubble. Unlike the previous case (in which the agents were not aware of the large price increases that would have occurred due to the triggering of the bubble), in this situation, the behavior of all agents is influenced by having already registered strong increases and declines, so that the bubble bursting phase has a lighter impact, especially since prices remain stable on average in the subsequent phase. On the *CI* side, the volatilities of the first four agents are even more similar, with an average volatility $\bar{\sigma} = 75\%$, while PPO has the lowest volatility (similar to the benchmark). We can also consider the behavior of the second best agent, who achieved excellent results in terms of all performance indicators: the DDPG. For what concerns the intra-category comparison, in this case, for the *ZI* agents the best behavior is that held by the AA agent, also in terms of *Sortino* and *Omega ratio*. On the *CI* side, however, despite the very similar behavior of the different agents, the winner is the DDPG, which obtains better results in terms of Drawdown, *Sharpe ratio*, *Sortino ratio* and return. Figure 13 shows the cumulative returns for these main agents.

After Finally, we can analyze the last time period. The reference values of benchmark during the last test period are available in the following box.

Table 5 Comparison between the different agents in the **During** period for Bitcoin bubble of 2018

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	110.010	11086.490	4.90	−98.930	73.18	8.02
ZIP	−18.914	1410.580	3.95	−93.882	12.19	2.30
GDX	−11.281	1474.995	3.59	−87.864	12.44	2.30
AA	−47.629	1062.85	3.58	−86.752	9.19	1.93
GVWY	−30.962	1571.142	3.55	−93.498	12.32	2.30

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	−58.634	88.280	−1.81	−67.485	−2.29	0.85
DDPG	−52.558	62.219	−0.90	−62.219	−1.18	0.77
PPO	−60.128	54.948	−1.41	−70.527	−1.79	0.77
TD3	−59.213	86.375	−1.76	−68.564	−2.01	0.83
SAC	−57.989	89.750	−1.83	−69.696	−2.30	0.88

(b) *CI* agents

Reference values BTC2018 *After*
– Annual returns: −47.96%;
– Volatility: 54.23%;
– Sharpe ratio: −1.17;
– Max Drawdown: −61.57%;
– Sortino ratio: −1.50;
– Omega ratio: 0.80.

Table 6 shows the behavior of different agents in this last time period.

This time period is characterized by the fact that agents have observed the entire bubble from birth to burst and must trade at a later stage. Here, the volatility of the benchmark is the lowest compared to the previous ones since prices have remained constant on average (or at least have not undergone abrupt changes over a day as in prior periods). Let us consider what happens to the different agents at this stage. The ZI/MI agents, in this phase, are characterized by having a different behavior from the previous periods as regards the cumulative returns. For example, the GDX, which has always found negative returns, obtains a positive return. On the other hand, the volatility of the ZIC decreased compared to the previous

Table 6 Comparison between the different agents in the **After** period for Bitcoin bubble of 2018

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	−64.385	7584.882	3.50	−98.969	97.28	11.00
ZIP	−64.285	2896.424	3.39	−99.425	11.77	3.57
GDX	148.506	4265.015	2.28	−94.172	21.84	3.35
AA	−17.092	1580.808	4.24	−94.520	13.88	2.45
GVWY	−24.490	1559.506	3.91	−98.674	15.37	2.38

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	−41.013	55.776	−1.36	−49.159	−1.78	0.74
DDPG	−36.251	44.901	−1.51	−49.647	−1.84	0.76
PPO	−46.970	49.301	−1.48	−52.079	−1.86	0.77
TD3	−39.443	46.855	−1.45	−49.821	−1.88	0.78
SAC	−43.492	52.249	−1.63	−53.889	−2.08	0.77

(b) *CI* agents

Table 7 Comparison between the different agents in the **Before** period for Bitcoin bubble of 2021

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	116.667	4558.301	3.97	−97.723	30.04	3.96
ZIP	−28.232	978.955	3.39	−85.451	8.56	1.88
GDX	−27.341	1163.953	3.66	−92.556	10.06	2.07
AA	−29.898	949.402	3.42	−96.743	8.39	1.87
GVWY	60.539	2685.954	2.59	−84.774	16.74	2.79

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	63.229	32.487	2.19	−21.693	3.67	2.00
DDPG	190.430	57.423	3.14	−23.198	5.28	2.07
PPO	161.165	44.592	3.11	−22.703	5.54	2.26
TD3	201.753	54.812	3.32	−26.534	5.45	1.99
SAC	254.428	61.592	3.07	−20.723	5.16	1.78

(b)*CI* agents

phases in line with the benchmark's. This leads us to think that the random agent can inform us about the market phase we are experiencing. *CI* agents also underwent a change in their behavior. In terms of volatility, we can observe how the DDPG agent got the lowest value, but the best behavior is the one followed by the A2C agent (despite not having obtained the highest return), as evidenced by *Sharpe*, *Sortino*, and *Omega ratio* (the closer they are to 0, the better their behavior). Furthermore, the average volatility in this frame is $\bar{\sigma} \approx 50\%$, so also *CI* agents follow the trend of volatility reduction in the phase following a bubble, further confirming the fact that this volatility movement indicator is handy. In terms of performance, following the best strategy of A2C agent, Fig. 14 shows the behaviors of the various agents mentioned.

4.2.2 Bitcoin bubble 2021

In this section, we offer details on the results obtained by *ZI/MI* and *CI* agents on the Bitcoin bubble of 2021 in three market situations, i.e., before, during and after the bubble.

Before

In this first time period, the reference values of benchmark during the test period are available in the following box.

Reference values BTC2021 *Before*

- Annual returns: 287.10%;
- Volatility: 51.56%;
- Sharpe ratio: 2.68;
- Max Drawdown: −25.40%;
- Sortino ratio: 4.37;
- Omega ratio: 1.63.

In the BTC2021 bubble, we can continue to use, for *ZI* agents, the volatility of the ZIC as an indicator of the pre/post-bubble phase. In this time frame, the recorded price of Bitcoin has undergone strong trends due to the ever-increasing use of cryptocurrencies and has begun its race to the top. On the side of the *ZI/MI* agents, as shown in Table 7, only the ZIC and the GVWY managed to get a positive return (from the extrapolation of various parameters), while the other agents obtained a negative return, as shown in Fig. 15. In particular, although chaotic, the behavior of the GVWY was proved more effective than that of the ZIC (considering only agents with positive returns given the expansionary phase of the market), achieving excellent results under all indicators. In the previous bubble of 2018 (same time frame), the best agent was AA. On the other hand, on the side of the *CI* agents, the A2C achieves a very different behavior from that of the opponents, starting the trading strategy late (compared to them) and obtaining a lower return, but which other indicators being equal as the best result. However, among the remaining four agents, the best strategy is the one followed by the SAC (as evidenced by the *Sharpe*, *Sortino*, and *Omega ratio* values).

During Observe that the second time period of 2021 deserves more attention. In such a period, the price of Bitcoin has reached a price never seen before and has continued its exponential run that began in the previous time frame. The reference values of benchmark during this test period are available in the following box.

Reference values BTC2021 *During*
– Annual returns: 25.93%;
– Volatility: 72.48%;
– Sharpe ratio: 0.80;
– Max Drawdown: −53.06%;
– Sortino ratio: 1.21;
– Omega ratio: 1.14.

Table 8 shows the behavior of different agents in this **During** time period.

As in the previous bubble, let us look at the volatility of the ZIC across the different time frames to understand the situation. The reduction of the last frame is evident, from which we can deduce that we are in a subsequent phase to the initial one (in fact in the *During*). Compared to the 2018 bubble, this situation in 2021 demonstrates as the volatility of the *ZI* agents (except for ZIC) is on average lower ($\bar{\sigma}_{2021} \approx 2000 \geq \bar{\sigma}_{2018} \approx 1300$). Regarding the behavior of such agents, four-fifths got a negative return (also opposite to that recorded in the benchmark). At the same time, only the GVWY achieved a positive return. For this reason, despite not having obtained better results in terms of the *Sortino*, *Omega*, and *Sharpe ratio*, based on the ratio between the return and the recorded variance, we can classify it as the best agent. For what concerns the *CI* agents, on the other hand, the PPO was the only one to get a negative return and very high volatility (even compared to the average of the different time frames), which allowed it to obtain a meager *Sharpe ratio* for example. However, having a return opposite to that showed in the benchmark, we cannot consider it among the best agents. Moreover, a very particular behavior is followed by the TD3 agent: it did not do any trading until a few days before the last date (7/31). Hence, various indicators such as the *Sortino* and the *Omega ratio* were not calculable. Therefore, we can state that the best *CI* agent was the SAC, having the lowest *Sharpe* and *Sortino ratios*. Figure 16 shows the returns of SAC and GVWY.

After Finally, we consider the last time frame of the Bitcoin 2021 bubble. The reference values of benchmark during this test period are available in the following box.

Reference values BTC2021 *After*
– Annual returns: 20.34%;
– Volatility: 54.77%;
– Sharpe ratio: 0.84;
– Max Drawdown: −31.62%;
– Sortino ratio: 1.22;
– Omega ratio: 1.14.

This period, however, is characterized by being at the same time the final phase of a bubble and the period of the bursting of a new one, which explains fairly high benchmark volatility. In addition, various news spreads on the markets and the ever-growing attention to crypto led to the follow-up of two (critical) bubbles in the same year. Table 9 shows the behavior of the different agents in this particular time frame.

Table 8 Comparison between the different agents in the **During** period for Bitcoin bubble of 2021

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	−47.978	3171.108	3.07	−98.634	19.22	2.99
ZIP	−54.837	1147.853	3.70	−93.500	9.87	2.05
GDX	−22.487	1681.030	4.36	−94.598	14.73	2.55
AA	−63.790	2418.458	3.63	−96.254	18.12	2.88
GVWY	26.208	1892.337	3.81	−92.536	15.43	2.63

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	17.330	60.579	0.83	−42.156	1.22	1.15
DDPG	14.441	51.941	0.78	−40.468	1.19	1.14
PPO	−14.891	71.996	−0.09	−52.797	−0.12	0.98
TD3	41.328	22.545	3.20	0	*NaN*	*NaN*
SAC	12.871	49.756	0.81	−37.236	1.14	1.15

(b) *CI* agents

Table 9 Comparison between the different agents in the **After** period for Bitcoin bubble of 2021

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	20.967	3076.271	4.19	−96.704	19.47	2.96
ZIP	−55.497	1176.584	3.50	−91.867	9.87	2.08
GDX	−15.214	1587.444	4.11	−90.415	13.79	2.42
AA	−34.059	1647.026	2.76	−95.668	11.81	2.18
GVWY	68.982	1615.411	3.89	−90.356	13.86	2.47

(a) *ZI/MI* agents

CI **agent**	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	15.874	51.682	0.94	−27.277	1.49	1.16
DDPG	19.842	61.821	1.01	−31.269	1.61	1.17
PPO	10.740	37.455	0.84	−24.189	1.39	1.18
TD3	16.756	56.886	0.99	−26.982	1.50	1.17
SAC	13.228	44.441	0.89	−24.606	1.41	1.15

(b) *CI* agents

The first aspect we can observe is how the volatility of the ZIC has slightly decreased compared to the time frame during (always 2021) but has remained almost constant. This means that the exit from a bubble has not been completed (as in the present case due to the entry into a new bubble). The GVWY has the best behavior among the *ZI* agents since the others obtained a negative return, opposite to the benchmark. From reading the additional indicators, it may seem like ZIP or AA are the best, but these values are due to the ratios between yield and volatility, which is not in line with what they should have achieved. However, among the *CI* agents, the two best behaviors were those of the SAC and PPO, which achieved the best values of *Sharpe* and *Sortino* (*Omega ratio* is, on average, stable among all). In Fig. 17 it is possible to see the returns' behavior of GVWY and PPO.

4.2.3 Ethereum bubble 2021

In this section, we offer details on the results obtained by *ZI/MI* and *CI* agents on the Ethereum bubble of 2021 in three market situations, i.e., before, during and after the bubble.

Before Here, we analyze the performance of the Ethereum in the **Before** period. The reference values of benchmark during this test period are available in the following box.

Reference values ETH2021 *Before*
– Annual returns: 54.53%;
– Volatility: 72.98%;
– Sharpe ratio: 2.72;
– Max Drawdown: −32.68%;
– Sortino ratio: 4.44;
– Omega ratio: 1.62.

Already graphically (the price plot 5), it is possible to see how the ETH bubble is very similar to the BTC one, but on a different price level. Table 10 shows the results obtained by the different agents. We continue the volatility analysis based on the ZIC. Compared to the same time frame of previous crypto (i.e., **Before** BTC2021), in this instance, ZIC agent experienced higher volatility that is in line with the benchmark average value. We can consider the GDX as the agent with the best behavior among the *ZI* agents. Therefore, we can exclude the agents with negative returns (opposite the benchmark). Among the remaining ones, even if the GDX does not have the highest *Sharpe ratio*, it is the agent with the lowest *Sortino* and *Omega ratio*. On the other hand, the *CI* agents all achieved a much higher return than the benchmark and recorded a very high level of volatility. Therefore, the best performing agent is the PPO (the highest *Sharpe* and the lowest *Sortino* and *Omega ratio*). Figure 18 shows the returns of GDX and PPO.

During We now consider the next time frame. The reference values of benchmark during this test period are available in the following box.

Table 10 Comparison between the different agents in the **Before** period for Ethereum bubble of 2021

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	512.010	6833.848	5.59	−98.848	54.05	6.30
ZIP	−14.733	1185.273	3.86	−90.911	10.35	2.11
GDX	23.080	1514.032	3.68	−88.579	12.52	2.32
AA	161.346	1942.896	3.73	−98.111	15.83	2.64
GVWY	−59.004	1983.383	3.91	−90.906	15.67	2.65

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	537.320	99.371	2.97	−34.486	5.70	1.78
DDPG	487.230	98.564	2.96	−33.232	5.67	1.77
PPO	510.023	98.592	2.97	−32.218	5.61	1.77
TD3	540.241	99.610	2.93	−32.566	5.70	1.78
SAC	218.380	89.776	2.15	−33.874	4.13	1.69

(b) *CI* agents

Reference values ETH2021 *During*
– Annual returns: 80.19%;
– Volatility: 96.37%;
– Sharpe ratio: 1.34;
– Max Drawdown: −57.12%;
– Sortino ratio: 2.00;
– Omega ratio: 1.25.

Table 11 shows the behavior of the different agents in the considered situation. Observing the volatility of the ZIC, we can see that this is down by about 20% compared to the previous time frame, so we can believe that we have entered a new bubble phase. Furthermore, compared to the time frame *During* of Bitcoin 2021, the average volatility of these agents is higher, as also highlighted by the benchmark. As for the best *ZI* agent, we can say that the best was the AA, followed by the ZIP; since GDX and GVWY got an inverse return compared to the benchmark, the ZIC has extremely high *Sortino* and *Omega ratios*. Instead, for *CI* agents, the first noticeable thing is the weird behavior of the PPO agent. It performs few transactions and attains a reasonably satisfactory result, but this is not enough to be ranked as the best agent due to the very high

Table 11 Comparison between the different agents in the **During** period for Ethereum bubble of 2021

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	500.767	5589.721	5.63	−99.281	40.58	5.35
ZIP	169.291	2030.170	4.02	−94.027	16.84	2.73
GDX	−1.244	1620.966	3.50	−93.365	13.05	2.30
AA	62.575	1607.796	4.47	−92.308	14.50	2.53
GVWY	−31.037	1404.838	3.97	−93.523	12.23	2.28

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	76.759	109.762	1.63	−56.335	2.33	1.31
DDPG	71.333	107.435	1.59	−56.657	2.37	1.33
PPO	61.240	61.591	1.86	−24.576	3.36	1.66
TD3	68.251	102.22	1.57	−57.434	2.38	1.33
SAC	79.749	112.282	1.61	−57.058	2.37	1.32

(b) *CI* agents

Sortino ratio. In view of this, it is evident how intelligent agents have been affected by the high volatility recorded (benchmark) since (except for the PPO) they have volatility higher than 100%, much higher than that recorded in the same time frame of Bitcoin 2021. Among these, the best agent was A2C, with good results on all the various indicators. Figure 19 shows the returns of AA and A2C.

