European Journal of Operational Research 276 (2019) 770–780

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Interfaces with Other Disciplines

Bitcoin price forecasting with neuro-fuzzy techniques

George S. Atsalakis[a], Ioanna G. Atsalaki[a], Fotios Pasiouras[b], Constantin Zopounidis[a,c,*]

[a] *School of Production Engineering & Management, Technical University of Crete, University Campus, 73100 Chania, Greece*
[b] *Montpellier Business School, 2300 Avenue des Moulins, Cedex 4, 34185 Montpellier, France*
[c] *Audencia Business School, 8 Route de la Jonelière, B.P. 31222, Cedex 3, 44312 Nantes, France*

ARTICLE INFO

Article history:
Received 11 May 2018
Accepted 16 January 2019
Available online 19 January 2019

Keywords:
Artificial intelligence
Fuzzy sets
Neuro-fuzzy forecasting
Bitcoin price forecasting

ABSTRACT

Cryptocurrencies, with Bitcoin being the most notable example, have attracted considerable attention in recent years, and they have experienced large fluctuations in their price. While a few studies employ conventional statistical and econometric approaches to reveal the driving factors of Bitcoin's prices, research on the development of forecasting models to be used as decision support tools in investment strategies is scarce. This study proposes a computational intelligence technique that uses a hybrid Neuro-Fuzzy controller, namely PATSOS, to forecast the direction in the change of the daily price of Bitcoin. The proposed methodology outperforms two other computational intelligence models, the first being developed with a simpler neuro-fuzzy approach, and the second being developed with artificial neural networks. Furthermore, the investment returns achieved by a trading simulation, based on the signals of the proposed model, are 71.21% higher than the ones achieved through a naive buy-and-hold strategy. The performance of the PATSOS system is robust to the use of other cryptocurrencies.

1. Introduction

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. Policy makers 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.[1] Nonetheless, investors appear to be attracted by the potential to earn high returns, the introduction of Bitcoin derivatives, and the potential diversification benefits.

The present study aims to develop, for the first time in the literature, a neuro-fuzzy model to forecast the direction of the Bitcoin price changes.[2] Our work relates to various strands found in the literature, and in the discussion that follows, we outline how our research differs from each one of these groups of studies.

First, our work relates to the existing studies on the determinants of Bitcoin returns. However, most of these studies are of an explanatory nature. More precisely, using econometric-based techniques (e.g. vector autoregressive -VAR, vector error correction - VEC, ordinary least squares, quantile regression), their aim is to reveal the driving factors of the Bitcoin exchange rate and returns, such as market sentiment (Makrichoriti & Moratis, 2016), Twitter sentiment ratio, Wikipedia searches and mining difficulty (Georgoula, Pournarakis, Bilanakos, Sotiropoulos, & Giaglis, 2015), economic policy uncertainty (Demir, Gozgor, Lau, & Vigne, 2018), Bitcoin popularity, number of transactions, sentiment in newspaper reports (Polasik, Piotrowska, Wisniewski, Kotkowski, & Lightfoot, 2015), and economic factors like the consumer price index and the

[*] Corresponding author at: School of Production Engineering & Management, Technical University of Crete, University Campus, 73100 Chania, Greece.
E-mail address: kostas@dpem.tuc.gr (C. Zopounidis).

[1] Some academic studies tend to support this argument. For example, Cheah and Fry (2015) find that the Bitcoin exhibits speculative bubbles, with its fundamental price being zero. Kristoufek (2015) concludes that the Bitcoin forms a unique asset possessing properties of both a standard financial asset and a speculative one. Additionally, Bouri, Shahzad, and Doubaud (2019) find that the likelihood of explosive periods in one cryptocurrency generally depends on the presence of explosivity in other cryptocurrencies.

[2] Forecasting the 'Bitcoin price direction' – i.e. forecasting Bitcoin price changes or rate – is preferred over forecasting the 'Bitcoin absolute values'. The rationale is that it is impossible to forecast future absolute values of Bitcoin on a daily basis, especially since the Bitcoin market is a memoryless system. However, it is postulated, based on obtained results from real case studies, that with appropriate training over any direction (uptrend, downtrend, flat), one has enough indicators to forecast direction with significant accuracy. Our approach is consistent with papers on both the stock market (e.g. Atsalakis & Valavanis, 2009) and the Bitcoin market (e.g. Greaves & Au, 2015; Madan et al., 2015).

https://doi.org/10.1016/j.ejor.2019.01.040
0377-2217/© 2019 Elsevier B.V. All rights reserved.

US dollar index (Zhu, Dickinson, & Li, 2017).[3] Others focus on the association between Bitcoin prices and trading volume (Balcilar, Bouri, Gupta, & Roubaud, 2017), Bitcoin attractiveness (Kristoufek, 2015), gold and aggregate commodity prices (Bouri, Gupta, Lahiani, & Shahbaz, 2018), and energy prices (Bouri, Jalkh, Molnár, & Roubaud, 2017a; Hayes, 2017). [4] In general, the studies discussed above focus on the influence of specific factors on the Bitcoin price, and not on the predictive or classification ability of the models.

In contrast, we aim to develop a forecasting model that could be used by investors as a decision support tool. For example, the proposed neuro-fuzzy model produces trading signals that could contribute to the minimization of losses and maximization of returns as the result of appropriate trading actions during Bitcoin price declines and increases. In general, the development of a forecasting model is a more challenging task than the estimation of an explanatory model. The main reason for this is that the out-of-sample and out-of-time prediction accuracy of the model, and its ability to generate investment returns, become of paramount importance.

This brings us to the second strand of the literature that relates to our work. This consists of a handful of studies that develop and test prediction models for the Bitcoin market. For example, Shah and Zhang (2014) propose a trading strategy based on a Bayesian regression model that allows them to earn important returns when tested on real data. In a similar context, Madan, Saluja, and Zhao (2015) propose the use of binomial regressions, support vector machines and random forest algorithms to predict the sign of the Bitcoin price change. Finally, using machine learning optimization, Greaves and Au (2015) obtain an up-down Bitcoin price movement classification accuracy of roughly 55%. To the best of our knowledge, neuro-fuzzy models have not been employed in the Bitcoin literature, and we aim to close this gap. As discussed below, neuro-fuzzy models have several characteristics that make them particularly suitable to the problem in hand.

More broadly, our work also relates to studies that use fuzzy models to forecast stock market prices and commodities prices with promising results. Among other things, these studies successfully forecast stock prices (Atsalakis & Valavanis, 2009; Atsalakis, Dimitrakakis, & Zopounidis, 2011), stock market index performance (Rubio, Bermudez, & Vercher, 2017; Talarposhti et al., 2016), carbon emissions futures prices (Atsalakis, 2016), the price of gold (Habibnia, 2010), the price of oil (Azadeh, Moghaddam, Khakzad, & Ebrahimipour, 2012) and the price of various commodities like coffee, cocoa, etc. (Atsalakis, Fratzis, & Zopounidis, 2016). However, none of these studies relates to the Bitcoin market, and we aim to bring together all these strands of the literature.

The above studies highlight several desirable characteristics of the neuro-fuzzy models that make them ideal candidates for the prediction of Bitcoin price movements, which motivated their use in the present study. For instance, neuro-fuzzy models learn the rules and adjust the values of membership functions from the data without the need to use domain experts or knowledge acquisition. Additionally, they can capture non-linear relationships and imprecise, ambiguous, vague information. In a nutshell, the power of neuro-fuzzy models lies in their ability to perform meaningful and reasonable operations on concepts that are outside the definitions available in conventional Boolean logic**.** Despite these attributes and the good performance of neuro-fuzzy models in the financial and commodity markets, 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 (Baur, Hong, & Lee, 2018), and it is particularly sensitive to regulatory and market events (Feng, Wang, & Zhang, 2018). Additionally, prices may be manipulated through suspicious trading activity (Gandal, Hamrick, Moore, & Oberman, 2018). Finally, some argue that Bitcoin prices contain a substantial speculative bubble component and have no fundamental value (Cheah & Fry, 2015). Therefore, one cannot argue that forecasting models performing well in other markets can work equally well in the Bitcoin market, unless they are empirically tested.

In the present study, we propose the use of a hybrid neuro-fuzzy controller system, namely PATSOS, consisting of two Adaptive Neuro-Fuzzy Inference System (ANFIS) sub-systems. To assess the forecasting ability of the model, we benchmark it against: (i) a simpler model developed with a single ANFIS, (ii) a model developed with artificial neural networks (ANN), and (iii) a naive buy-and-hold (B&H) approach. The results show that the PATSOS neuro-fuzzy model outperforms both the ANFIS and the ANN models by generating higher returns for a potential investor. We confirm this finding when we use three alternative cryptocurrencies, namely Ethereum, Litecoin, and Ripple.

The rest of the paper is organized as follows. Section 2 provides a background discussion of the methodologies applied in the present study. Section 3 describes the dataset and discusses the empirical results. Section 4 provides further discussion, and Section 5 concludes the paper.

2. Methodological framework

In this section we provide a brief discussion of the theoretical background of fuzzy logic, neural networks, and the ANFIS neuro-fuzzy model, along with a more detailed description of the PATSOS neuro-fuzzy inverse controller.

2.1. Fuzzy logic

Zadeh (1965) proposed the Fuzzy Logic framework as an extended version of the Aristotelian binary logic. In more detail, fuzzy logic is an approach to computing, based on degrees of truth rather than the usual dichotomous true or false denoted by 1 or 0. While fuzzy logic includes 0 and 1, it considers them as extreme cases. Its advantage, as an approach, is that it also includes various states of truth between the two extremes. For example, the results of a comparison between two things should not necessarily be high or low but it could also be 0.70 of being high. This resembles the method of human reasoning that includes intermediate possibilities like certainly yes, possibly yes, etc.[5]

One key concept underlying fuzzy logic is that of the fuzzy if-then rule. Although rule-based systems are common in the field of artificial intelligence (AI), such AI approaches do not handle fuzzy consequents and fuzzy antecedents. In fuzzy logic, this is accomplished by the calculus of fuzzy rules.[6] Using these “if-then” rules, the fuzzy inference system (FIS) can model qualitative aspects of

[3] Georgoula, Pournarakis, Bilanakos, Sotiropoulos, and Giaglis (2015) also employ a state-of-the art machine learning technique. However, they only use it for the purposes of sentiment analysis in the first stage of their work. Then, they rely on OLS regressions and a vector-error correction model to examine the driving factors of the Bitcoin price.

[4] Other academic studies explore issues like: Bitcoin’s banning (Hendrickson & Luther, 2017), informed trading (Feng, Wang, & Zhang, 2018), speculative bubbles (Cheah & Fry, 2015), transaction costs (Kim, 2017), the market’s informational efficiency (Tiwari, Jana, Das, & Roubaud, 2018), diversification benefits (Brière, Oosterlinc, & Szafarz, 2015), the use of Bitcoin as a hedge instrument, medium of exchange and speculative asset (Baur, Hong, & Lee, 2018; Bouri et al., 2017b).

[5] One can say that the human brain aggregates data and forms a number of partial truths. These are aggregated further into higher truths which in turn, when certain thresholds are exceeded, lead to further results.

[6] Ruan and Kerre (2000) present a detailed overview of if-then rules in computational intelligence.

human knowledge and reasoning processes without using quantitative data or analysis.[7] Therefore, another key concept of fuzzy logic is that of a linguistic variable, whose values are words rather than numbers.

To use some basic notation, the fuzzy decision rules are the way a FIS relates an input variable x to an output variable y. In the case where more than one variable is involved on the premise side, the structure of the rule takes the form:

if x_1 is A and x_2 is B, then y is Z

where x_1 and x_2 are the input variables and A, B and Z are linguistic values (small or big etc.) defined as a membership function (MF) in the input and output spaces.

In fuzzy modelling, the membership functions and rule base are generally determined by trial-and-error approaches. Although this approach is straightforward, the determination of best-fitting boundaries of membership functions and the number of rules are demanding tasks.

In general, the steps to create a fuzzy inference model are as follows:

i) Fuzzification: the input variables are compared with the MFs on the premise part of the fuzzy rules to obtain the probability of each linguistic label.
ii) Combination (through logic operators) of the probability on the premise part to get the weight (fire strength) of each rule.
iii) Application of firing strength to the premise MFs of each rule to generate the qualified consequent of each rule depending on their weight.
iv) Defuzzification: Aggregate the qualified consequents to produce a crisp output.

To sum up, fuzzy logic is useful in diverse practical applications related to control, forecasting and inference. Its key advantages are: (i) it shows simplicity and a natural structure, (ii) it is based on analogical or ratio sets that have an infinite number of points between zero and one (Hiebert, 2008) rather than binary oppositions, (iii) it models uncertainties, and (iv) it can be efficiently combined with other intelligence methods to form hybrid models. Those interested in more detailed information may refer to Zimmerman (1991) and Ross (1995).[8]

2.2. Neural networks

Neural networks can be applied to various financial problems, like forecasting of yield curves, exchange rates, bond rates, stocks, commodities, etc. The principal motivation for their use in market forecasting is twofold: (i) market data are highly complex and hard to model; therefore, a non-linear model is beneficial, and (ii) a large set of interacting input series is often required to explain a specific market, which is well suited to the use of neural networks.

The underlying idea of the ANN models is that of the brain that can learn in the presence of a teacher. Therefore, during the learning stage, the “teacher” specifies the right responses to the input examples. Also, ANN can learn without a teacher, based on principles of self-organization (Kacprzyk and Pedrycz, 2015). There are two kinds of neurons that represent fundamental building blocks of an ANN: input and computational ones. The input neurons are located in the input layer and they mediate network inputs and computational neurons. Computational neurons are located within the hidden layers of the system and multiply their inputs with the associated synaptic weights, thereby calculating the sum of the product. Each neuron operates independently and performs a predetermined mathematical operation to produce a single output. The resulting sum is the argument of the activation function. ANNs capture the relationships among the data sets and use these to forecast the future prices. In addition, they use the actual results as they become available, to adjust the model and end up with a more accurate forecasting of future prices. In fact, they learn from their mistakes. Therefore, over time as they continue to accumulate data and feedback, the neural networks will retest and optimise the model to achieve more accurate forecasts.