After Finally, we can consider the last time frame for Ethereum. The reference values of benchmark during this test period are in the following box.

Reference values ETH2021 *After*

- Annual returns: 42.28%;
- Volatility: 67.77%;
- Sharpe ratio: 1.21;
- Max Drawdown: −30.05%;
- Sortino ratio: 1.84;
- Omega ratio: 1.21.

Table 12 shows the results of the different agents. Also, as for the Bitcoin 2021 bubble, the time frame **After** represents a second bubble, as evidenced by the volatility of the ZIC very close to the previous time frame. Among the *ZI* agents, the best result is attained by the ZIP; while for the *CI* agents, again, the PPO made few transactions (as in the **During**), but its behavior did not lead to good results. The best agent was the A2C again. Figure 19 shows the returns of ZIP and A2C.

4.2.4 Summarization and some recommendations for investors

From the results shown above, it is natural to ask ourselves which are the best agents to use to understand what market phase we are in and, consequently, which strategy to follow. A first answer could be that the best trading strategies are those of *CI* agents: Even if true as an answer, it must be said that these agents arise from a learning process complex and deep. Although these agents can follow the real price trend in the strategy and very often perform better than a human trader can do only with his own considerations, they lose in terms of explainability. Conversely, however, the *ZI/MI* agents are not able to follow the actual price trend of the asset considered but are entirely explainable in economic terms, and the previous results allow us to state that they can guess in which market phase we are (as before, thanks to volatility). The result of the high explainability is not to be underestimated.

For example, in Table 13 we report the volatility values recorded by the ZIC in the different time frames and for the different cryptocurrencies. The use of the ZIC agent lies in the fact that strategies do not influence its buying/selling activities, as is the case for other *ZI/MI* agents (albeit minimal). In this way, it is possible to notice the decrease in volatility in the passage from one frame to the next and the particular situation of 2021 in which the two frames of during and after having similar volatility (a symptom of the succession of two bubbles). Instead, with regards to the behavior of the different agents, in Table 14, we can summarize, for each time frame, the agents that have achieved the best results for the *ZI/MI* and the *CI*.

Table 12 Comparison between the different agents in the **After** period for Ethereum bubble of 2021

ZI/MI agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
ZIC	−27.455	5020.559	5.21	−97.940	45.02	5.02
ZIP	48.871	1031.884	3.91	−86.371	9.56	2.06
GDX	−7.636	1512.437	3.30	−93.910	11.87	2.23
AA	−17.878	1104.778	3.84	−87.161	10.03	2.04
GVWY	90.656	1190.669	3.82	−91.943	10.88	2.11

(a) *ZI/MI* agents

CI agent	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
A2C	33.033	65.566	1.37	−22.234	2.12	1.24
DDPG	34.504	73.202	1.32	−27.995	2.23	1.20
PPO	−2.057	52.134	0.16	−25.955	0.23	1.03
TD3	40.040	75.986	1.44	−28.843	2.25	1.25
SAC	36.703	71.128	1.41	−27.177	2.19	1.25

(b) *CI* agents

Table 13 Summary of volatilities recorded by ZIC agent

Dataset	Volatilities (%)		
	Before	During	After
BTC-USD2018	27697.151	11086.49	7584.882
BTC-USD2021	4558.301	3171.108	3076.271
ETH-USD2021	6833.848	5589.721	5020.559

Table 14 Best agents (*ZI/MI*, *CI*) in different time frames

Dataset	Before	During	After
BTC-USD2018	AA, A2C	AA, DDPG	GDX, A2C
BTC-USD2021	GVWY, SAC	GVWY, SAC	GVWY, PPO
ETH-USD2021	GDX, PPO	AA, A2C	ZIP, A2C

It is evident that some agents are present more often than others (e.g., the TD3, which has never been the best agent). However, the *CI* agents generally have a more realistic and benchmark-compliant behavior than the *ZI/MI*.

What we can recommend to investors is the following "rule": If he/she intends to follow a "machine-based" strategy that is highly performing but which he is not aware of and which he may not fully understand, then the best choice is to opt for a *CI* agent; however, if the investor already has his own strategy that he/she intends to follow and wants to understand what market phase is (to adjust it accordingly), then the best choice is to use a *ZI/MI* agent. It often happens that investors do not have a real strategy, but are based on some simple economic principles which (in several cases) are the same ones that govern *ZI/MI* agents. In these cases, the ideal choice is to use the intuition in the market phase of this type of agent and try to imitate (within the possible price limits) their strategy.

5 *GGSMZ*: a neuro-fuzzy trading agent

In this section, we present and detail *GGSMZ*, a neuro-fuzzy trading agents that we developed in the light of the results obtained above. First, we show the methodology adopted to build the neuro-fuzzy systems at the basis of our *GGSMZ* trading agent (Sect. 5.1). Lastly, we present the implementation of *GGSMZ* and its pseudo-code (Sect. 5.3), and the results obtained when *GGSMZ* operates during the different time frames (Sect. 5.4).

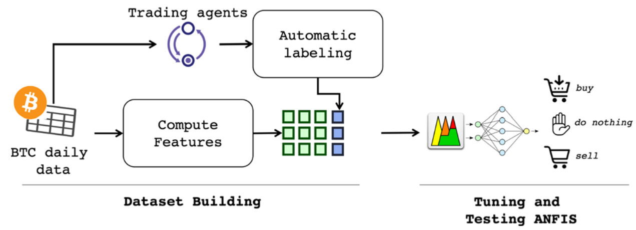


Fig. 6 The methodology adopted for building the neuro-fuzzy system at the basis of GGSMZ trading agent

5.1 Methodology: building a neuro-fuzzy system

To build our neuro-fuzzy system, we defined a methodology consisting of the two following steps (see Fig. 6):

1. *dataset building* (Sect. 5.1.1): it involves the use of various datasets previously presented (Sect. 3.4) with the integration of new features computed for the samples, and a (automatic) labeling process based on a criterion defined through the *CI* agents output, as well as other preprocessing steps;
2. *tuning and testing of ANFIS* (Sect. 5.2): it includes evaluating different ANFIS configurations to find the most suitable one for our problem and testing it on real-world financial bubble data.

5.1.1 Dataset building

The dataset building phase was made for each dataset defined in Sect. 3.4, i.e., BTC-USD2018, BTC-USD2021 and ETH-USD2021. For simplicity, we will only refer to BTC-USD, but the process has been repeated since they have the same features.

To create the dataset used to train and test the proposed neuro-fuzzy system, hereinafter `fuzzyds`, we have relied on the BTC-USD dataset presented in Sect. 3.4, which describes the Bitcoin price in USD over three years period. Each sample of `fuzzyds` is represented with the set of features taken from the BTC-USD dataset (i.e., OHLCV features) augmented with a set of handcrafted features (economic indicators) that are summarized in Table 2.

Formally, let $\texttt{s}_i = \langle \texttt{open}_i, \texttt{high}_i, \texttt{low}_i, \texttt{close}_i, \texttt{volume}_i \rangle$ be the *i*th sample in the BTC-USD dataset, with $i > 0$. We define a sample $\texttt{s'}_i$ to be inserted in `fuzzyds` as follows: $\texttt{s'}_i = \langle \texttt{open}_i, \texttt{high}_i, \texttt{low}_i, \texttt{close}_{i-1}, \texttt{volume}_i, \texttt{macd}, \texttt{boll}_{up}, \texttt{boll}_{dn}, \texttt{rsi}_{30}, \texttt{cci}_{30}, \texttt{dx}_{30}, \texttt{sma}_{30}, \texttt{sma}_{60}, \texttt{vix}, \texttt{turb} \rangle$.

Table 2 provides details about the features engineered to build $\texttt{s'}_i$. We remark that, if $i = 0$ the feature $\texttt{close}_{i-1}$ has been taken from the BTC-USD dataset regarding the year 2014 (on 12/31/2014).

Given neuro-fuzzy systems learn in a supervised fashion, we needed to label the samples of fuzzyds. Informally, the labeling has been conducted by exploiting the outputs of the *CI* agents (i.e., their operations on the market) and assigning to each sample the most common operation performed by such agents among *selling*, *buying* and *waiting*. Formally, let $\mathtt{o}_i = \langle \mathtt{A2C}_i, \mathtt{DDPG}_i, \mathtt{PPO}_i, \mathtt{TD3}_i, \mathtt{SAC}_i \rangle$ be the *i*th output obtained by collecting each *CI* agent's output when processing $\mathtt{s}_i$. We remark that the *CI* agents' outputs are in $\mathbb{Z}$. Therefore, we translate such outputs into classes, i.e., *0, 1, 2* to express the operations *waiting*, *selling* and *buying*, respectively, through the following formula:

$$\mathtt{CIagent}'_i = \begin{cases} 0 \text{ if } \mathtt{CIagent}_i = 0 \\ 1 \text{ if } \mathtt{CIagent}_i < 0 \\ 2 \text{ if } \mathtt{CIagent}_i > 0 \end{cases} \tag{1}$$

where $\mathtt{CIagent} \in \{\mathtt{A2C}, \mathtt{DDPG}, \mathtt{PPO}, \mathtt{TD3}, \mathtt{SAC}\}$. We obtain $\mathtt{o}'_i = \langle \mathtt{A2C}'_i, \mathtt{DDPG}'_i, \mathtt{PPO}'_i, \mathtt{TD3}'_i, \mathtt{SAC}'_i \rangle$. Lastly, the label for $\mathtt{s}'_i$, that is l_i, is computed as the most common operation performed by *CI* agents, denoted with $l_i = majority(\mathtt{o}'_i)$.

For example, let $\mathtt{A2C}_i = -63$, $\mathtt{DDPG}_i = +2$, $\mathtt{PPO}_i = +25$, $\mathtt{TD3}_i = +33$, and $\mathtt{SAC}_i = 0$, then $\mathtt{o}_i = \langle -63, +2, +25, +33, 0 \rangle$. By applying 1 we obtain $\mathtt{o}'_i = \langle 1, 2, 2, 2, 0 \rangle$, and $|\mathtt{o}'_i|_2 = 3$, $|\mathtt{o}'_i|_1 = 1$, and $|\mathtt{o}'_i|_0 = 1$. Then, since $|\mathtt{o}'_i|_2 > |\mathtt{o}'_i|_1$ and $|\mathtt{o}'_i|_2 > |\mathtt{o}'_i|_0$, we set $majority(\mathtt{o}'_i) = 2$, i.e., *buying*. Let us consider another example. Let $\mathtt{A2C}_i = -13$, $\mathtt{DDPG}_i = +22$, $\mathtt{PPO}_i = +25$, $\mathtt{TD3}_i = 0$, and $\mathtt{SAC}_i = -41$, then $\mathtt{o}_i = \langle -13, +22, +25, 0, -41 \rangle$. By applying 1 we obtain $\mathtt{o}'_i = \langle 1, 2, 2, 2, 0 \rangle$, and $|\mathtt{o}'_i|_2 = 2$, $|\mathtt{o}'_i|_1 = 2$, and $|\mathtt{o}'_i|_0 = 1$. Then, in this case it is not possible to find $x \in \{0, 1, 2\}$ such that $|\mathtt{o}'_i|_x > |\mathtt{o}'_i|_y$, for each $y \in \{0, 1, 2\}$ and $y \neq x$. So, we set $majority(\mathtt{o}'_i) = 0$, i.e., *waiting*.

We are aware that the number of rules of ANFIS exponentially grows with the number of inputs (via grid partitioning, i.e., the widely adopted method for FIS generation). Indeed, when we tried to develop our neuro-fuzzy model using fuzzyds as it is, i.e., with 15 features, it would have exceed the available RAM on our system[6] (16GB RAM). To avoid this problem, and reduce the feature space, we performed a preprocessing step via principal component analysis [1] (PCA) on fuzzyds. PCA reduces the number of variables while maintaining the majority of the important information. It transforms a number of variables that may be correlated into a smaller number of uncorrelated variables, known as principal components. The principal components are linear combinations of the original variables weighted by their variances in a particular orthogonal dimension. The main objective of PCA is to simplify the model features into fewer components to help the model run faster. Using PCA also reduces the chance of overfitting by eliminating features with high correlation.

We use explained variance ratio as a metric to evaluate the usefulness of our principal components and to choose how many components to use in the neuro-fuzzy system. The explained variance ratio is the percentage of variance that is attributed by each of the selected components. We chose the components to include in our model by adding the explained variance ratio of each of them until we reached a total of 0.90. By applying PCA on fuzzyds, we obtained that 4 components ($\mathrm{PCA}(\mathtt{s}'_i) = \langle \mathtt{input1}_i, \mathtt{input2}_i, \mathtt{input3}_i, \mathtt{input4}_i \rangle$) were enough to explain 90% of the information.

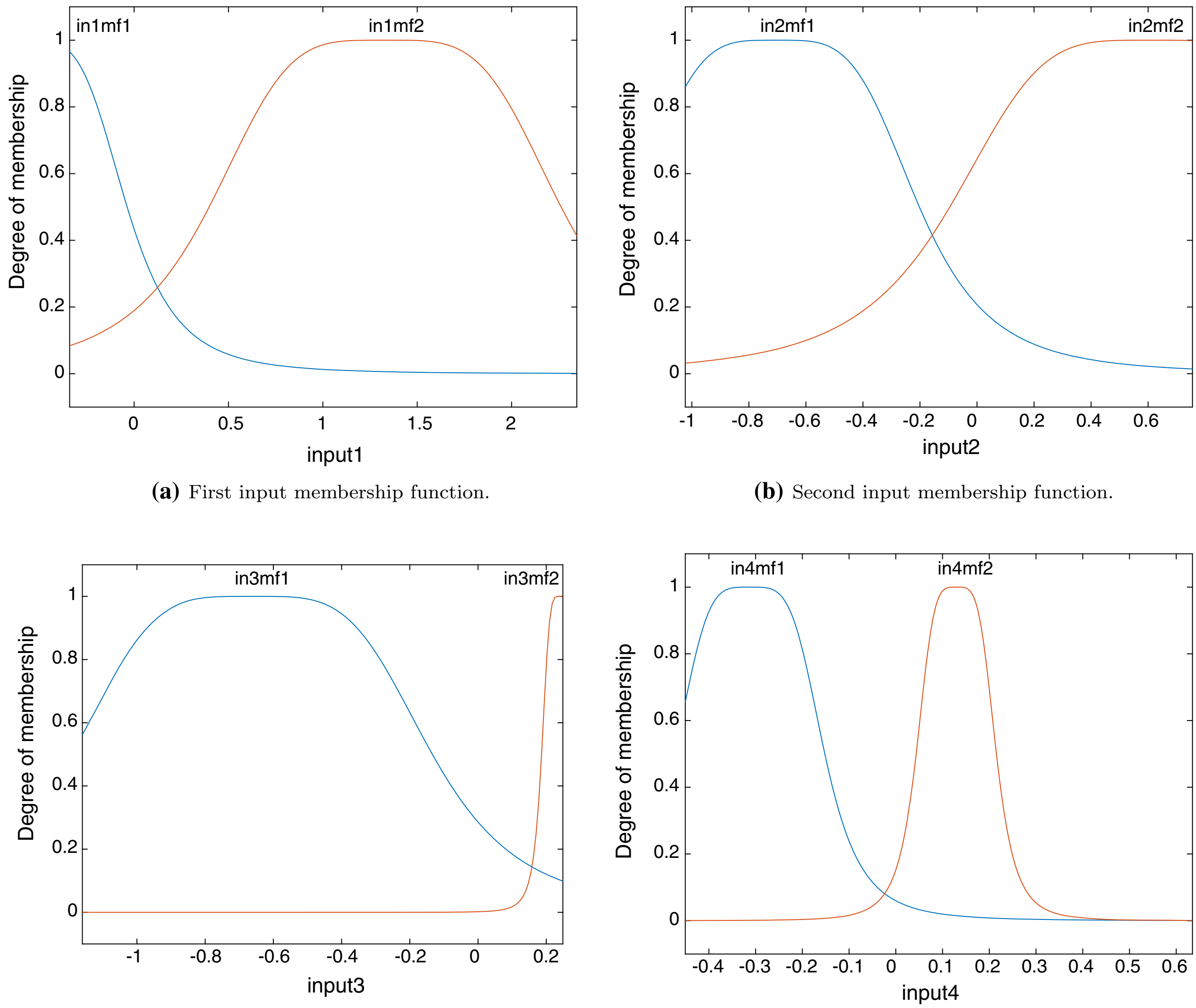
5.2 Tuning and testing ANFIS

In the present paper, to identify the Sugeno-type fuzzy inference systems parameters [70], we use a hybrid learning algorithm. Additionally, the proposed system produces fuzzy logic rules. ANFIS combines the least-squares and back-propagation gradient descent method to train FIS membership function parameters to emulate the given input on output. Thus, it is a very powerful, computationally efficient tool to handle imprecision and nonlinearity.