In the present paper, we develop a model based on a feedforward neural network with a single hidden layer. Given the training data $A = \{(x_{i,} t_i)\}_{i=1}^{N}$, the output function of the single hidden layer feedforward neural network with L hidden neuros can be written as:

$$f(x_i) = \sum_{j=1}^{L} \beta_j h_j(\mathrm{a}_j, b_j, x_i) = \mathrm{h}(x_i)\beta, \mathrm{i} = 1, \ldots, \mathrm{N} \tag{2}$$

where: $\beta = [\beta_1, \ldots, \beta_L]^T$ is the output weighted matrix, $\mathrm{h}(x_i) = [h_1(\mathrm{a}_1, b_{1,}x_i), \ldots, h_j(\mathrm{a}_L, b_L, x_i)]$ is the network output associated with the training sample, x_i, $h_j(\cdot)$ is a nonlinear piecewise continuous function, and $\mathrm{a}_j \in R^d$ and $\beta_j \in R$ $(j = 1, 2, \ldots, L)$ are parameters of the jth hidden node, respectively.

The aim of the training is to find the parameters that minimize the error function $H\beta - \mathrm{T}_2$, where the hidden layer output matrix and the target output, respectively, are given by:

$$H = \begin{bmatrix} h(x_1) \\ . \\ . \\ . \\ h(x_N) \end{bmatrix}, \quad T = \begin{bmatrix} t_1^T \\ . \\ . \\ . \\ t_N^T \end{bmatrix} \tag{3}$$

Analytical details related to the development and implementations of an ANN model are available in, among others, Gallant (1993), Hagan and Menhaj (1994), Haykin (1994), and Kacprzyk and Pedrycz (2015).

2.3. Adaptive Neuro-Fuzzy Inference System (ANFIS)

Neuro-fuzzy systems are hybrid models that combine the functionality of fuzzy systems with the learning abilities of neural networks. Consequently, one of the main advantages of a neuro-fuzzy system is its ability to 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 fuzzy logic has also been applied in the development of controllers which have the following advantages (Mamdani & Asilian, 1975): (i) they do not require the model of the plant to be known exactly, and (ii) their linguistic form is closer to the way human reasoning is expressed and formalized.

The use of the fuzzy if-then rules that can be used due to the universal approximating characteristics of the fuzzy logic allow the modelling of the nonlinearities of the system. However, an approximate model of the system is used since it is not possible to capture all the possible cases. The main challenge in designing fuzzy controllers is to define the parameters of the controller. Therefore, many techniques have been proposed for autotuning of the parameters of the controllers in batch mode, mostly using off-line learning techniques from the area of neural networks or genetic algorithms (Pomares et al., 2002). Kacprzyk and Pedrycz (2015) presents an overview of some adaptive and evolving control approaches.

The adaptive network-based fuzzy inference system (ANFIS) used in the present study was proposed by Jang (1993). It consists

[7] A fuzzy inference system (FIS) is a computing framework that combines the concepts of fuzzy logic, fuzzy decision rules, and fuzzy reasoning.

[8] Bellman and Zadeh (1970) also provide a very insightful discussion of fuzzy sets, and three basic concepts: fuzzy goal, fuzzy constraint, and fuzzy decision.

of five layers of adaptive network with two inputs, *x* and *y*, and one output *z*; due to its self-learning capabilities, it can be trained without the need for expert knowledge. Rules allow for the analysis of both qualitative and quantitative data and are sufficiently simple to facilitate the reasoning behind the model's results. In addition, ANFIS allows as inputs some 'fuzzified' values (e.g. small, medium, large) rather than a wide range of real numerical values, thereby reducing the amount of time the system needs to learn and adapt (Jang, 1993).

In the present paper, to identify the Sugeno-type fuzzy inference systems parameters (Sugeno, 1985), 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 $\mu_A(x) \in [0, 1]$ 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). A widely utilised membership function (MF) is the generalised bell MF shown in Eq. (4), where the three adaptive parameters {a, b, c} describe the bell function:

$$\mathrm{gbellmf}(x, \mathrm{a}, b, c) = \frac{1}{1 + \left|\frac{x-c}{\mathrm{a}}\right|^{2b}} \tag{4}$$

The back-propagation algorithm adapts the respective values using the learning algorithm. Changes of the linguistic term are contingent upon changes in the values of the other parameters. Readers interested in a more comprehensive discussion of the ANFIS function might be referred to Jang (1997).

2.4. The neuro-fuzzy controller forecasting system PATSOS

The main approach that we propose in the present study for the forecasting of Bitcoin prices, is the PATSOS neuro-fuzzy controller forecasting system. Initially suggested by Atsalakis and Valavanis (2009) in the prediction of stock prices, this system consists of two ANFIS sub-systems, forming a closed-loop feedback system. The first subsystem is an ANFIS model that forms an inverse learning neuro-fuzzy controller referred to as CON-ANFIS. The second sub-system, called PR-ANFIS, models the process (i.e. Bitcoin price forecasting) to be controlled. This dual system has been successfully applied to the forecasting of stock market and commodities movements. In the present paper, it is appropriately modified and applied for the first time in the forecasting of Bitcoin prices.

To develop a controller that is trained by an inverse learning technique requires a learning and an application phase, also known as general learning. These two phases function in parallel (Jang, 1997). In the first phase, an off-line technique is used to model the inverse dynamics of the process (Bitcoin pricing). In the application phase, the obtained neuro-fuzzy controller represents the inverse dynamics of the process. This is used to generate control actions that drive the Bitcoin pricing process (PR-ANFIS) to produce forecasts. This framework perfectly complements classical adaptive control schemes. Fig. 1 presents the PATSOS controller during the training phase. In the application phase, the CON-ANFIS produces a control action that leads the PR-ANFIS model to make forecasts related to Bitcoin prices.

2.4.1. Training the CON-ANFIS controller

The CON-ANFIS controller produces control actions for the PR-ANFIS process model that forecasts the Bitcoin price trend one step ahead (i.e. it forecasts the following day's trend). In discrete time, the overall system process and controller equations apply to both phases:

$$y(k+1) = f(y(k), u(k)) \tag{5}$$

$$u(k) = g(y(k)) \tag{6}$$

where $y(k+1)$ represents the Bitcoin price at time $k+1$, $y(k)$ signifies the Bitcoin price at time k, and $u(k)$ is the control signal (action) at time k.

It is assumed that the process dynamics are unknown and meant to build a first-order ANFIS that maps a given input pair $[y(k), y(k+1)]$ to a desired control action $u(k)$. The CON-ANFIS model is a Sugeno first-order model consisting of two inputs, $y(k+1)$ and $y(k)$, and one output. To train the CON-ANFIS, data of the form [*y(k), y(k+1), u(k)*] are used. The model's inputs are: (i) the current trend of a Bitcoin price $y(k)$, and (ii) the next day's actual trend $y(k+1)$. The output for this model is the control action $u(k)$ that is used as input for the PR-ANFIS model, leading to the trend $y(k+1)$. During the training, $u(k)$ remains a positive control action calculated as:

$$u(k) = \sqrt{(y(k) - y(k+1))^2} \tag{7}$$

Following a trial and error procedure, we choose two Gaussian membership functions that correspond to the linguistic terms: *small* and *big*. Combining the two inputs with the two membership functions results in the following four rules:

if $y(k+1)$ is small and $y(k)$ is big, then (u) is $f_1 = p_1 \cdot y(k+1) + q_1 \cdot y(k) + r_1$,

where $\{p_i, q_i, r_i\}$ is a parameter set with values calculated and optimised during the learning phase.

2.4.2. Training the PR-ANFIS process

The change in the Bitcoin prices is nonlinear, making the use of identification techniques necessary for an effective modelling. The system identification is based exclusively on measured data, and ANFIS is thus used to model the process (i.e. Bitcoin pricing). Further, PR-ANFIS is trained to generate forecasts that are one step ahead of the process output. This is a Sugeno first-order model with three inputs and one output. The inputs are: (i) current price $y(k)$, (ii) one-step delayed price $y(k-1)$, and (iii) the control action of the output of the CON-ANFIS controller, $u(k)$. The model's output is $y(k+1)$. To train the PR-ANFIS, we use training data pairs of the following form: $[y(k-1), y(k), u(k); y(k+1)]$.

In addition, two Gaussian membership functions are chosen. These correspond to the linguistic terms *small* and *big* for each input, thereby yielding eight rules such that:

if $y(k-1)$, is small and $y(k)$, is small and (u) is small, then $y(k+1)$ is $f_1 = p_1 \cdot y(k+1) + q_1 \cdot y(k) + s_1 \cdot u(k) + r_1$,

where the $\{p_i, q_i, s_i, r_i\}$ parameters are optimised during model training.

During the training, the model is designed to forecast one-step-ahead Bitcoin prices. According to Norgaard, Ravnm, and Poulsen (2003), since the CON-ANFIS controller input $y(k+1)$ is unknown during the evaluation, the desired input $y_d(k+1)$ is used instead.

In the proposed methodology, $y_d(k+1)$ is defined as the change rate for the three-day Bitcoin price moving average. Eqs. (8) and (9) illustrate the calculation of the three-day moving average and the corresponding rate of change, respectively:

$$\mathrm{SMA}(k) = \frac{\text{Sum of close price, day}k, k-1, k-2}{3} \tag{8}$$

$$\text{moving average rate} = \frac{\mathrm{SMA}(k) - \mathrm{SMA}(k-1)}{\mathrm{SMA}(k-1)} \tag{9}$$

During the out-of-sample evaluation, the inputs for the CON-ANFIS are: (i) the three-day moving average rate (because the out-of-evaluation input $y(k+1)$ used during the training is unknown,

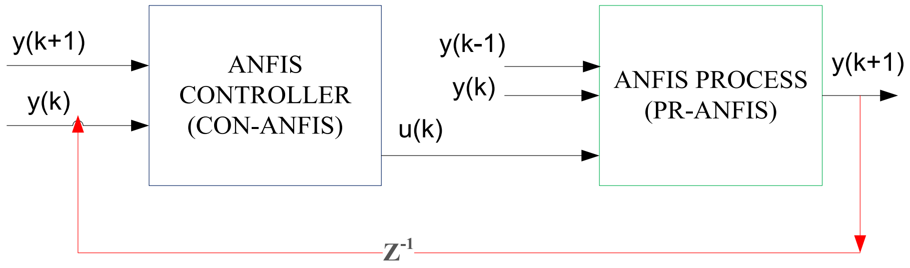

Fig. 1. PATSOS forecasting system during the training phase.

$y_d(k+1)$) is used instead), and (ii) the current price $y(k)$. The CON-ANFIS controller drives the PR-ANFIS model output at time $k+1$ (Atsalakis & Valavanis, 2009).

3. Forecasting Bitcoin prices

3.1. Data description

In the present study, we use daily historical time series data of Bitcoin closing prices as raw inputs to the model, to forecast the direction in the change of the price. Data are from the period September 13, 2011 to October 12, 2017, a total of 2201 observations. All the data used in the modeling process were acquired from "bitcoincharts.com". The prices are those which predominate on the Mt. Gox2 exchange at 12:00 a.m. GMT (UTC).

Data were divided into two subsets: a training set, and a validation set. There are several methods for splitting the sample. Many authors split the data based on convenience, for instance: 50:50, 60:40, 70:30, 80:20 or 90:10. In the present paper, approximately 97% of the dataset (2141 observations) is used for training the model, and the remaining 3% (60 observations) is used for validation purposes. The number of 60 observations corresponds to a three-month investment-trading horizon (excluding the weekends). This period is suggested as a benchmark period for evaluating a trading strategy, and it is similar to the one used in other studies (Atsalakis, 2009, 2016; Shah & Zhang, 2014). It should be emphasized here that the validation dataset is out-of-time – i.e. the 60 observations are from a future period compared to those used for training purposes. There is no doubt that 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.[9]

Based on the above split of the data, the models are trained using the training set, and their ability to forecast is assessed using the validation set. Prior to each training session, relevant data series were transformed and preprocessed to calculate the price change rate. Then, the lagged differences $u(k)$ were computed for each period such that $u(k) = y(k) - y(k\text{-}1)$, where $y(k)$ is any variable (either predictor or independent) observed in period k. Fig. 2 presents a graphical representation of a part (30 samples) of the training data for each input and output.

3.2. Forecasting performance measures

The assessment of the models is based on three frequently used measures, namely Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). The use of these measures aims to assess discrepancies between forecasts provided by the models and the real-world results (i.e. actual outcomes). The mathematical formulas are as follows:

$$\text{Root Mean Square Error RMSE} = \sqrt{\frac{\sum_{t=1}^{N} e_t^2}{N}} \qquad (10)$$

$$\text{Mean Square Error MSE} = \frac{1}{N} \cdot \sum_{t=1}^{N} e_t^2 \qquad (11)$$

$$\text{Mean Absolute Error MAE} = \frac{1}{N} \cdot \sum_{t=1}^{N} |e_t| \qquad (12)$$

Due to a lack of measures to illustrate the "magnitude" of price movements, a further evaluation is performed through a comparison of the results of PATSOS with a real-world outcome from a buy-and-hold strategy.

3.3. Neural network forecasting results

First, we develop the neural network model, against which we compare the performance of the proposed PATSOS neuro-fuzzy model. The underlying idea is that the popularity of neural networks in the forecasting of financial markets makes them an ideal benchmark for the proposed model.

A backpropagation ANN is defined by many factors, like the number of layers, the number of nodes and the activation function. In the present study we use a sigmoid function as the activation function in the hidden layer, while the output value is bounded between -1 and 1. Therefore, the input and output data are mapped to $[-1, 1]$. It is important not to have too many nodes in the hidden layer because the ANN may overlearn, or in other words, learn by example in the training sample but fail to generalize out-of-sample (Baum, 1989). To evaluate the effect of the input and hidden layer selection on the ANN model's operation, we adopt a procedure where many similar experiments are conducted. That is, 50 ANN are developed with up to 5 lagged inputs, and up to 10 hidden layers.

The results show that the ANN architecture that is structured with one lag and one hidden layer produces the lowest RMSE, MSE and MAE. Consequently, we use this model as the ANN benchmark against the PATSOS model. The results of the out-of-sample evaluation of this model are as follows: 0.0847 (RSME), 0.0072 (MSE), and 0.0476 (MAE). Thus, the errors appear to be low but there is still room for improvement. Possibly, this illustrates that the ANN has a low ability to effectively capture the nonlinear dynamic behaviour of Bitcoin prices.

[9] One 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 for two reasons. First, our sample size is sufficient, and it is possible to split it in the two datasets without particular concerns. Second, using a cross-validation approach would not allow us to take time into account. Currently, the model is developed until *t-1* and then tested in *t*. Using a cross-validation approach would not allow us to split the sample while taking this time dimension into account. Rather, both the training and validation datasets would include observations from the entire period of the study for which we have data. From a practitioner's point of view this is not realistic.