The fuzzy system component of ANFIS is mathematically expressed in the form of membership functions that are continuous and differentiable piecewise. These functions transform the input value x into a membership degree (i.e., a value between 0 and 1).

To define the optimal fuzzy rules' number, trial-and-error method was used considering various rule numbers, i.e., $r = \{2, 3, 4, 5\}$. They have been trained for each input and then evaluated based on system error and more precisely based on mean squared error (MSE), computed as $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ and root mean squared error (RMSE), computed as $\mathrm{RMSE} = \sqrt{MSE}$. (The objective was to minimize RMSE.) It should be noted that the lowest number of rules is always of interest in designing neuro-fuzzy models. Therefore, according to the obtained results, each input (in each ANFIS model) with 2 rules receives better performance prediction as compared to others, i.e., 3, 4 and 5. Hence, each ANFIS model with total fuzzy rules of $(2 \times 2 \times 2 \times 2)$ or 16 was designed and developed for prediction of trading actions (selling, buying, waiting) on the Bitcoin market. We applied the Gaussian fuzzy membership function, which has been commonly adopted in the

[6] Our system is equipped with 16 GB RAM and 2,8 GHz Intel Core i7 quad-core.



(a) First input membership function.

(b) Second input membership function.

(c) Third input membership function.

(d) Fourth input membership function.

Fig. 7 Post-training membership functions shape

Table 15 Main information about the ANFIS tuned and developed

ANFIS	
# nodes	55
# linear parameters	80
# nonlinear parameters	24
# fuzzy rules	16

literature [3, 4, 65], in the modeling process. Additionally, the type of output membership function was considered as linear one. In Table 15, we summarize main information of our ANFIS. These settings hold true for all the experiments with ANFIS.

For example, for BTC2018 (and in particular for the experiment concerning the **Before** phase) we trained the ANFIS on 543 samples (that is 543 days), while the testing was performed on 221 samples (that is 221 days) concerning a time frame just prior the bursting of the financial bubble (as reported in Sect. 3.4). This period is suggested as a benchmark period for evaluating a trading strategy, and it is similar to the one used in other studies [5, 7, 9, 67]. This constitutes a strong test for the forecasting ability of the model; however, at the same time, it provides a true assessment of its ability to work as if it were being applied in practice by an investor[7].

The optimal setting for ANFIS, in this case, was reached after 48 epochs. We obtained training $MSE = 0.119$ and $RMSE = 0.346$[8], while for testing we got $MSE = 4.888$ and $RMSE = 2.210$. For all other ANFIS results, we refer the reader to Appendix 3.

Figures 7 and 8 illustrates the shapes of the post-training membership functions. The graphics are for each of the four inputs. In addition, we depict in Fig. 9 an overview on the developed ANFIS and its structure.

5.3 Inside *GGSMZ* trader

The ANFIS model built in the previous section has been put at the basis of a trading agent, namely *GGSMZ*. This trader's pseudo-code is available in Algorithm 1. Every day *i* the *GGSMZ* agent takes one action on the market based on different parameters; it takes as input the *balance* (i.e., money available), the *total* (i.e., the profit generated by BTC/ETH in possession plus the balance), *cfolio* (i.e., number of BTC/ETH it has got), and *daydata* (that is the BTC/ETH data for the day *i*, hence $\mathtt{s}_i$). It computes the features shown in Table 2 (lines 1:2), creating the sample $\mathtt{s}'_i$. It then applies PCA on the sample, loads the ANFIS model and predicts the suitable action to perform (lines 3:5): If the suggested action is *buying*, *GGSMZ* buys as much BTC/ETH as possible with its current *balance* (line 7); if the suggested action is *selling*, *GGSMZ* sells all the BTC/ETH available in its *cfolio* (line 12); otherwise the agent *waits* without performing transactions. Lastly, *total* is computed and returned together with the current *balance*, and *cfolio* (lines 16:17).

Algorithm 1: GGSMZ pseudo-code.

```
   Input  : balance, total, cfolio, daydata
   Output: balance, total, cfolio
1  features ← computeFeatures(daydata);
2  sample ← [daydata features];
3  samplePCs ← myPCA(sample);
4  anfis ← loadANFIS();
5  action ← anfis.predict(samplePCs);
6  avgCrypto ← averageCryptoprice(daydata);
7  if action≥2 then
       // buy as many Cryptos as possible
8      nbtc ← ⌊balance/avgCrypto⌋;
9      cfolio ← cfolio + nbtc;
10     balance ← balance % avgCrypto;
11 else
12     if 1≤action < 2 then
           // sell all Cryptos
13         balance ← balance + cfolio × avgCrypto;
14         cfolio ← 0;
15     else
           // do nothing
16 total ← cfolio × avgCrypto + balance;
17 return balance, total, cfolio;
```

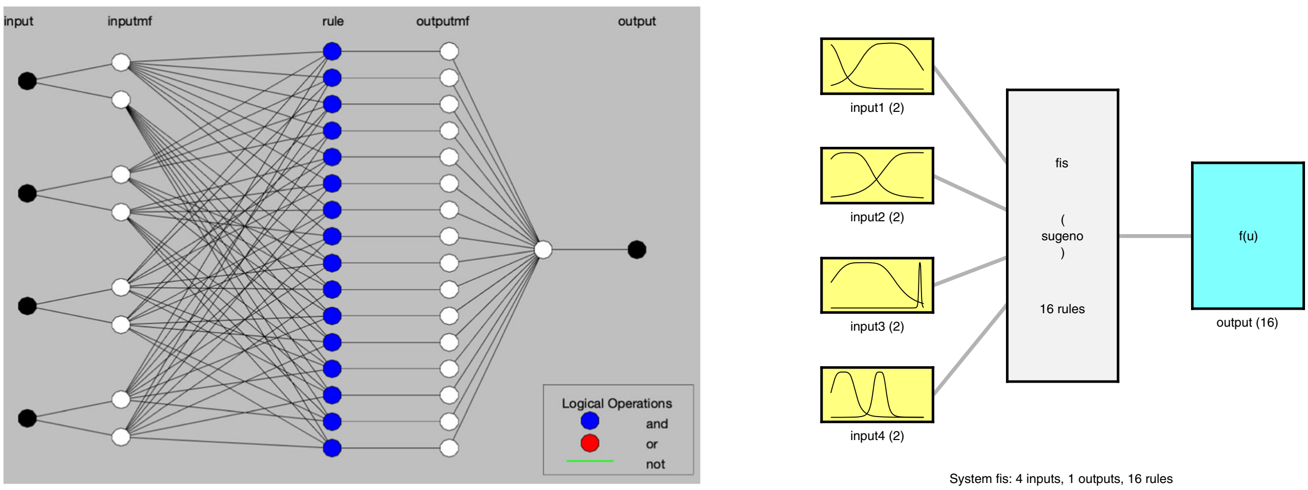
5.4 Discussion

We can evaluate the behavior of *GGSMZ* in the different time frames, as done previously for all agents, and compare the results with them. As for other experiments, we gave *GGSMZ* 20,000 USD as starting balance for BTC-USD 2018, 30,000 USD for BTC-USD 2021 and 10,000 USD for ETH-USD 2021. Using the indicators described above *(Returns, Volatility, Sharpe, Drawdown, Sortino and Omega ratio)*, our neuro-fuzzy agent achieved the following results, shown in Table 16 and then comparing them with the benchmark and agents with best behavior in Table 17 (i.e., the best ones described in Table 14).

The first evident result is that *GGSMZ* has attained a positive return in any time frame, regardless of the benchmark result. This result is certainly justified by the fact that *GGSMZ* bases its market entry/exit decisions on those made by *CI* agents. In this way, following what in financial markets is often defined as **herd behavior**, our agent was able to exploit the decisions of the other agents and choose the most widespread on the market (in this case, represented by the five *CI* agents). Therefore, its training enables it to make the best operational choices. Furthermore, a second interesting result is how the volatility changes in the different time frames: In all the datasets, *GGSMZ* follows the rule we have used so far, demonstrating how the progressive reduction of volatility marks the phase change (of the bubble) from **Before** to **After**. As for the agent's behavior with the different crypto, the highest avg. *Sharpe ratio* ($Sharpe_{avg} = 2.9$) and the lowest avg. Drawdown ($Drawdown_{avg} = -36.22\%$) are the

[7] We may also consider the use of a cross-validation approach. This approach has certain advantages over a simple split of the sample in training and validation datasets when the sample is rather small. In the present study, we have not followed this approach: using a cross-validation approach would not allow us to take time into account. Using a cross-validation approach would not allow us to split the sample while taking the whole time dimension into account. To better elucidate, suppose we split into 5-folds—f_1, f_2, f_3, f_4, f_5—which take time into consideration so that f_i contains observations which are taken in a previous period than f_j $\forall i < j$. During cross-validation iterations, we will find f_2, f_3, f_4, f_5 as training set and f_1 as validation set, thus a situation where we train the model on future observations compared to f_1 ones.

[8] We have further split the training set, reserving 20% of observations as validation set and computed MSE and RMSE on it.



(a) ANFIS structure proposed in this study.

(b) ANFIS overview.

Fig. 8 Overview on the developed ANFIS

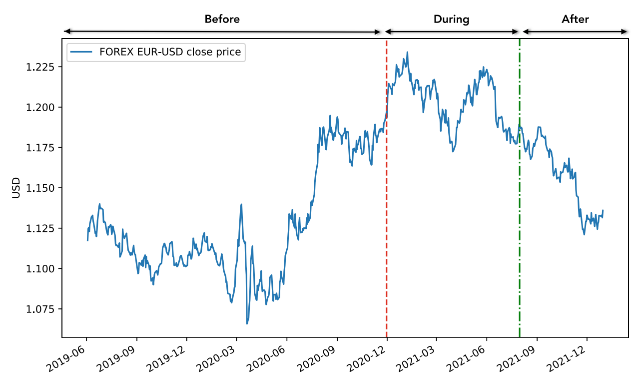


Fig. 9 FOREX USD-EUR close prices over the period of interest (bubble 2021). Before: from start to the red dashed line. During: from the red dashed line to the green dot-and-dash line. After: from the green dot-and-dash line to the end

one on Ethereum, while the lowest *Sortino* ($Sortino_{avg} = 2.7$) and *Omega ratio* ($Omega_{avg} = 1.4$) were on Bitcoin 2018, probably due to the **After** phase in Bitcoin 2018 in which the agent had a more "protective" behavior by carrying out fewer transactions (which correspond to a lower return).

Comparing *GGSMZ* with other agents (Table 17), in the case of BTC2018, our agent achieved the best results in terms of *Sharpe*, *Sortino*, and *Omega ratio* and outperformed the benchmark in terms of returns. For example, in the case of the **After** frame, it did not obtain the highest *Sharpe ratio* (lower than the GDX), but it can be considered the best agent is given the results of the remaining indicators. In the case of BTC2021, on the other hand, in the **Before** frame, *GGSMZ* obtains the best results (under all indicators, also outperforming the benchmark). At the same time, in the **During** and **After**, its behavior is overcome by the intelligent agents (SAC and PPO, respectively, for the time frame), which, although not having the highest returns and *Sharpe ratio*, get the best results under the other indicators. Finally, for the ETH2021 dataset, *GGSMZ* again attained the best results for the **Before** and **After** frames. At the same time, in the case of the **During**, a particular situation is configured: The A2C agent achieves better

Table 16 Results achieved by *GGSMZ* in the different phases (before, during, and after) of the Bitcoin bubble of 2018, Bitcoin bubble of 2021, and Ethereum bubble of 2021

Bubble	Market phase	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
BTC 2018	Before	1810.080	128.430	2.74	−52.631	3.90	1.76
	During	218.510	90.410	2.40	−34.357	3.80	1.60
	After	6.755	50.693	0.48	−30.615	0.65	1.10
BTC 2021	Before	421.810	115.120	1.82	−46.577	3.52	1.65
	During	124.320	63.430	2.10	−38.980	3.35	1.45
	After	71.839	57.971	1.84	−34.485	2.67	1.40
ETH 2021	Before	1142.420	124.329	2.68	−52.371	4.07	1.71
	During	63.432	113.286	2.10	−28.980	3.34	1.45
	After	415.030	81.277	3.97	−27.317	1.63	1.84

Table 17 Values of the different economic indicators for *GGSMZ* agent compared with those obtained in previous steps of the project by other trading agents experimented with particular focus on best *ZI/MI* and best *CI* agent for each bubble and each phase of it. In **bold** font the best results