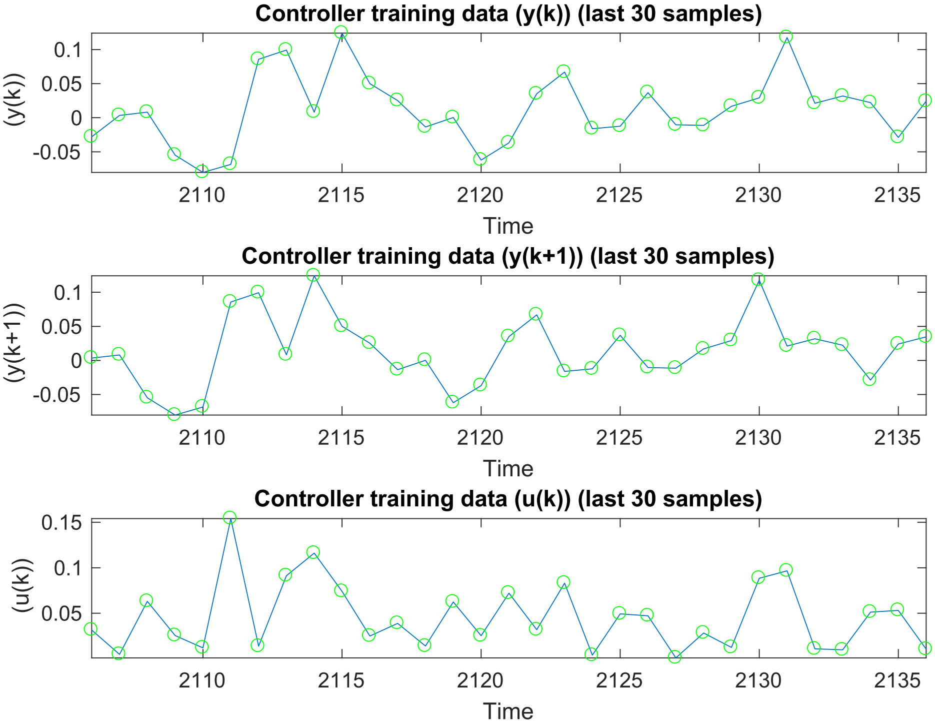

Fig. 2. Graphical representation of selected training datasets (rate of change).

3.4. Adaptive neuro-fuzzy inference system forecasting results

The implementation process of the ANFIS model consists of the following steps:

Step 1: Collect and process the data.
Step 2: Determine the training and validation datasets.
Step 3: Generate the initial first-order TSK fuzzy inference system through the fuzzy subtractive clustering:
  Step 3-1: Estimate cluster, and
  Step 3-2: Extract the rule and the initial rule parameters.
Step 4: Tune the antecedent and consequent parameters by the ANFIS hybrid learning algorithm. After setting the training parameters, the learning algorithm will continue until it reaches the specified error goal or the maximum number of epochs.
Step 5: Calculate the performance measures.
Step 6: Set the cluster radius and the step size and then run the model.
Step 7: Select the model with the minimum validation error.

By using cluster information and a neural learning methodology in the ANFIS structure, we establish the first-order TSK fuzzy inference system that models the behaviour of the data in a more efficient way. The hybrid ANFIS model is trained with the same input data used in the case of the ANN model. Once again, we perform several trial-and-error iterations using different MF types for each input variable. Specifically, the key types of MFs that we consider include the bell, gauss, gauss2, triangular, and trapezoid types. In addition, the algorithm considers several variations of the membership functions for each input variable. Two of them show the best fit to the data. More detailed, using two Gaussian membership functions for each input yielded the lowest errors, with the following results: 0.0489 (RMSE), 0.0023 (MSE), and (0.0402) MAE.

3.5. PATSOS neuro-fuzzy controller forecasting results

Finally, we use the same datasets to train and evaluate the main model of the present study, namely the PATSOS neuro-fuzzy controller. Fig. 3 presents the evolution of the Root Mean Square Error (RMSE) and the fluctuation in the step size according to the number of epochs used during the training phase. Initially, the step size increases when the error reaches 150 consecutive epochs. However, after a certain number of epochs, the step size decreases when the error measure reaches 250 consecutive combinations of an increase followed by a decrease. Finally, after 500 training epochs, there is a convergence to the RMSE curve. This means that any further training epoch up to 2000 epochs does not improve the RMSE. Incidentally, the same number of epochs is necessary to reach convergence in the PR-ANFIS.

Fig. 4 illustrates the initial shapes of the two membership functions of the CON-ANFIS. The first two graphs concern the pre-training phase and the last two depict the shape of the post-training membership functions. The graphs are for both inputs. Finally, Table 1 presents the basic parameters of the two ANFIS subsystems (CON-ANFIS and PR-ANFIS) that form the inverse controller. These parameters are the optimal outcome of the trial-and-error procedure.

During the trial-and-error phase, in order to choose the right type of membership function, three primary statistical measures are computed from the out-of-sample forecasts of each membership function. As described in the previous section, each series of forecasts is compared to the 60 actual cases in the holdout validation set. Table 2 summarises the performance errors of PATSOS for the following five types of membership functions: generalised bell, triangular, gauss2, gauss, and trapezoid. The same membership functions are utilised for both ANFIS sub-systems (CON-ANFIS and PR-ANFIS). The Gaussian membership function achieves the lowest RMSE (0.376), MSE (0.0014), and MAE (0.0307) and it is subsequently used to configure the final structure of the PATSOS system to be compared against the ANFIS and ANN models.

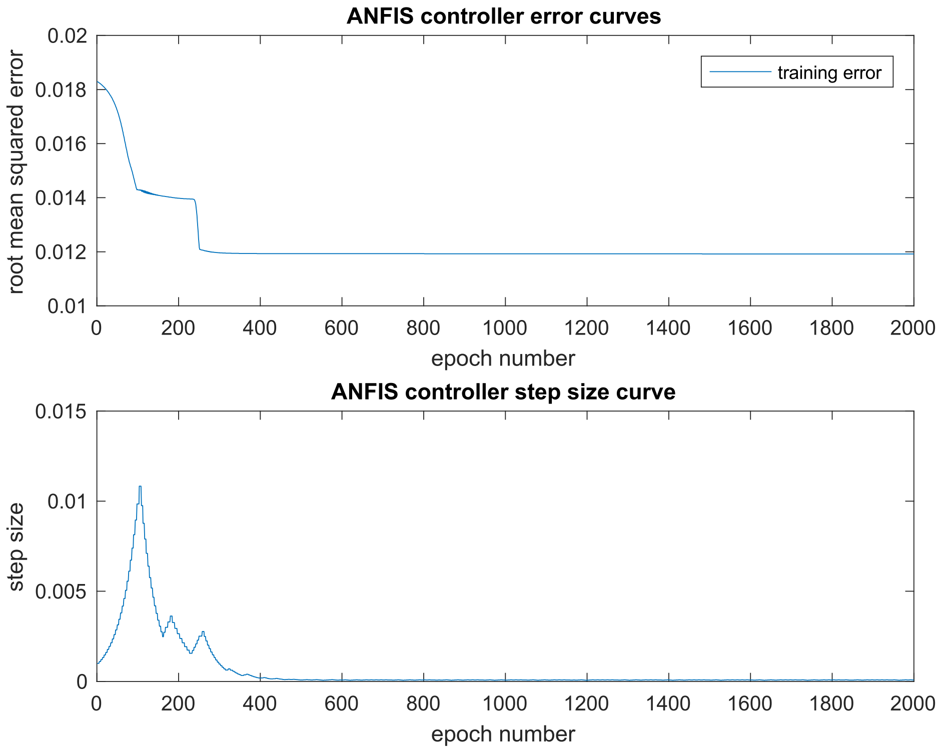

Fig. 3. The error curve and the step size during CON-ANFIS controller training.