Bubble	Phase	Agent	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
BTC2018	Before	Benchmark	1574.780	76.120	3.33	−35.500	5.89	1.82
		A2C	1557.399	88.146	4.09	**− 34.436**	7.28	2.04
		AA	28.481	1478.105	4.26	−92.475	13.32	2.37
		GGSMZ	**1810.080**	**128.430**	**2.74**	−52.631	**3.90**	**1.76**
	During	Benchmark	− 61.250	87.400	−1.23	−65.280	−1.64	0.82
		DDPG	− 52.558	62.219	−0.90	−62.219	−1.18	0.77
		AA	− 47.629	1062.85	3.58	−86.752	9.19	1.93
		GGSMZ	**218.510**	**90.410**	**2.40**	**− 34.357**	**3.80**	**1.60**
	After	Benchmark	− 47.960	54.230	−1.17	−61.570	−1.50	0.80
		A2C	− 41.013	55.576	−1.36	−49.159	−1.78	0.74
		GDX	**148.506**	4265.015	**2.28**	−94.172	21.84	3.35
		GGSMZ	6.755	**50.693**	0.48	**− 30.615**	**0.65**	**1.10**
BTC2021	Before	Benchmark	287.100	51.560	2.68	−25.400	4.37	1.63
		SAC	254.428	**61.592**	3.07	**− 20.723**	5.16	1.78
		GVWY	60.539	2685.954	2.59	−84.774	16.74	2.79
		GGSMZ	**421.810**	115.120	**1.82**	−46.577	**3.52**	**1.65**
	During	Benchmark	25.930	72.480	0.80	−53.060	1.21	1.14
		SAC	12.871	9.756	**0.81**	**− 37.236**	**1.14**	**1.15**
		GVWY	26.208	1892.337	3.81	−92.536	15.43	2.63
		GGSMZ	**124.320**	**63.430**	2.10	−38.980	3.35	1.45
	After	Benchmark	20.340	54.770	0.84	−31.620	1.22	1.14
		PPO	10.740	37.455	**0.84**	**− 24.109**	**1.39**	**1.18**
		GVWY	68.982	1615.411	3.89	−90.356	13.86	2.47
		GGSMZ	**71.839**	**57.971**	1.84	−34.485	2.67	1.40
ETH2021	Before	Benchmark	54.530	72.980	2.72	−32.680	4.44	1.62
		PPO	510.023	**98.592**	2.97	**− 32.218**	5.61	1.77
		GDX	23.080	1514.032	3.68	−88.579	12.52	2.32
		GGSMZ	**1142.420**	124.329	**2.68**	−52.371	**4.07**	**1.71**
	During	Benchmark	80.190	96.370	1.34	−57.120	2.00	1.25
		A2C	**76.759**	**109.762**	**1.63**	−56.335	**2.33**	**1.31**
		AA	62.575	1607.796	4.47	−92.308	14.50	2.53
		GGSMZ	63.432	113.286	2.10	**− 28.980**	3.34	1.45
	After	Benchmark	42.280	67.770	1.21	−30.050	1.84	1.21
		A2C	33.033	**65.566**	1.37	**− 22.234**	2.12	1.24
		ZIP	48.871	1031.884	3.91	−86.371	9.56	2.06
		GGSMZ	**415.030**	81.277	**3.97**	−27.317	**1.63**	**1.18**

results in terms of *Sortino* and *Omega*. In contrast, our agent outperforms the remaining indicators. In a Cliff [64] manner, we can introduce a dominance-hierarchy on all the different agents tested, which takes into account the results obtained and the ability to identify the particular market situation where, despite everything, our agent can be classified as the most optimal for investor support: *GGSMZ* $\simeq$ A2C > PPO $\simeq$ SAC > DDPG > TD3 > *ZIC* > GVWY $\simeq$ AA $\simeq$ GDX > ZIP.

6 Conclusion

In the last decade, investors have witnessed the bursting of different financial bubbles which represent a serious problem for world economy. This is even truer for the cryptocurrencies market.

In this work, we faced the problem of financial bubbles in the cryptocurrencies market, with particular interest toward Bitcoin and Ethereum with their peculiar features such as high volatility, high sensitivity to news and the growing interest by governments in its use as a decentralized currency. We have presented a comparison and throughout evaluation of autonomous, adaptive, automated traders in the Bitcoin market and Ethereum market in 2018 and 2021. In more details, our aim was to study how the different traders perform in such market in several phases (before, during and after a bubble). To the best of our knowledge, this was one of the first works analyzing such aspects and involving a broad set of traders in the experiments.

We have included in this study two set of traders: *ZI/MI* traders such as *ZIC*, *ZIP*, *GDX*, *GVWY*, *AA* and *CI* traders such as *A2C*, *PPO*, *DDPG*, *TD3* and *SAC*. The traders have been evaluated in terms of cumulative returns, volatility, *Sharpe ratio*, Max Drawdown, *Sortino ratio* and *Omega ratio* based on their behavior in the trading test period.

The results show that the *ZI/MI* agents, although not able to trade at the realistic price of the market, have absorbed the intrinsic characteristics of the market, proving themselves (some of them in particular) as excellent agents to identify the possible market phase. On the contrary, the *CI* agents have found excellent results in all the indicators taken as reference. However, the complexity of the latter type of agents makes one lose the explainability of the strategies followed, compared to the *ZI/MI*, whose implementation is easily explained even to non-experts.

In the light of the obtained results, we have proposed a new trading agent, namely *GGSMZ*, whose core is a neuro-fuzzy system trained following indications of *CI* traders previously analyzed. We have evaluated *GGSMZ* in the same frames as the other agents for each cryptos and we found that it outperforms other traders. Specifically, *GGSMZ* showed a cumulative return beyond that obtained by other traders and a volatility value very close to that recorded by benchmark in the same period. Although in a couple of time frames this agent's behavior was surpassed by that of the best *CI* agent, we argue our neuro-fuzzy model could be used by investors as a decision support tool.

Limitations One of the aspects we have overlooked in our study is related to the optimization of hyperparameters of *CI* agents. Due to our computational limits, we could not perform such tuning in this phase of the project, but we aim to propose, as future works, *ad-hoc* configurations of *CI* agents and evaluate the performance of such agents with parameters tuned against the settings provided by [55]. Another aspect is that of including more cryptocurrencies in the experiments. We are aware that there are different cryptocurrencies (less famous than Bitcoin and Ethereum) we have not considered in the present work, such as Cardano, Solana and Dogecoin. This will be part of future efforts of the project, where we aim to study agents' behavior on all the cryptocurrencies.

Future works We are currently planning a broader evaluation of trading agents in different markets (e.g., stock market, futures market) and with different constraints/scenarios (e.g., maximum or minimum number of daily operations on the markets, choose other indicators to compare agents, select time points other than bubbles). Meanwhile, we are working on surveying potential investors about the usage of *GGSMZ* in the cryptocurrencies market [28, 43]. Our aim is also to enhance the developed neuro-fuzzy system in order to enable it to sense and process further information such as the sentiment of news around the world (which it is known to have impacts on financial markets [15, 68, 80]). Another future direction could be that of improving the underlying mechanism of *GGSMZ* evaluating other learning and trading strategies.

Appendix A: FOREX 2021

We are interested in testing phase detection capabilities even outside crypto markets. Let us consider the Foreign Exchange Market (so-called FOREX), a market in which currencies (not cryptocurrencies) are traded. It is a different market from the crypto one because is a closing market (therefore with fewer trading days if we consider the same time interval). It is characterized by a very high volume of trades (which are not recorded, therefore an unusable feature) and is strongly influenced by events related to the real economies of different countries. Among the various currencies traded, we have selected the EUR-USD, which is fundamental in the European economic system. In particular, the reference period considered is 6/1/2019 to 12/31/2021, similar to the previous periods of 2021 considered but composed of about 700 observations, of which an extract is represented in Table 18.

We remark that, for this market, various exchanges do not register the feature volume, so the classic dataset becomes OHLC. Then, as before, we divide the time interval into three-time frames (which, however, especially for **Before** and **During**, correspond to those of Bitcoin and Ethereum 2021)—depicted in Fig. 10—and let various agents *CI* and *ZI/MI* operate in this market with the different conditions. In particular, the agents will have a portfolio of 100 price units at their disposal.

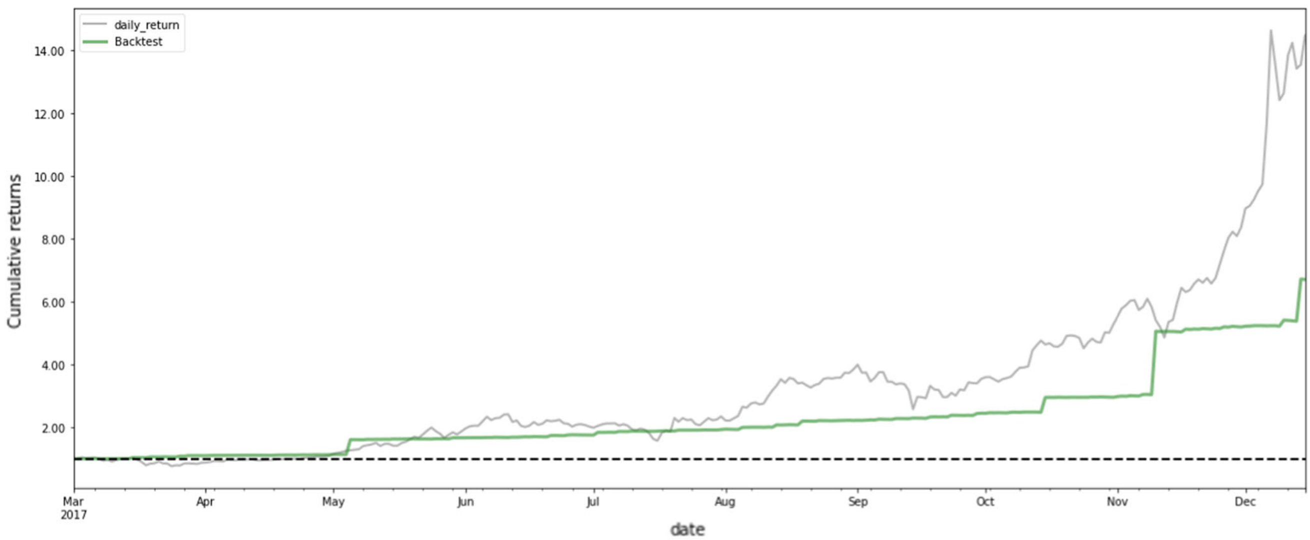
Before Time frame characterized by training from 6/1/2019 to 4/30/2020 (230 days), test from 5/1/2020 to 11/30/2020 (147 days) and the following benchmark values.

Table 18 Extract of the EUR-USD price dataset

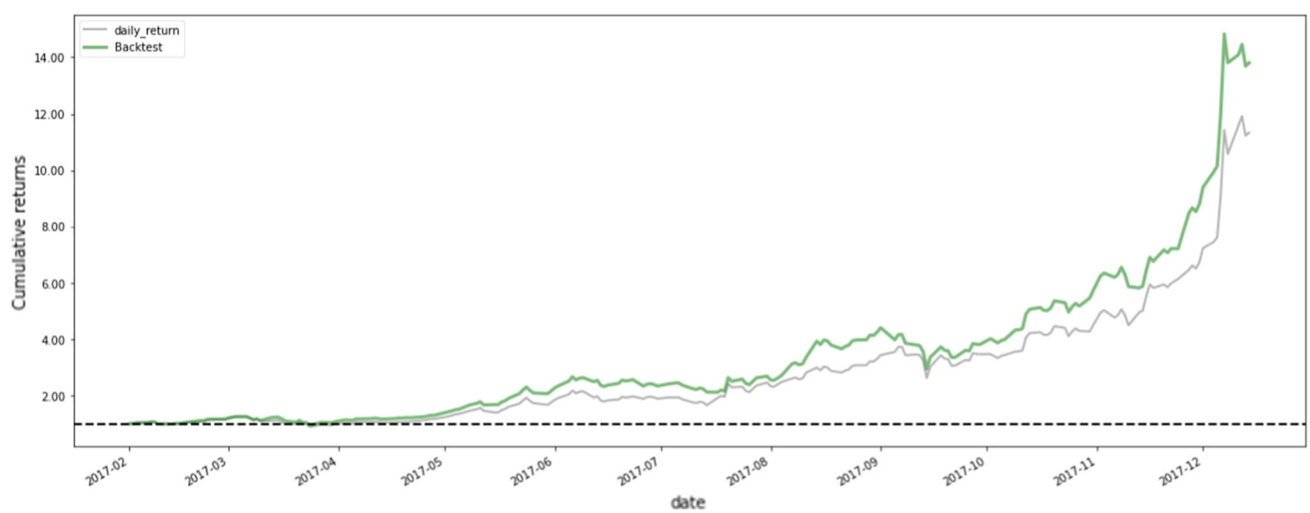
Date	Open	High	Low	Close
6/3/2019	1.1173	1.1211	1.1162	1.1174
6/4/2019	1.1245	1.1277	1.1227	1.1246
6/5/2019	1.1255	1.1304	1.1239	1.1254
6/6/2019	1.1227	1.1307	1.1223	1.1229
6/7/2019	1.1275	1.1348	1.1253	1.1276
⋮	⋮	⋮	⋮	⋮

Reference values FOREX2021 *Before*

- Annual returns: 10.73%;
- Volatility: 6.48%;
- Sharpe ratio: 2.06;
- Max Drawdown: −2.61%;
- Sortino ratio: 3.26;
- Omega ratio: 1.37.



(a) *ZIC* volatility



(b) A2C volatility

Fig. 10 Volatility of most relevant trading strategies (**Before**), Bitcoin bubble of 2018

Table 19 Comparison between the different agents in the **Before** period for EUR-USD situation of 2021

ZI/MI	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
(a) *ZI/MI* agents						
ZIC	354.57	12597.52	2.62	−99.156	52.74	6.61
ZIP	−2.692	1879.38	3.39	−92.739	14.26	2.56
GDX	41.398	2558.59	2.88	−96.488	16.27	2.73
AA	132.02	1195.364	3.95	−89.99	10.94	2.18
GVWY	−10.247	1573.708	3.90	−90.315	13.35	2.44
CI	**Return**	**Volatility**	**Sharpe**	**Drawdown**	**Sortino**	**Omega**
(b) *CI* agents						
A2C	−4.626	4.149	−1.51	−5.175	−2.13	0.74
DDPG	10.437	6.396	2.12	−2.582	3.41	1.39
PPO	1.833	2.512	0.98	−0.769	1.92	1.39
TD3	10.589	6.463	2.12	−2.609	3.41	1.39
SAC	7.561	5.685	1.75	−3.377	2.83	1.36

During Time frame characterized by training from 6/1/2019 to 11/30/2020 (378 days), test from 12/1/2020 to 7/31/2021 (167 days) and the following benchmark values.

Reference values FOREX2021 *During*
– Annual returns: −0.37%;
– Volatility: 5.88%;
– Sharpe ratio: −0.06;
– Max Drawdown: −5.01%;
– Sortino ratio: −0.08;
– Omega ratio: 0.98.

After Time frame characterized by training from 6/1/2019 to 31/7/2021 (546 days), test from 8/1/2021 to 12/31/2021 (106 days) and the following benchmark values.

Reference values FOREX2021 *After*
– Annual returns: −4.27%;
– Volatility: 5.23%;
– Sharpe ratio: −1.91;
– Max Drawdown: −5.74%;
– Sortino ratio: −2.38;
– Omega ratio: 0.71.

Results Tables 19, 20 and 21 summarize the behaviors of the different agents in the three time frames. The first noticeable thing is the low volatility of the benchmark (also recorded by *ZI* agents), typical of this type of market. This happens because the price changes occur on the eight decimal places recorded on the FOREX. While we cannot talk about a direct bubble on FOREX (as happened for cryptocurrencies, instead), we considered significant dates in dividing the 3 time frames to take into account the economic situation occurring in those periods. Furthermore, as shown graphically in Fig. 10, it can be seen that the time frames thus created show a situation similar to the previous bubble. Let us analyze the behavior of the different agents in these situations and compare it with that of the *GGSMZ* (shown in Table 22). The first essential behavior to note is the reduction of the ZIC volatility in the passage from one-time frame to the next, also highlighting the usefulness of the rule, previously defined, through which this agent can be an excellent indicator of the phase market (it is evident the similarity in the volatility of the frames **During** and **After**). In the first time frame (**Before**), the best agent for the *ZI/MI* was the AA (which got the best values of *Sharpe* and *Drawdown*, *Sortino* and *Omega*, and a very high return). In contrast, for the *CI* agents, the best behavior is contended by DDPG and TD3 (which have identical results). On the other hand, the behavior of the A2C stands out (opposite to its competitors), which carried out very few transactions that led to a negative return. Regarding, instead, our ANFIS agent, its behavior permitted it to obtain a very high return. However, the other indicators show that it was not the best agent (particularly for the very high *Sortino ratio*, drastically higher than the CI agents).

In the second time frame (**During**), the volatility of the ZIC was halved, and the best *ZI/MI* agent was the GDX (since it was the agent with the highest Sharpe and the lowest *Drawdown*, *Sortino*, and *Omega* at the same time, while the ZIC got only the highest *Sharpe*). In contrast, the CI agents followed a particular behavior. The first evidence

Table 20 Comparison between the different agents in the **During** period for EUR-USD situation of 2021

ZI/MI	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
(a) *ZI/MI* agents						
ZIC	−28.516	4843.731	4.72	−95.4	36.28	4.56
ZIP	52.692	1789.634	3.85	−92.656	15.02	2.67
GDX	19.611	1290.658	4.10	−89.658	11.70	2.26
AA	−14.877	1279.428	3.91	−92.192	11.31	2.19
GVWY	−5.194	2328.485	3.39	−92.234	17.07	2.80
CI	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
(b) *CI* agents						
A2C	−4.07	4.189	−1.48	−5.191	−1.87	0.73
DDPG	−0.468	5.687	−0.10	−4.906	−0.13	0.93
PPO	0.031	4.275	0.03	−3.663	0.05	1.01
TD3	−0.473	5.746	−0.10	−4.956	−0.13	0.98
SAC	−1.077	3.58	−1.03	−1.583	−1.38	0.94

Table 21 Comparison between the different agents in the **After** period for EUR-USD situation of 2021

ZI/MI	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
(a) *ZI/MI* agents						
ZIC	30.976	4208.564	3.31	−97.322	35.06	4.45
ZIP	85.325	1113.194	4.31	−84.908	10.9	2.14
GDX	−18.097	1538.056	4.27	−91.835	13.92	2.43
AA	−73.026	2927.738	3.01	−96.564	19.43	3.10
GVWY	25.211	1217.477	3.65	−89.274	11.19	2.20
CI	Return	Volatility	Sharpe	Drawdown	Sortino	Omega
(b) *CI* agents						
A2C	−3.961	4.872	−1.97	−4.987	−2.44	0.69
DDPG	−4.358	5.285	−2.00	−5.622	−2.48	0.71
PPO	−4.043	4.737	−2.07	−5.662	−2.53	0.70
TD3	−4.298	5.211	−2.00	−5.545	−2.48	0.71
SAC	−1.018	1.069	−2.29	−1.083	−2.84	0.60

Table 22 Results obtained by *GGSMZ* in the different phases (before, during, and after) of FOREX 2021

Market	Time frame	Return (%)	Volatility (%)	Sharpe	Drawdown (%)	Sortino	Omega
FOREX 2021	Before	18.986	6.309	4.63	−1.783	8.53	2.12
	During	10.936	5.676	3.60	−2.171	6.26	1.80
	After	−2.451	4.329	−1.30	−4.902	−1.68	0.68

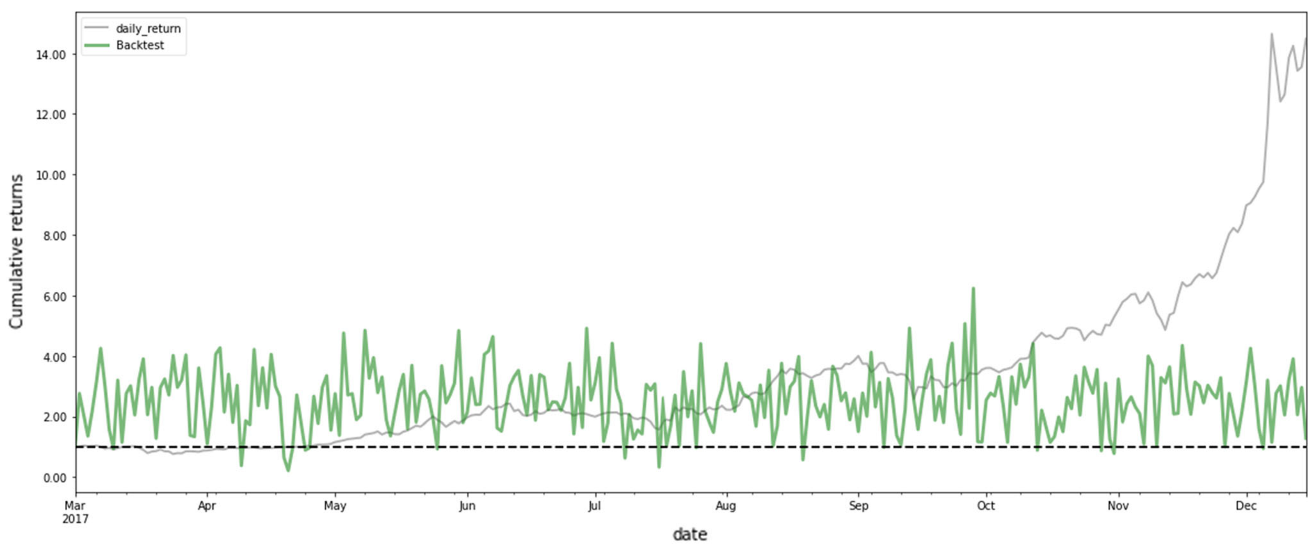
is the PPO that obtained a positive return, the highest *Sharpe*, and the lowest *Sortino* (but not the lowest *Omega*) and performed a limited number of transactions at certain strategic moments, which enabled it to get a positive result. On the other hand, the lowest Omega ratio was achieved by DDPG, which maintained a behavior very similar to TD3. Therefore, we could say that the best agents in this phase were PPO and DDPG. As for *GGSMZ*, however, as in the previous time frame, its return was very high (considering that it is diametrically opposite to that recorded by the benchmark), but its *Sortino* and *Omega ratio* indicators are not the best.

Finally, in the last time frame (**After**), the volatility of the ZIC decreased compared to the **During**, but remained in any case on a constant level (highlighting the transition to a subsequent phase). Furthermore, in this situation, even the benchmark values highlight that we are in a decreasing market phase (probably the continuation of a market peak, as it happened). The best *ZI/MI* agent was the ZIP, which despite its casual behavior, managed to obtain the best performance indicators. On the side of the *CI* agents, the best agent was A2C. Despite having maintained a somewhat similar behavior to its competitors, it stood out for having recorded the best values of the various indicators. However, for our agent ANFIS, this time frame was one of the best as *GGSMZ* recorded the best behavior (even compared to the previous frames), which is in line with the benchmark and outperformed the *CI* competitors.

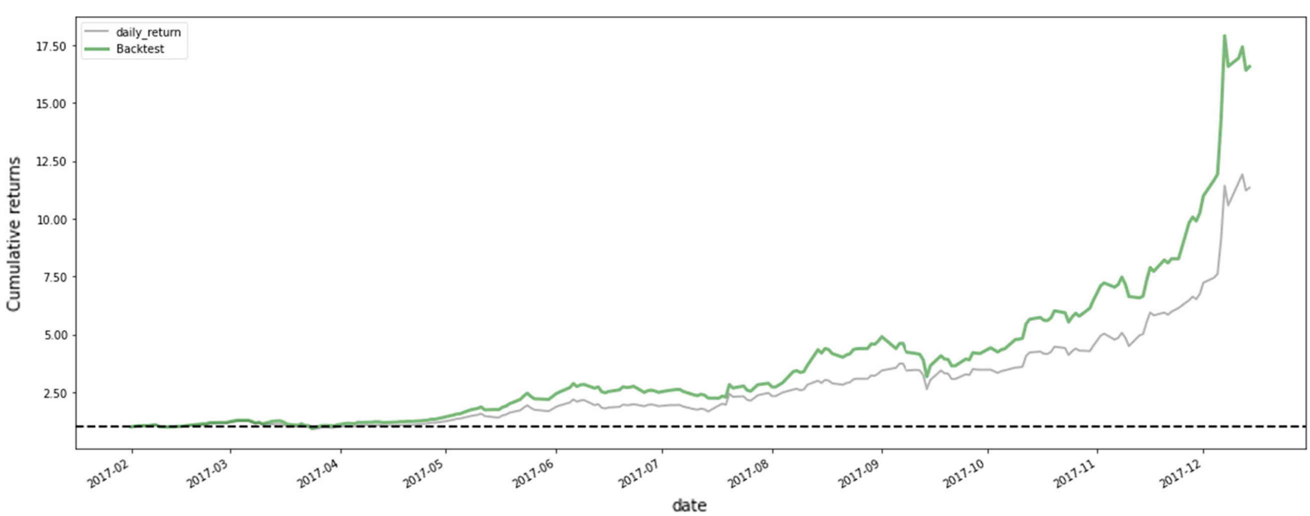
From these results, we can conclude how the use of *ZI/MI* agents is also functional in this market to understand the market phase (in particular of the ZIC). The *CI* agents, on the other hand, compared to the crypto markets, despite having achieved good results, have not fully shown their trading skills, in some cases carrying out a minimal number of transactions. This is probably why the GGSMZ agent did not have an ideal behavior in the **Before** and **During** frames, as it was trained on the transactions of the previous 5 *CI* agents. Therefore, intelligent agents are recommended with great caution in the FOREX market.

Appendix B: Depicting volatility and cumulative returns of traders

In this section, we provide the graphics illustrating the volatility and the cumulative returns of the *best* experimented traders during the different bubbles faced. In particular, Bitcoin 2018 (Sect. 1), Bitcoin 2021 (Sect. 2), Ethereum 2021 (Sect. 3) and FOREX 2021 (Sect. 4). For all the other graphics, we refer the reader to https://bit.ly/3wrkwi7.

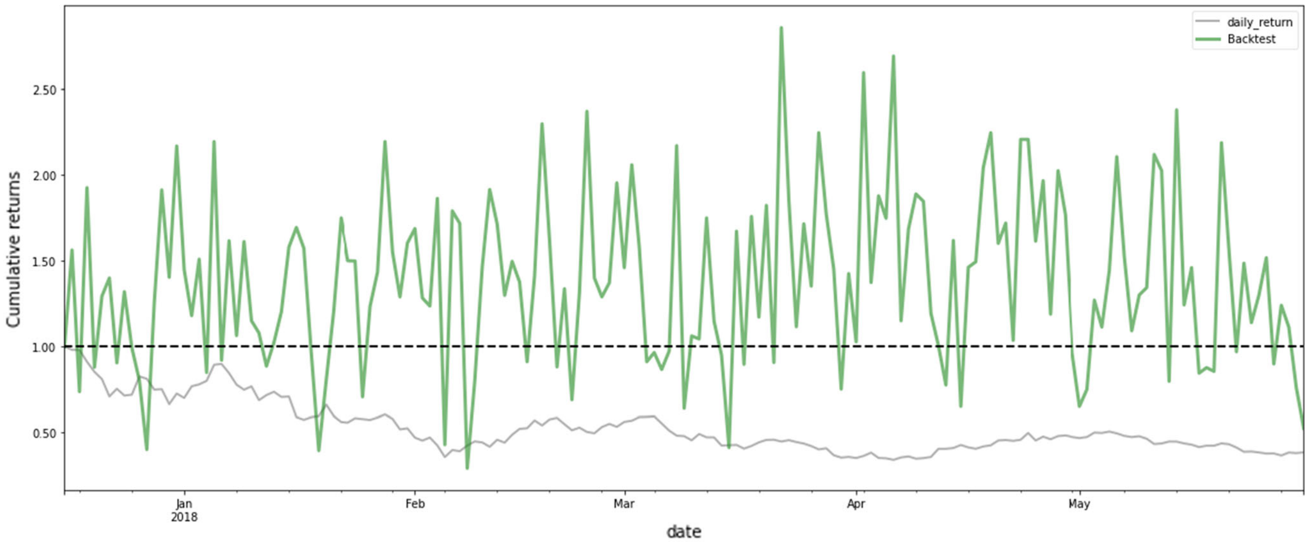


(a) AA

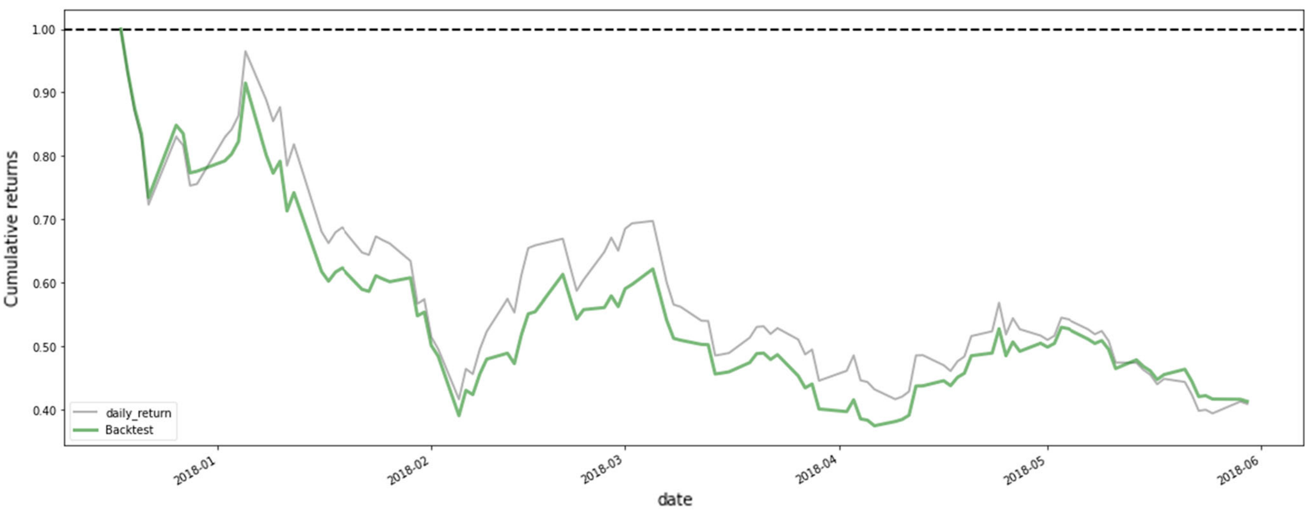


(b) A2C

Fig. 11 Cumulative returns of most relevant trading agents (**Before**), Bitcoin bubble of 2018