Table 1
Parameter configuration of the two PATSOS subsystems.

Specifications	CON-ANFIS	PR-ANFIS
Inference mechanism type	Sugeno	Sugeno
Number of inputs	2	3
1st input	$y(k)$	$y(k\text{-}1)$
2nd input	$y(k+1)$	$y(k)$
3rd input	-	$u(k)$
Outputs	1	1
Output	$u(k)$	$y(k+1)$
And method	Product	Product
Or method	Max	Max
Imp. Method	Product	Product
Agg. Method	Max	Max
Defuzz. Method	Wtaver	Wtaver
Number Of Rules	4	8
Type of MFs	Gauss	Gauss
Number for training data	2136	2136
Number for evaluation data	60	60

Table 2
PATSOS forecasting results using various types of MFs.

Errors/type of MFs	Gbell	Triangular	Gauss2	Gauss	Trapezoid
RMSE	0.0403	0.0400	0.0393	0.0376	0.0397
MSE	0.0016	0.0016	0.0015	0.0014	0.0016
MAE	0.0320	0.0322	0.0312	0.0307	0.0319

Using the estimations of the PATSOS system, we can also forecast not only the trend in the price, but also the Bitcoin value. Fig. 5 depicts a scatter diagram of the actual Bitcoin values (asterisk in the red line) and the out-of-sample values estimated with the use of the PATSOS system (square signs in blue line). It is evident from Fig. 5 that the forecasted values and changes in price move in tandem with the actual ones, indicating that the developed model can imitate the short-time tendency of Bitcoin prices. Therefore, holding other things constant, this model provides the investors with an important forecasting tool.

Table 3
Performance comparison of competing models (out-of-sample).

	RMSE	MSE	MAE	Minutes
ANN	0.0847	0.0072	0.0476	0.56
ANFIS	0.0489	0.0023	0.0402	0.38
PATSOS	0.0376	0.0014	0.0307	1.13

3.6. Results of competing models

At this stage we provide a comparative discussion of the results of the three models. As previously mentioned, in all the cases the training and validation of the forecasting ability of the models is performed using the same input sets and the same validation set. However, the differences in the algorithms and the implementation process affect the time that each model requires for the forecasting. Therefore, Table 3 summarizes the statistical measures of performance along with the required time (in minutes) for the forecasting.

ANNs produce the highest error. It seems that despite their nonlinear nature, when applied individually, the ANNs cannot capture the nonlinear patterns inherent to Bitcoin prices with high volatility and irregularity. This is possibly due to the randomly set initial weights. The ANFIS model achieves a higher forecasting accuracy than ANN, but lower than PATSOS. The main advantage of ANFIS is that it is a hybrid model that combines the advantages of fuzzy and neural networks. In other words, methods that work synergistically may produce lower errors and therefore more accurate forecasts. In addition, by using the linguistic variables during the fuzzification phase of the input space, the ANFIS model manages to capture the underling non-linear relationships between the input and the output. Further, a hybrid intelligent system like ANFIS not only combines the learning capabilities of a neural network but it also incorporates reasoning by using fuzzy inference, thereby enhancing the forecasting capability of the system.

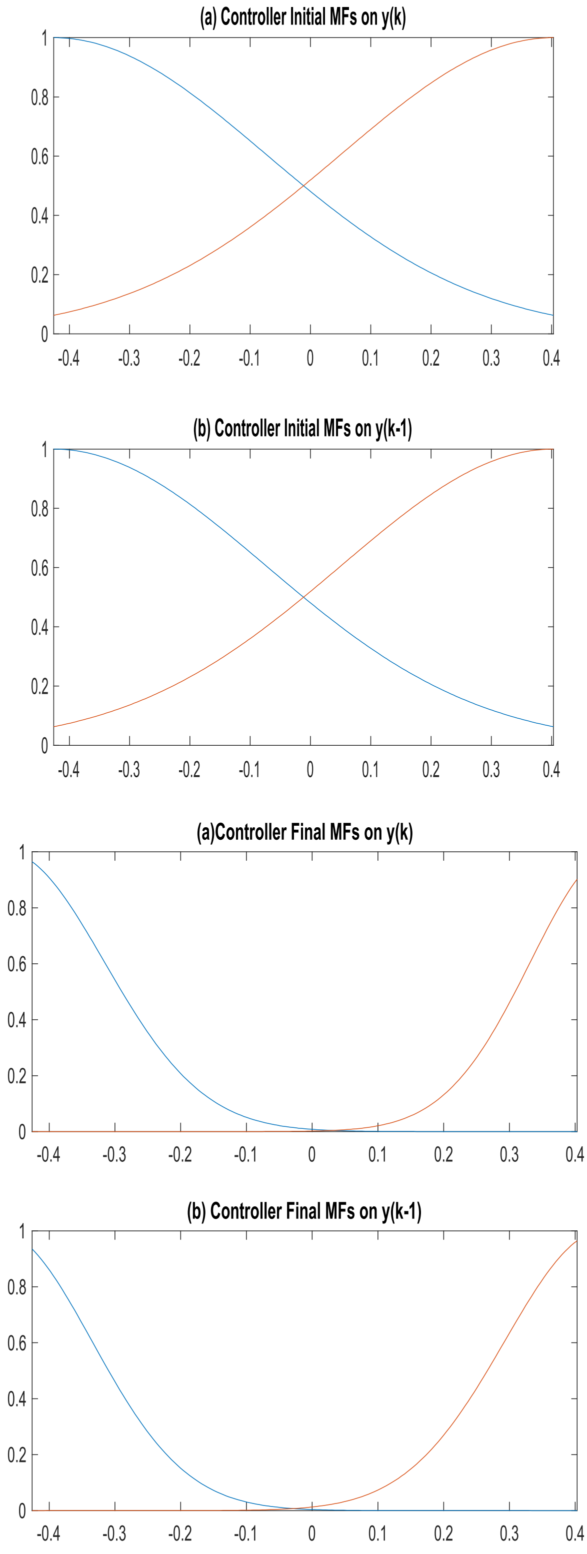

Fig. 4. Initial and post-training MF shapes of CON-ANFIS.

Table 4
Hit rate performance comparison.

	Hit rate (%)
ANN	55.10
ANFIS	57.70
PATSOS	63.22

The PATSOS model produces uniformly smaller out-of-sample forecast errors than both the ANN model and the ANFIS model. The superior forecasting performance of the PATSOS model implies that the prior information endorsed in the updating closed-loop process increases predictability. As mentioned before, the PATSOS architecture is composed of two ANFIS subsystems that form an inverse controller. Therefore, in addition to the ANFIS advantages discussed in the previous paragraph, the superior performance of PATSOS is also the result of the feedback mechanism achieved by the closed loop of the controller. The comparison strongly suggests that the correlations within the PATSOS returns are both significant and relevant. Therefore, it appears that due to its ability to capture the qualitative aspects of human reasoning and the decision-making process, this model generates superior forecasts.

Turning to the computational time, ANFIS requires 0.38 minutes to forecast, being the fastest model. ANN and PATSOS need 0.56 and 1.13 minutes, respectively, to produce the forecasts. The higher number of parameters of the two subsystems that must be tuned during the training phase make the estimations of PATSOS more time-consuming. Despite taking three times more than ANFIS, the required time of 1.13 minutes is rather short and acceptable as an absolute figure from a practical point of view. Therefore, the difference in terms of the computational time is not significant to alter our conclusion for the forecasting superiority of PATSOS.

3.7. *Hit rate and comparison of trading results against a buy-and-hold strategy*

In general, the performance of a forecasting model can be measured by its RMS error. For example, during the training of PATSOS, the aim was to minimize the RMS error while avoiding overtraining. However, the usefulness of the PATSOS Bitcoin forecasting system depends on its ability to accurately predict the direction in the change of the future Bitcoin prices. This can be formally assessed by the out-of-sample and out-of-time hit rate calculated with the following formula:

$$\text{Hit rate} = \frac{h}{n} \tag{10}$$

where h denotes the number of correct forecasts of the Bitcoin trend and n denotes the number of tests (i.e. 60 sessions in this case).

The output of PATSOS is categorized as a buy signal if it is greater than or equal to 0; otherwise, it is categorized as a sell signal. In more detail, at market closing time, the system produces a buy signal if it forecasts that the closing price of Bitcoin the next day will be higher than the closing price of the current trading day. Otherwise, it produces a sell signal. According to the aforementioned computation times, these trading signals can be calculated in almost one minute.

At this point we compare the forecast of PATSOS over the 60 seasons with the direction of the actual price movement. The objective of this analysis is to determine whether PATSOS can be used successfully as a decision support tool in real world trading with a cryptocurrency such as Bitcoin. Within this context, the investor allocates assets towards purchasing Bitcoin when the buy signal appears. Respectively, when a sell signal appears, the investor is expected to sell. For reasons discussed below, let us call this the PATSOS-driven investment strategy.

To be potentially useful as a trading decision support system, the overall hit rate (i.e. the percentage of accurate forecasts) must be higher than 50%. As shown in Table 4, the forecasting accuracy of the PATSOS system is 63.22%, which is higher than the corresponding figure of both the ANFIS (57.10%) and the ANN (55.10%). Thus, all the models produce a hit rate that is considerably higher

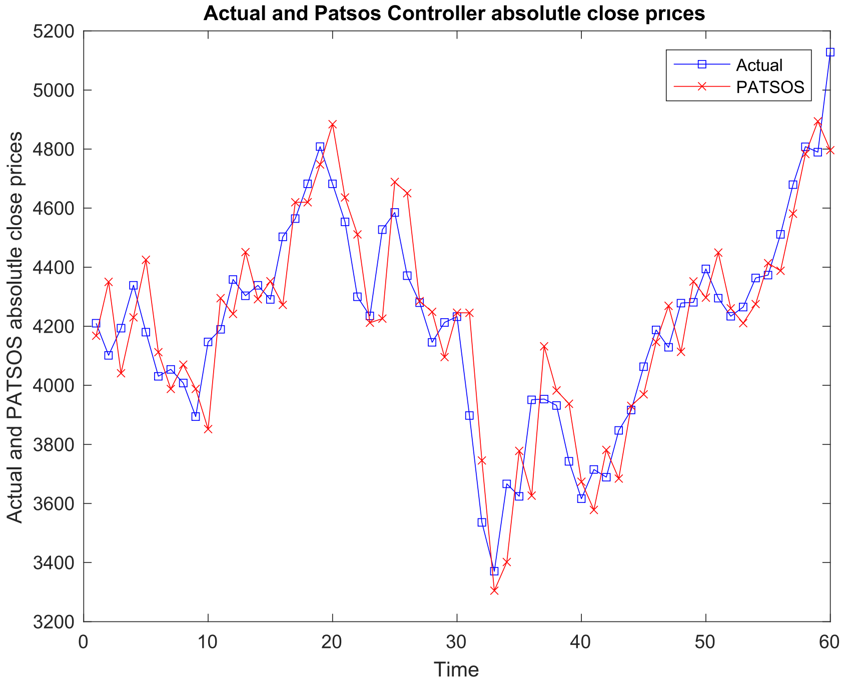

Fig. 5. Neuro-fuzzy forecasts versus actual prices for 60 out-of-sample prices (actual and PATSOS controller absolute close prices).

than the threshold of 50% that could be obtained by chance (i.e. flip of a coin).[10]

Before proceeding further, it is important to point out that PATSOS only gives trading signals; it does not inform the investor about how to trade. The investor must first devise some trading strategies to benefit from the forecasting ability of the proposed system. Therefore, for the purposes of the present study, we need a baseline naive trading strategy to use it as a benchmark for the effectiveness of the PATSOS-driven investment strategy. Assuming that the Efficient Market Hypothesis (EMH) holds, it is futile trying to make a profit by forecasting prices. Thus, the best strategy to compare the trading performance of the proposed system is the buy-and-hold one, under which the investor buys Bitcoins in the first session of the 60-session trading period and sells them in the last session. The performance measure to compare the two trading strategies is the investment Rate of Return (RoR) calculated as follows (11):

$$RoR = \frac{net\ gain\ from\ bitcoin}{initial\ investment} \tag{11}$$

Table 5 presents the rate of return achieved with the use of the out-of-sample forecasts over the 60-session investment horizon, assuming the allocation of an initial investment capital of €100,000 in the first buy signal. The results clearly demonstrate that the proposed system outperforms the naive B&H strategy. More precisely, an investor that is trading on the Bitcoin market following the PATSOS-driven investment strategy achieves a RoR of 37.34% (increasing the initial investment to €137,340) after a 60-session investment period. This represents a return that is approximately 1.72 times or 71.21% higher than the one achieved from a B&H strategy over the same period (i.e. (37.34-21.81)/21.81 = 71.21%).

Table 5
RoR trading against B&H strategy.

Investment strategies	Rate of Return (ROR)			
	Bitcoin	Ethereum	Litecoin	Ripple
PATSOS trading strategy	37.34 %	−5.76%	28.55%	112.45%
B&H strategy	21.81%	−8.53%	20.91%	58.23%
Performance difference	71.21%	32.47%	36.53%	93.11%

As a robustness test, Table 5 also presents the out-of-sample results when the PATSOS-driven investment strategy is used over the same period to trade on three other well-known cryptocurrencies, namely Ethereum, Litecoin, and Ripple. In all the cases, the PATSOS trading strategy outperforms the B&H strategy, providing support to our main finding.

As a final note, it should be mentioned that losses that result as an outcome of erroneously forecasted upward Bitcoin trends are not the same as those incurred because of erroneously forecasted downward trends. Higher returns can be obtained if the investor allocates the assets in alternative investment choices during the sessions that the forecasts produce sell signals. Finally, for simplicity reasons, we have ignored the transactions fees. However, the Bitcoin transaction fees have traditionally been low. For instance, the average Bitcoin transaction fees were as low as $0.005 at the beginning of our analysis in mid-September 2011 (associated with a Bitcoin price of around $6), increasing to $5.174 towards the end of the period in our study in mid-October 2017 (associated with a Bitcoin price in excess of $5000).