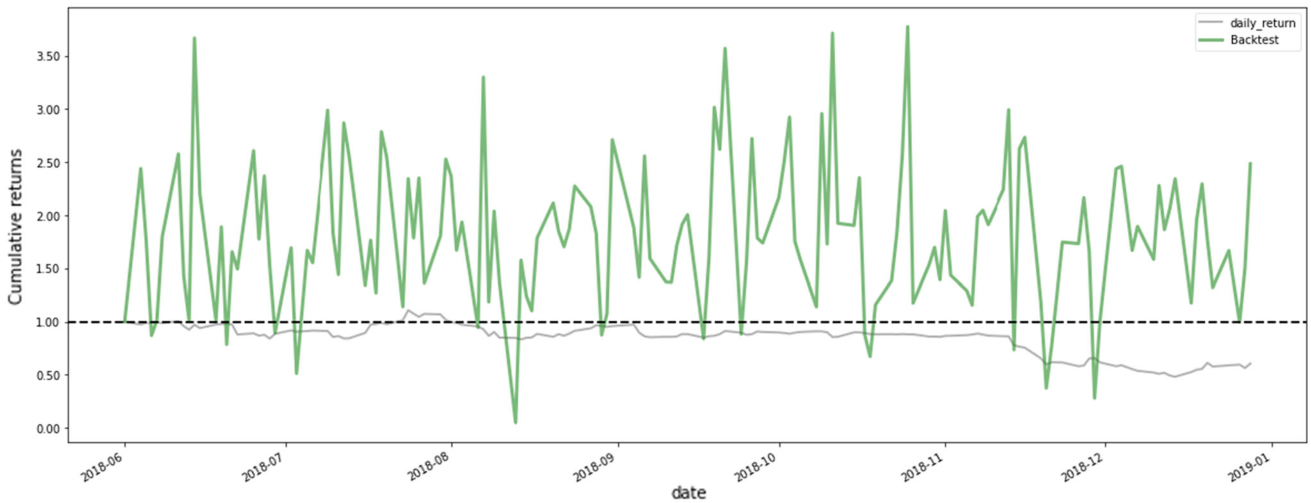


(a) AA

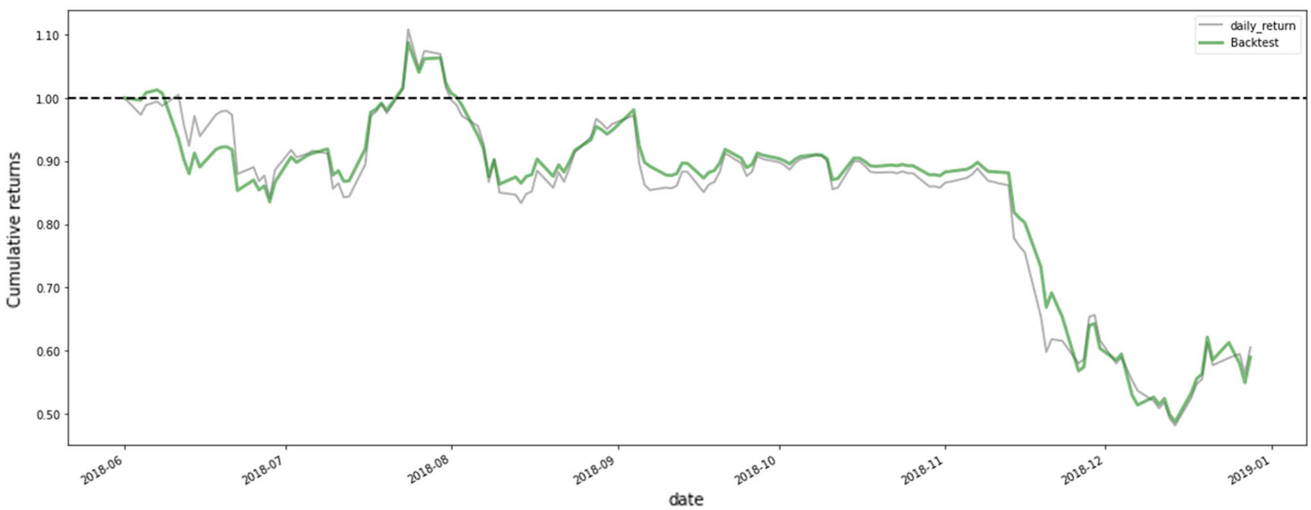


(b) DDPG

Fig. 12 Cumulative returns of most relevant trading agents (**During**), Bitcoin bubble of 2018

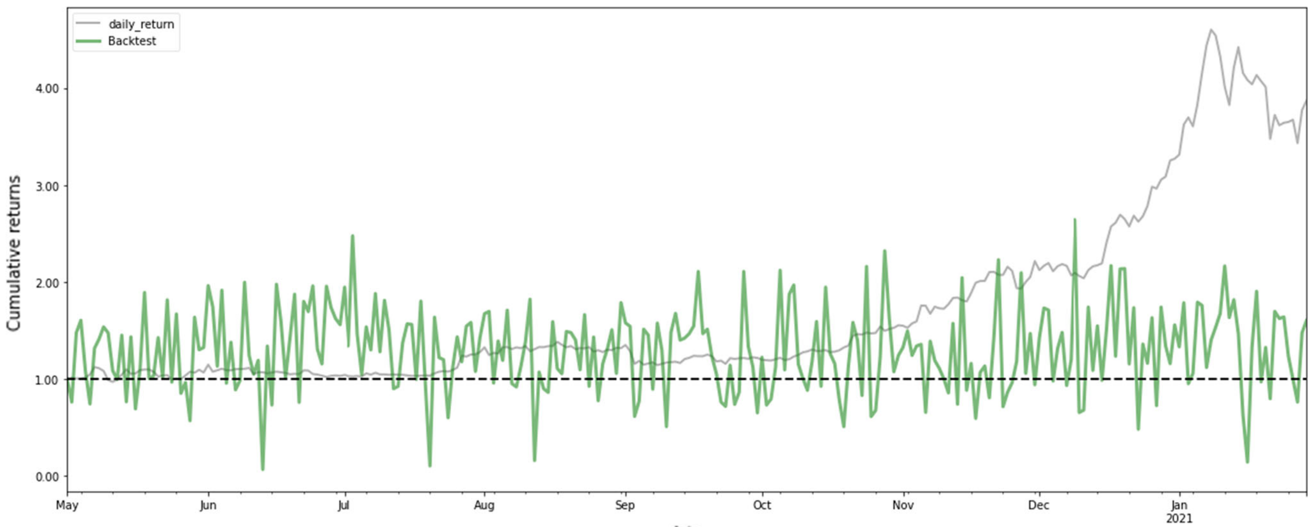


(a) GDX

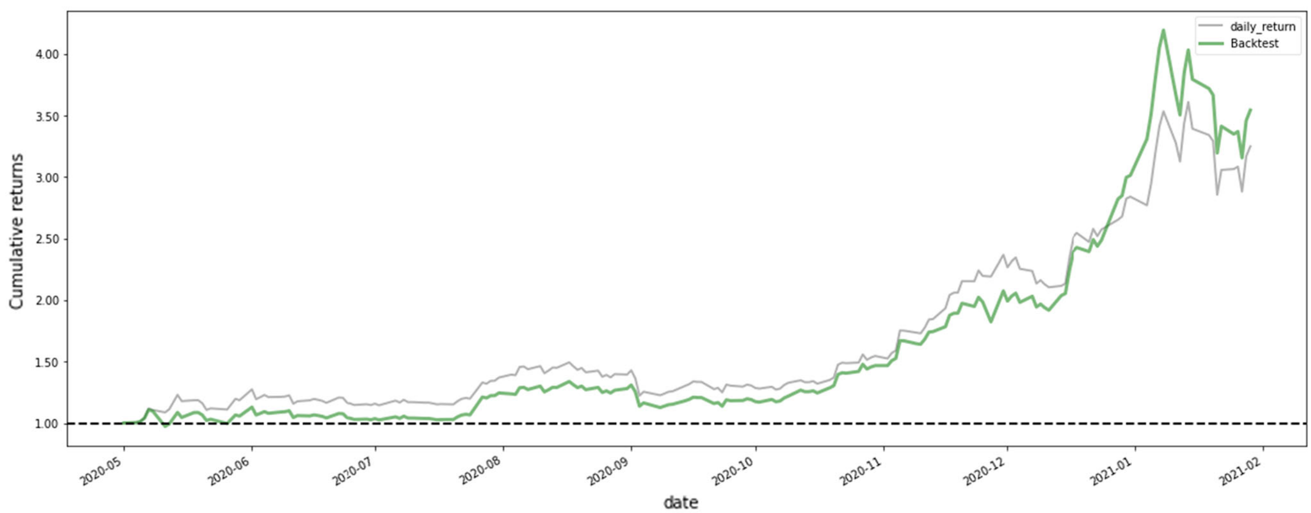


(b) A2C

Fig. 13 Cumulative returns of most relevant trading agents (**After**), Bitcoin bubble of 2018

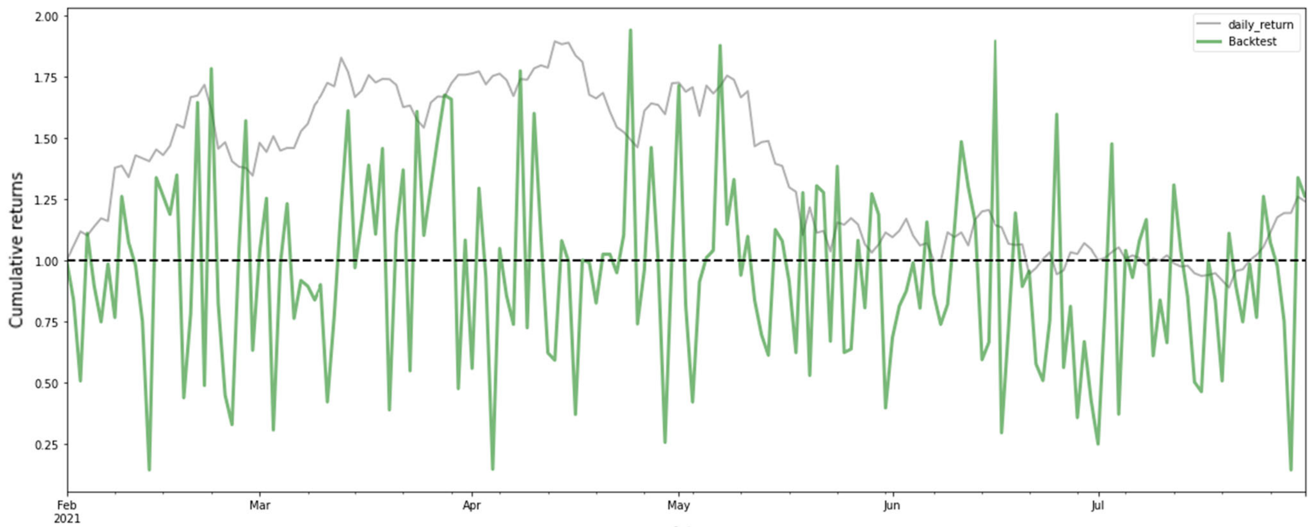


(a) GVWY

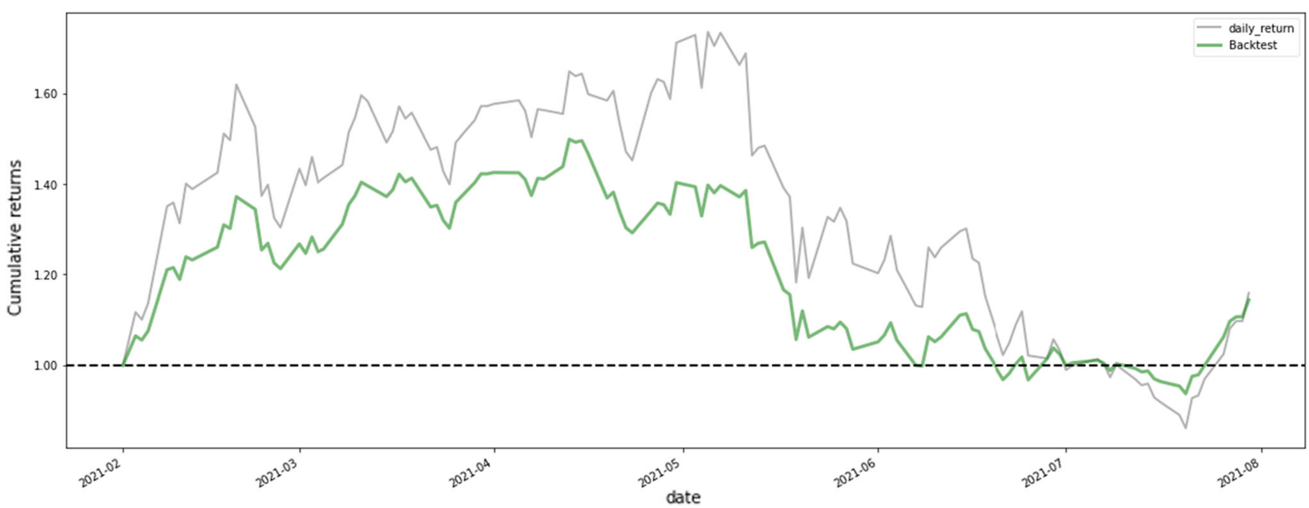


(b) SAC

Fig. 14 Cumulative returns of most relevant trading agents (**Before**), Bitcoin bubble of 2021

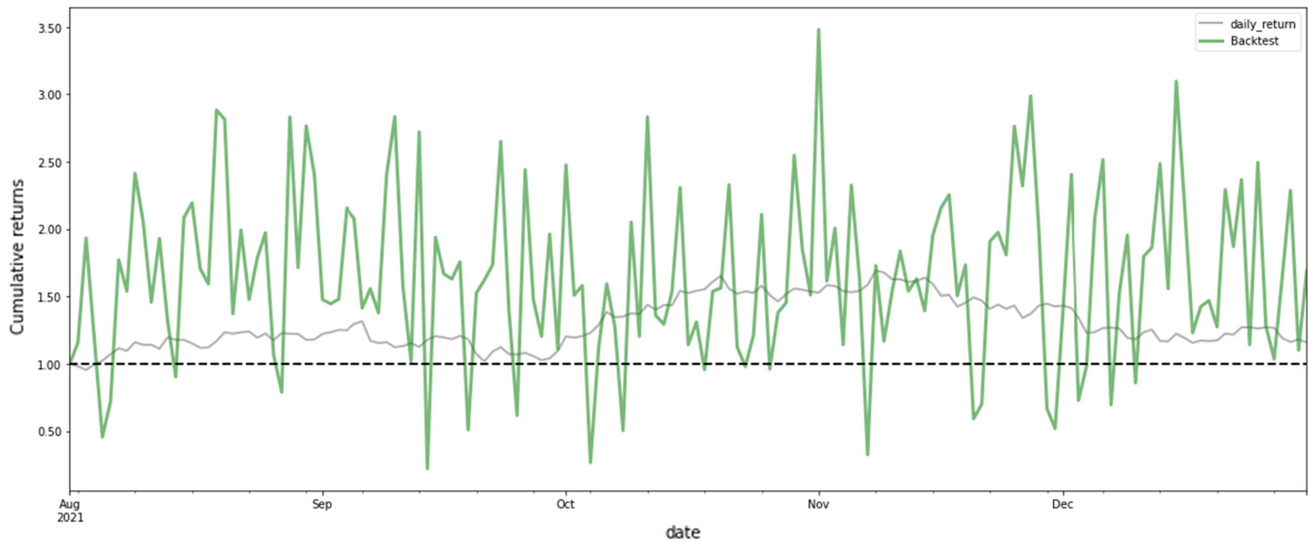


(a) GVWY

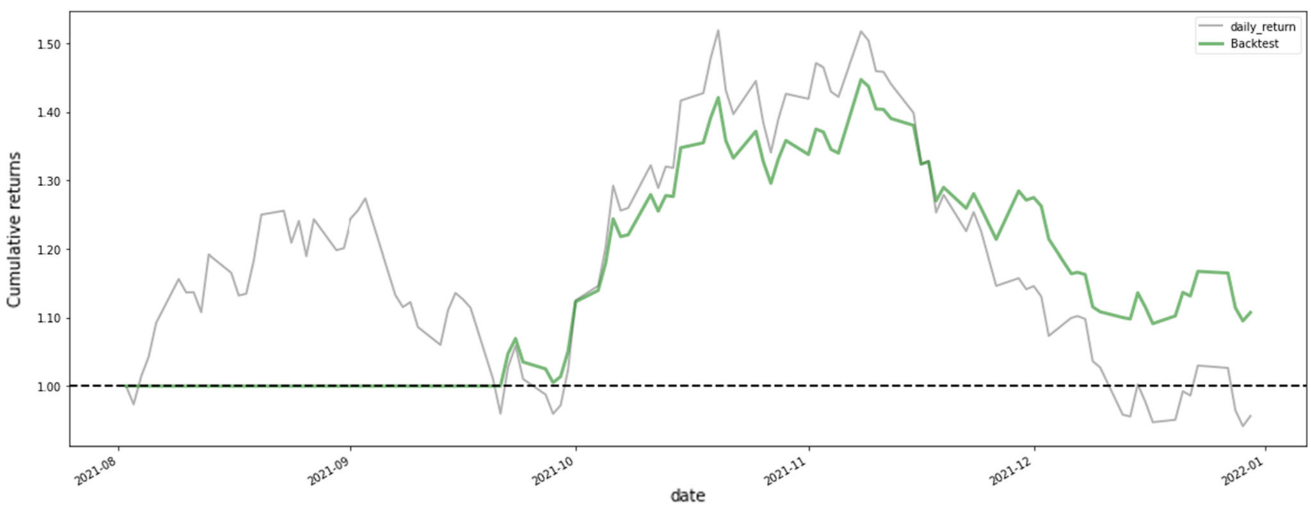


(b) SAC

Fig. 15 Cumulative returns of most relevant trading agents (**During**), Bitcoin bubble of 2021

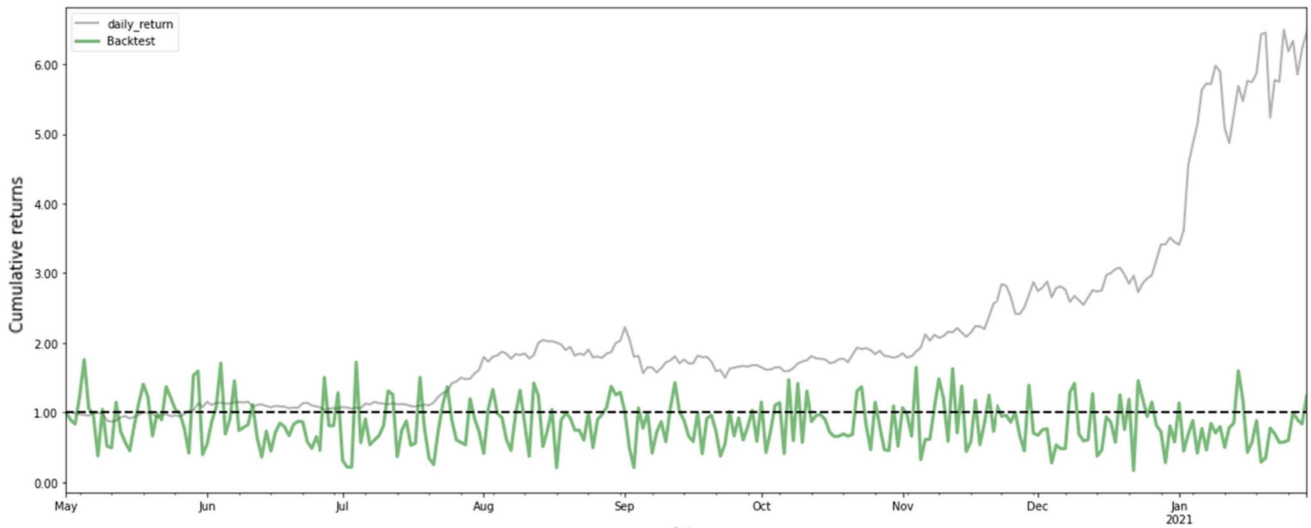


(a) GVWY

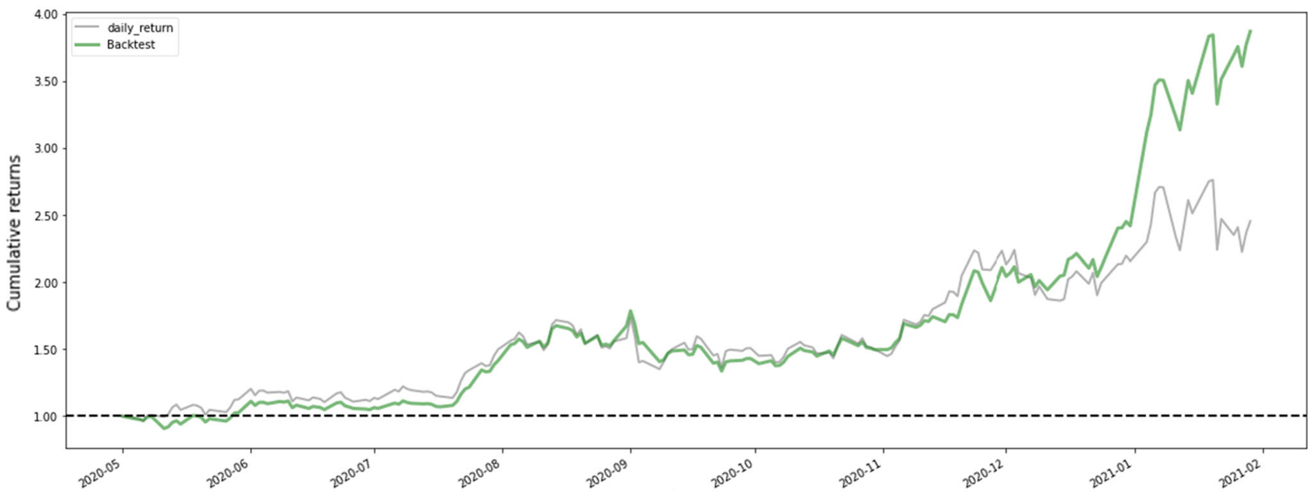


(b) PPO

Fig. 16 Cumulative returns of most relevant trading agents (**After**), Bitcoin bubble of 2021

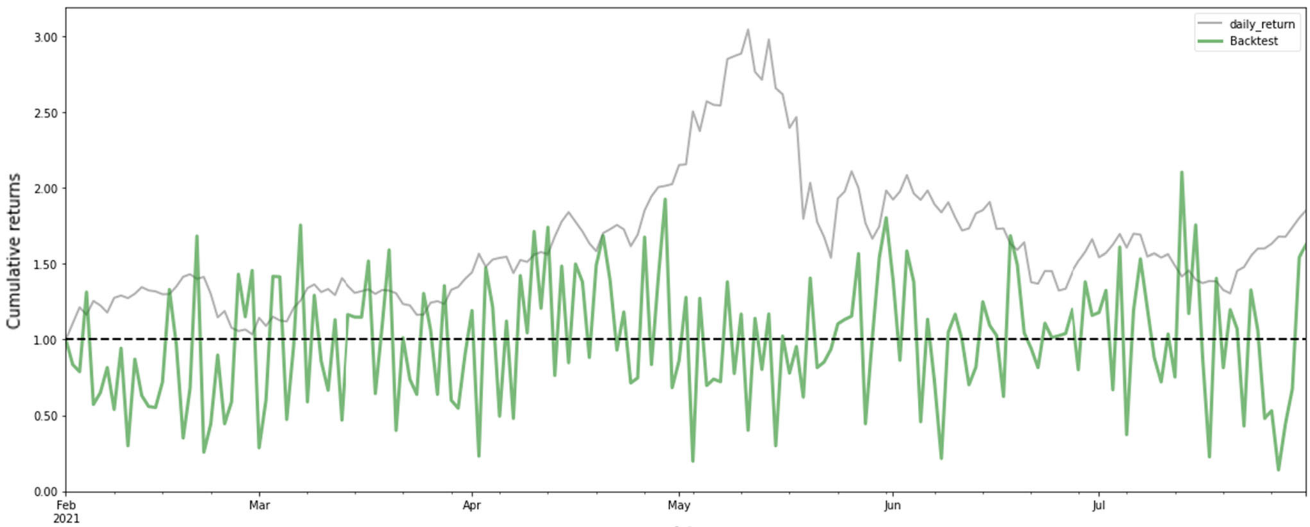


(a) GDX

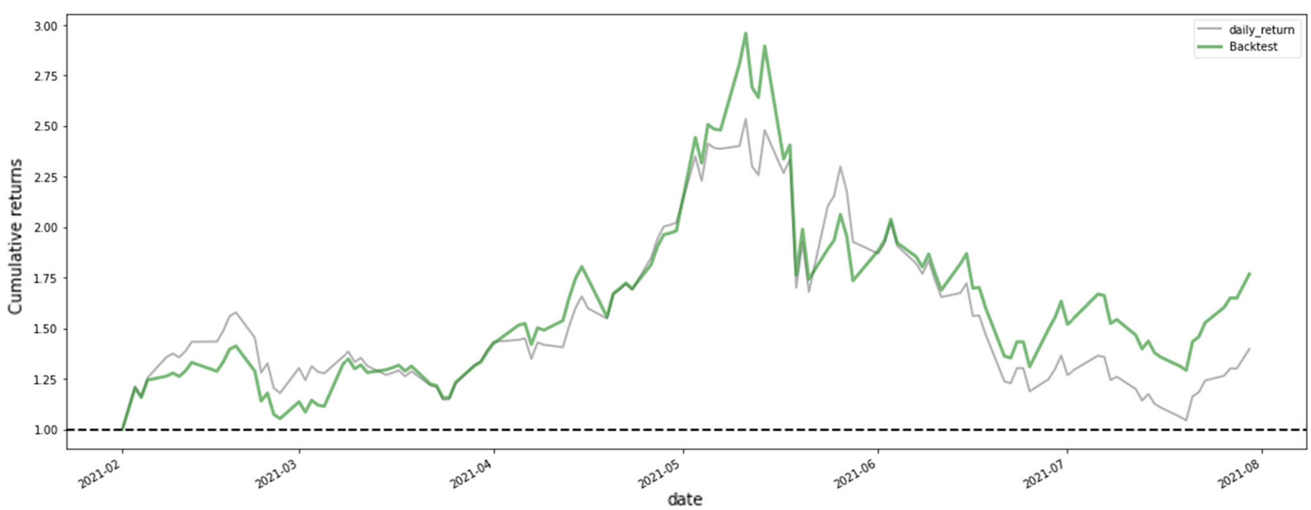


(b) PPO

Fig. 17 Cumulative returns of most relevant trading agents (Before), Ethereum bubble of 2021

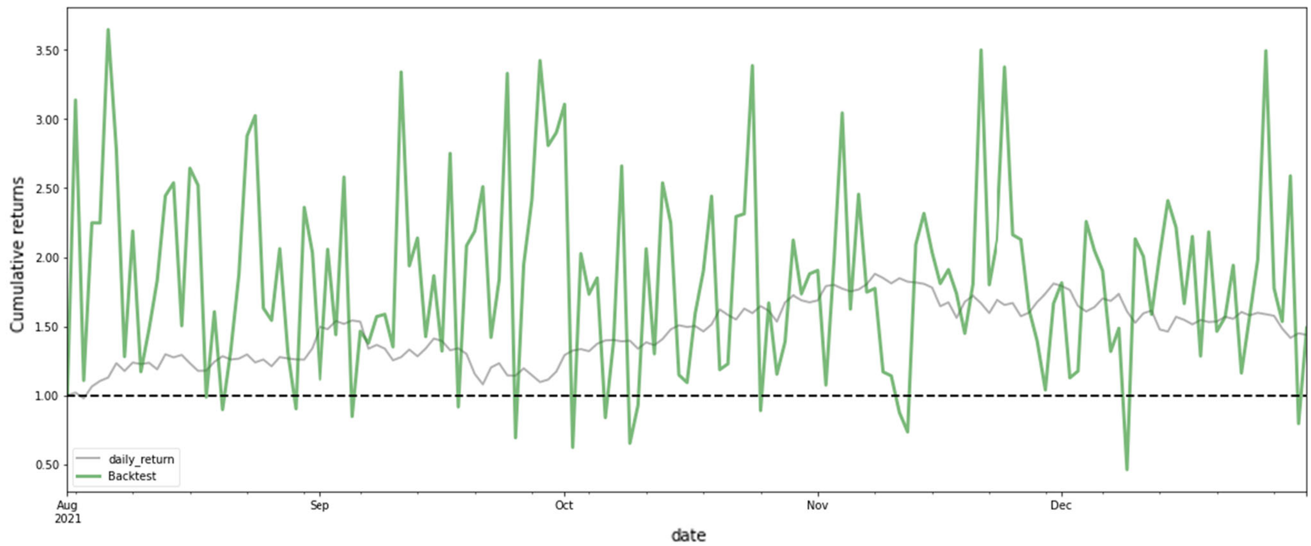


(a) AA

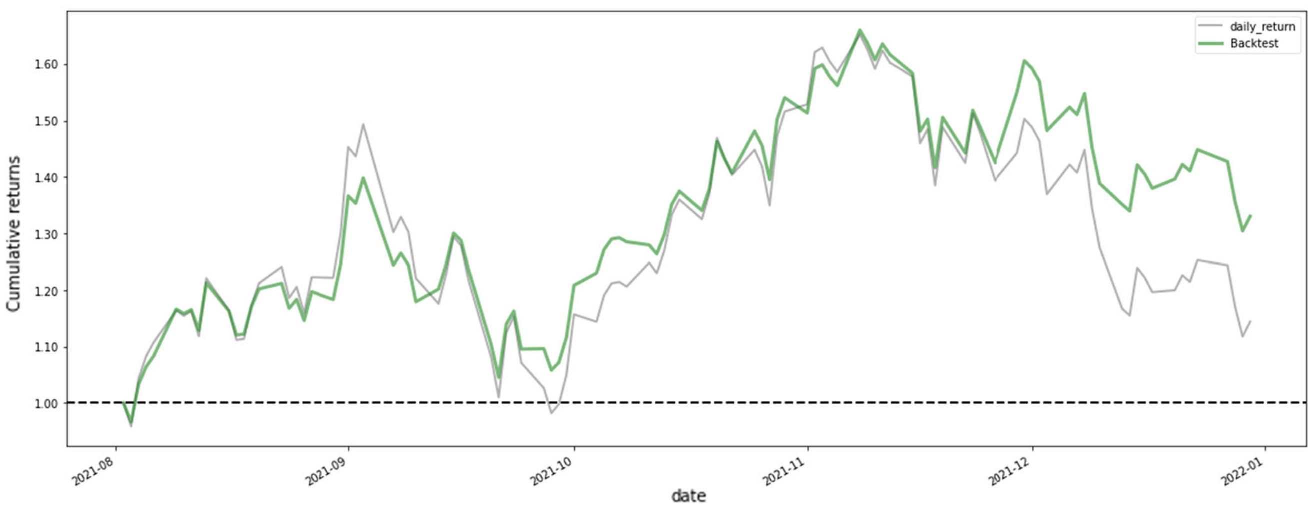


(b) A2C

Fig. 18 Cumulative returns of most relevant trading agents (During), Ethereum bubble of 2021



(a) ZIP



(b) A2C

Fig. 19 Cumulative returns of most relevant trading agents (After), Ethereum bubble of 2021

BTC2018 bubble

In this section, we provide the graphics illustrating the volatility and the cumulative returns of the best experimented traders during the Bitcoin 2018 bubble. See Figs 11, 12, 13, 14, 15, 16, 17, 18, 19.

BTC2021 bubble

In this section, we provide the graphics illustrating the volatility and the cumulative returns of the best experimented traders during the Bitcoin 2021 bubble.

ETH2021 bubble

In this section, we provide the graphics illustrating the volatility and the cumulative returns of the best experimented traders during the Ethereum 2021 bubble.

FOREX2021

In this section, we provide the graphics illustrating the volatility and the cumulative returns of the best experimented traders concerning the Forex 2021 market.

Table 23 MSE and RMSE of ANFIS during training and testing phases

Bubble (Phase)	Epochs	Training		Testing	
		MSE	RMSE	MSE	RMSE
BTC2018 (During)	51	0.199	0.446	4.005	2.001
BTC2018 (After)	59	0.119	0.346	4.665	2.160
BTC2021 (Before)	40	0.181	0.425	3.588	1.894
BTC2021 (During)	48	0.2225	0.474	3.718	1.928
BTC2021 (After)	54	0.211	0.459	4.111	2.027
ETH2021 (Before)	40	0.211	0.459	3.211	1.791
ETH2021 (During)	47	0.393	0.626	3.551	1.884
ETH2021 (After)	53	0.222	0.471	3.001	1.732

Appendix C: Further ANFIS training and testing results

In this section, we report the results obtained by ANFIS during the different experiments in terms of MSE and RMSE reached in training and testing. Table 23 shows such results.

Funding Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abdi H, Williams LJ (2010) Principal component analysis. Wiley Interdiscip Rev Comput Stat 2(4):433–459
2. Almahdi S, Yang SY (2017) An adaptive portfolio trading system: a risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. Expert Syst Appl 87:267–279. https://doi.org/10.1016/j.eswa.2017.06.023
3. Armaghani DJ, Hajihassani M, Sohaei H, Mohamad ET, Marto A, Motaghedi H, Moghaddam MR (2015) Neuro-fuzzy technique to predict air-overpressure induced by blasting. Arab J Geosci 8(12):10937–10950
4. Armaghani DJ, Mohamad ET, Momeni E, Narayanasamy MS et al (2015) An adaptive neuro-fuzzy inference system for predicting unconfined compressive strength and young's modulus: a study on main range granite. Bull Eng Geol Environ 74(4):1301–1319
5. Atsalakis G, Frantzis D, Zopounidis C (2016) Commodities' price trend forecasting by a neuro-fuzzy controller. Energy Syst 7(1):73–102
6. Atsalakis GS (2016) Using computational intelligence to forecast carbon prices. Appli Soft Comput 43:107–116
7. Atsalakis GS, Atsalaki IG, Pasiouras F, Zopounidis C (2019) Bitcoin price forecasting with neuro-fuzzy techniques. Eur J Oper Res 276(2):770–780
8. Atsalakis GS, Dimitrakakis EM, Zopounidis CD (2011) Elliott wave theory and neuro-fuzzy systems, in stock market prediction: the wasp system. Expert Syst Appl 38(8):9196–9206
9. Atsalakis GS, Valavanis KP (2009) Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert Syst Appl 36(7):10696–10707
10. Azadeh A, Moghaddam M, Khakzad M, Ebrahimipour V (2012) A flexible neural network-fuzzy mathematical programming algorithm for improvement of oil price estimation and forecasting. Comput Ind Eng 62(2):421–430
11. Baralis E, Cagliero L, Cerquitelli T, Garza P, Pulvirenti F (2017) Discovering profitable stocks for intraday trading. Inf Sci 405:91–106. https://doi.org/10.1016/j.ins.2017.04.013
12. Bau W, Liu XY (2019) Multi-agent deep reinforcement learning for liquidation strategy analysis. arXiv:1906.11046 pp 1–9
13. Baur DG, Hong K, Lee AD (2018) Bitcoin: medium of exchange or speculative assets? J Int Financ Mark Inst Money 54:177–189
14. Becker GS (1962) Irrational behavior and economic theory. J Polit Econ 70(1):1–13
15. Bollen J, Mao H, Zeng X (2011) Twitter mood predicts the stock market. J Comput Sci 2(1):1–8. https://doi.org/10.1016/j.jocs.2010.12.007
16. Bouri E, Shahzad SJH, Roubaud D (2019) Co-explosivity in the cryptocurrency market. Financ Res Lett 29:178–183
17. Cason TN, Friedman D (1996) Price formation in double auction markets. J Econ Dyn Control 20(8):1307–1337
18. Cheah ET, Fry J (2015) Speculative bubbles in bitcoin markets? An empirical investigation into the fundamental value of bitcoin. Econ Lett 130:32–36
19. Cliff D (2003) Explorations in evolutionary design of online auction market mechanisms. Electron Commer Res Appl 2(2):162–175. https://doi.org/10.1016/S1567-4223(03)00017-6
20. Cliff D (2006) Zip60: an enhanced variant of the zip trading algorithm. In: The 8th IEEE international conference on e-commerce technology and the 3rd IEEE international conference on enterprise computing, e-commerce, and e-services (CEC/EEE'06), pp 15–15. https://doi.org/10.1109/CEC-EEE.2006.99
21. Cliff D (2018) Bse: a minimal simulation of a limit-order-book stock exchange. arXiv:1809.06027
22. Cliff D (2018) Bse: a minimal simulation of a limit-order-book stock exchange. In: Affenzeller M, Bruzzone A, Jimenez E, Longo F, Merkuryev Y, Piera M (eds) 30th European modeling and simulation symposium (EMSS 2018). DIME University of Genoa, Italy, pp 194–203
23. Cliff D (2019) Exhaustive testing of trader-agents in realistically dynamic continuous double auction markets: Aa does not dominate. In: ICAART (2), pp 224–236

24. Cliff D, Bruten J (1997) Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical Report HPL-97-91, HP Labs
25. Das R, Hanson JE, Kephart JO, Tesauro G (2001) Agent-human interactions in the continuous double auction. In: International joint conference on artificial intelligence, vol 17, pp 1169–1178. Lawrence Erlbaum Associates Ltd
26. De Luca M, Cliff D (2011) Agent-human interactions in the continuous double auction, redux. Proceedings ICAART-2011
27. De Luca M, Cliff D (2011) Human-agent auction interactions: adaptive-aggressive agents dominate. In: Twenty-second international joint conference on artificial intelligence
28. De Prisco R, Guarino A, Lettieri N, Malandrino D, Zaccagnino R (2021) Providing music service in ambient intelligence: experiments with gym users. Expert Syst Appl 177:114951
29. Deng Y, Bao F, Kong Y, Ren Z, Dai Q (2017) Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans Neural Netw Learn Syst 28(3):653–664. https://doi.org/10.1109/TNNLS.2016.2522401
30. Duffy J, Ünver MU (2006) Asset price bubbles and crashes with near-zero-intelligence traders. Econ theory 27(3):537–563
31. Feng W, Wang Y, Zhang Z (2018) Informed trading in the bitcoin market. Financ Res Lett 26:63–70
32. Friedman D (1991) A simple testable model of double auction markets. J Econ Behav Org 15(1):47–70
33. Fujimoto S, van Hoof H, Meger D (2018) Addressing function approximation error in actor-critic methods. In: ICML, pp 1582–1591
34. Gandal N, Hamrick J, Moore T, Oberman T (2018) Price manipulation in the bitcoin ecosystem. J Monet Econ 95:86–96
35. Gjerstad S, Dickhaut J (1998) Price formation in double auctions. Games Econ Behav 22(1):1–29
36. Gode DDK, Sunder S (2004) Double auction dynamics: structural effects of non-binding price controls. J Econ Dyn Control 28(9):1707–1731. https://doi.org/10.1016/j.jedc.2003.06.001
37. Gode DK, Spear S, Sunder S (2004) Convergence of double auctions to pareto optimal allocations in the edgeworth box. Yale School of Management Working Papers, ICF
38. Gode DK, Sunder S (1993) Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. J Polit Econ 101(1):119–137
39. Greaves A, Au B (2015) Using the bitcoin transaction graph to predict the price of bitcoin. No Data
40. Grossklags J, Schmidt C (2003) Artificial software agents on thin double auction markets: a human trader experiment. In: IEEE/WIC international conference on intelligent agent technology. IAT 2003, pp 400–407. IEEE
41. Grossklags J (2006) Schmidt, C (2006) Software agents and market (in) efficiency: a human trader experiment. IEEE Trans Syst Man Cybern Part C (Appl Rev) 36(1):56–67
42. Guan M, Liu XY (2021) Explainable deep reinforcement learning for portfolio management: an empirical approach. In: 2nd ACM international conference on AI in finance (ICAIF'21). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3490354.3494415
43. Guarino A, Lettieri N, Malandrino D, Zaccagnino R (2021) A machine learning-based approach to identify unlawful practices in online terms of service: analysis, implementation and evaluation. Neural Comput Appl 33(24):17569–17587
44. Haarnoja T, Zhou A, Abbeel P, Levine S (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv:1801.01290v2
45. Habibnia A (2010) Forecasting the world gold price using optimized neuro-fuzzy with genetic algorithm (ga-anfis) and smooth transition regression with long memory (fi-star) modelling. SSRN 2010545
46. Jamal K, Sunder S (1996) Bayesian equilibrium in double auctions populated by biased heuristic traders. J Econ Behav Org 31(2):273–291. https://doi.org/10.1016/S0167-2681(96)00892-X
47. Jang JS (1993) Anfis: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685
48. Jiang Z, Xu D, Liang J (2017) A deep reinforcement learning framework for the financial portfolio management problem. arXiv:1706.10059 pp 1–31
49. Jin Z, Yang Y, Liu Y (2020) Stock closing price prediction based on sentiment analysis and lstm. Neural Comput Appl 32(13):9713–9729
50. Kristoufek L (2015) What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis. PLoS ONE 10(4):e0123923
51. Ladley D (2012) Zero intelligence in economics and finance. Knowl Eng Rev 27(2):273–286. https://doi.org/10.1017/S0269888912000173
52. Liang E, Liaw R, Nishihara R, Moritz P, Fox R, Goldberg K, Gonzalez JE, Jordan MI, Stoica I (2018) Rllib: Abstractions for distributed reinforcement learning. In: International Conference on Machine Learning (ICML)
53. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2019) Continuous control with deep reinforcement learning. arXiv:1509.02971v6
54. Liu XY, Li Z, Wang Z, Zheng J (2021) Elegantrl: a scalable and elastic deep reinforcement learning library. Https://github.com/AI4Finance/Foundation/ElegantRL
55. Liu XY, Yang H, Gao J, Wang CD (2021) FinRL: deep reinforcement learning framework to automate trading in quantitative finance. In: ACM International Conference on AI in Finance (ICAIF 2021)
56. Liu Y, Liu Q, Zhao H, Pan Z, Liu C (2020) Adaptive quantitative trading: an imitative deep reinforcement learning approach. Proc AAAI Conf Artif Intell 34(02):2128–2135. https://doi.org/10.1609/aaai.v34i02.5587
57. Lu W, Li J, Wang J, Qin L (2021) A cnn-bilstm-am method for stock price prediction. Neural Comput Appl 33(10):4741–4753
58. Madan I, Saluja S, Zhao A (2015) Automated bitcoin trading via machine learning algorithms. URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan **20**
59. Mamdani EH, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man-Mach Stud 7(1):1–13
60. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In: Balcan MF, Weinberger KQ (eds) Proceedings of the 33rd international conference on machine learning, *proceedings of machine learning research*, vol 48, pp 1928–1937. PMLR, New York, New York, USA
61. Mudassir M, Bennbaia S, Unal D, Hammoudeh M (2020) Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05129-6
62. Preist C, van Tol M (1998) Adaptive agents in a persistent shout double auction. In: Proceedings of the first international conference on Information and computation economies, pp 11–18
63. Raffin A, Hill A, Ernestus M, Gleave A, Kanervisto A, Dormann N (2019) Stable baselines3. https://github.com/DLR-RM/stable-baselines3
64. Rollins, M., Cliff, D.: Which trading agent is best? using a threaded parallel simulation of a financial market changes the pecking-order. arXiv preprint arXiv:2009.06905 (2020)
65. Safa M, Sari PA, Shariati M, Suhatril M, Trung NT, Wakil K, Khorami M (2020) Development of neuro-fuzzy and neuro-bee predictive models for prediction of the safety factor of eco-protection slopes. Phys A Stat Mech Appl 550:124046

66. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv:1707.06347
67. Shah D, Zhang K (2014) Bayesian regression and bitcoin. In: 2014 52nd annual Allerton conference on communication, control, and computing (Allerton), pp 409–414. IEEE
68. Smailović J, Grčar M, Lavrač N, Žnidaršič M (2014) Stream-based active learning for sentiment analysis in the financial domain. Inf Sci 285:181–203. https://doi.org/10.1016/j.ins.2014.04.034
69. Snashall D, Cliff D (2019) Adaptive-aggressive traders don't dominate. In: International conference on agents and artificial intelligence, pp 246–269. Springer
70. Sugeno M (1985) Industrial applications of fuzzy control. Elsevier, Amsterdam
71. Talarposhti FM, Sadaei HJ, Enayatifar R, Guimarães FG, Mahmud M, Eslami T (2016) Stock market forecasting by using a hybrid model of exponential fuzzy time series. Int J Approx Reason 70:79–98
72. Tesauro G, Bredin JL (2002) Strategic sequential bidding in auctions using dynamic programming. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2, pp 591–598
73. Vach D (2015) Comparison of double auction bidding strategies for automated trading agents
74. Vytelingum P, Cliff D, Jennings NR (2008) Strategic bidding in continuous double auctions. Artif Intell 172(14):1700–1729
75. Wang J, Zhang Y, Tang K, Wu J, Xiong Z (2019) Alphastock: a buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD '19, pp 1900–1908. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3292500.3330647
76. Wang Z, Huang B, Tu S, Zhang K, Xu L (2021) Deeptrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. Proc AAAI Conf Artif Intell 35(1):643–650
77. Wilson RB (1987) On equilibria of bid-ask markets. In: Arrow and the ascent of modern economic theory, pp 375–414. Springer
78. Wu X, Chen H, Wang J, Troiano L, Loia V, Fujita H (2020) Adaptive stock trading strategies with deep reinforcement learning methods. Inf Sci 538:142–158
79. Ye Y, Pei H, Wang B, Chen PY, Zhu Y, Xiao J, Li B (2020) Reinforcement-learning based portfolio management with augmented asset movement prediction states. Proc AAAI Conf Artif Intell 34(01):1112–1119. https://doi.org/10.1609/aaai.v34i01.5462
80. Yu Y, Duan W, Cao Q (2013) The impact of social and conventional media on firm equity value: a sentiment analysis approach. Decis Support Syst 55(4):919–926. https://doi.org/10.1016/j.dss.2012.12.028

Authors and Affiliations

Alfonso Guarino[1] · Luca Grilli[2] · Domenico Santoro[3] · Francesco Messina[4] · Rocco Zaccagnino[5]

✉ Rocco Zaccagnino
rzaccagnino@unisa.it

Alfonso Guarino
alfonso.guarino@unifg.it

Luca Grilli
luca.grilli@unifg.it

Domenico Santoro
domenico.santoro@uniba.it

Francesco Messina
messina.1961177@studenti.uniroma1.it

[1] Department of Humanities, University of Foggia, Via Arpi, 176, 71121 Foggia, FG, Italy

[2] Department of Economics, Management and Territory, University of Foggia, Via da Zara, 11, 71121 Foggia, FG, Italy

[3] Department of Economics and Finance, University of Bari, Largo Abbazia S. Scolastica, 70124 Bari, BA, Italy

[4] Department of Statistical Sciences, University La Sapienza of Rome, Piazzale A.Moro 5, 00185 Rome, RM, Italy

[5] Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 130, Fisciano, SA, Italy