4. Discussion

Due to the high volatility of the markets (and in particular the Bitcoin market), forecasting the movements of prices or market indices can be a difficult task. Additionally, in finance, the efficient

[10] For example, let us say that an investor makes one trade per day by flipping a fair coin. If the coin comes up heads (tails), the investor bets all his money that the market will rise (fall). Therefore, each day, by flipping a fair and random coin, the investor has a 50% chance to be right about the market direction, a 50% chance to double his money. It should be emphasized here that: The first day, the investor's chance to succeed is 0.50. The second day, the investor's chance to succeed is also 0.50. Each following day, the investor's chance to succeed is the same: 0.50. Therefore, regardless of what has happened before, each new coin toss has a probability of 0.50 to come up heads. Having said that, the entire story can be different if we are interested in a sequence of events. For example, the probability that an investor will be right twenty times in a row is about 1 in a million. However, the probability to guess correctly in any one of the 20 tests remains 0.50.

market hypothesis asserts that in an efficient market, price movements are random and unpredictable, which means that abnormal profiting from predicting price movements is impossible (Fama, 1965).

The results of this paper support previous studies that succeed in the forecasting of market returns. The derived input variables of unanticipated economic and financial information that are incorporated in the Bitcoin prices allow the PATSOS model to generate returns that exceed the random walk returns. We assert that this is due to the use of the closed-loop controller feedback mechanism formed by the two ANFIS subsystems. Therefore, the proposed model challenges the weak form of EMH, demonstrating that historical data can forecast Bitcoin prices. This has been observed in various other markets (Atsalakis, 2016), and it seems that it applies to the Bitcoin market as well.

An attractive feature of the model is that the obtained results can be easily interpreted due to the 'if-then' rule knowledge base. This contrasts with other approaches, like for example the ANN one, frequently described as a 'black box' due to its incapability to explain how the final output was derived. In the case of the PATSOS system, all the synergies incorporated into the proposed system are successfully used to capture the nonlinear relationships of cryptocurrency prices. Additionally, the use of linguistic rules enhances the transparency of the decision-making process. Examples of such linguistic rules in our case are the following:

If input 1 is small and input 2 is small and input 3 is small, then the output is small
If input 1 is small and input 2 is small and input 3 is medium, then the output is small
If input 1 is small and input 2 is medium and input 3 is small, then the output is small
If input 1 is small and input 2 is medium and input 3 is medium, then the output is small
If input 1 is medium and input 2 is small and input 3 is small, then the output is medium
If input 1 is medium and input 2 is small and input 3 is medium, then the output is medium
If input 1 is medium and input 2 is medium and input 3 is small, then the output is medium
If input 1 is medium and input 2 is medium and input 3 is medium, then the output is medium

Initially, these linguistic rules result in the fuzzification of the real numbers of the inputs. Then, at the output level, a defuzzification of the fuzzy numbers takes place, transforming them back to real numbers (crisp numbers). Therefore, by utilizing fuzzy logic controllers in our model, we can capture the qualitative aspects of human reasoning and decision-making processes. The adaptive capabilities inherent to these models (which are introduced in conjunction with learning techniques in the neural networks domain) eliminate the need to use experts or knowledge acquisition methods to form the 'if-then' rules and the membership functions.

Finally, the superiority of the PATSOS system for Bitcoin forecasting, is consistent with the results of studies illustrating its use in other financial markets (Atsalakis, 2016). In a more general context, our findings also support studies illustrating the superiority of hybrid models (Mehdi & Mehdi, 2011; Taskaya & Casey, 2005; Yu et al., 2005; Zhu & Wei, 2013).

5. Conclusions

Bitcoin price forecasting models have only recently appeared, and despite their significance for market practitioners, the empirical work in the field is scarce. To fill this gap in the literature, three computational intelligence models have been employed in the present study. The proposed model, namely PATSOS, is an artificially intelligent, neuro-fuzzy controller incorporating a closed-loop. We benchmark its performance against a hybrid ANFIS model, and an artificial neural network model. The application of neuro-fuzzy models for the forecasting of Bitcoin price movements is proposed for the first time in the literature.

The use of the feedback mechanism of the inverse controller in the forecasting process improves the accuracy of the proposed model, making it superior to both the ANN and the ANFIS models. Additionally, when tested in an out-of-sample period, the PATSOS model improves the returns earned by a naive buy-and-hold investment strategy by 71.21%. We obtain similar results when testing the PATSOS model with data on three other well-known cryptocurrencies, namely Ethereum, Litecoin, and Ripple.

Therefore, the PATSOS system appears to be an efficient method to forecast Bitcoin prices. The buy-and-sell signals, produced by the forecasting system, minimize the risk from losses that may occur during the decline of the market due to high price volatility. Additionally, the easiness of dealing with the proposed forecasting system and the short computational time that is required encourage its adoption by end users. Furthermore, the PATSOS system estimates Bitcoin prices without the need to make any presumptions. It is a model-free, easily implementable approach. Similarly, the ANFIS subsystems implement a single-fitting procedure for nonlinear data that does not require the establishment of a formal model for the problem in hand. Therefore, there are no requirements for *a priori* information about the empirical relationship between the model's input and output. Furthermore, the results produced by the proposed model can be easily interpreted due to the 'if-then' rule knowledge base. In contrast, the ANN models have traditionally been characterized as 'black box models', given the difficulties of providing insights and explaining how the models produce the final output.

To conclude, motivated by the growth of the Bitcoin market, and the recent interest of market participants and academics, this study extends both the Bitcoin literature and the literature on fuzzy modelling by demonstrating that the use of a closed-loop or feedback control technique can cope with uncertainties associated with the dynamic behaviour of the price of Bitcoin, and achieve positive returns. Future attempts to improve the PATSOS model should focus on the development of a user-friendly interface to increase the ease with which the model can be utilised by Bitcoin investors. Also, accessing the PATSOS Bitcoin forecasting model online using cloud technology could be another important option. Moreover, in future work, the PATSOS model could be benchmarked against additional forecasting models from other disciplines.

Acknowledgments

Montpellier Business School (MBS) is a founding member of the public research center *Montpellier Research in Management, MRM* (EA 4557, Univ. Montpellier). We would like to thank Roman Slowinski (Editor) and three anonymous reviewers for various comments that helped us improve earlier versions of the manuscript. Any remaining errors are our own.

References

Atsalakis, G. (2016). Using computational intelligent to forecast carbon prices. *Journal of Applied Soft Computing, 43*, 107–116.

Atsalakis, G., & Valavanis, K. (2009). Forecasting stock Market short-term trends using a neuro-fuzzy based methodology. *Expert Systems with Applications, 36*, 10696–10707.

Atsalakis, G., Dimitrakakis, E., & Zopounidis, C. (2011). Elliot wave theory and neuro-fuzzy systems, in stock market prediction. The W.A.S.P. system. *Expert System with Application, 38*, 9196–9206.

Atsalakis, G., Fratzis, D., & Zopounidis, C. (2016). Commodities' price trend forecasting by a neuro-fuzzy controller. *Energy Systems, 7*, 73–102.

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. *Computers and Industrial Engineering, 62*, 421–430.

Balcilar, M., Bouri, E., Gupta, R., & Roubaud, D. (2017). Can volume predict Bitcoin returns and volatility? A quantiles-based approach. *Economic Modelling, 64*, 74–81.

Baur, D. G., Hong, K., & Lee, A. D. (2018). Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions & Money, 54*, 177–189.

Bellman, R. E., & Zadeh, L. A. (1970). Decision – Making in a fuzzy environment. *Management Science, 17*, B141–B164.

Bouri, E., Jalkh, N., Molnár, P., & Roubaud, D. (2017a). Bitcoin for energy commodities before and after the December 2013 crash: Diversifier, hedge or safe haven? *Applied Economics, 49*, 5063–5073.

Bouri, E., Gupta, R., Lahiani, A., & Shahbaz, M. (2018). Testing for asymmetric nonlinear short- and long-run relationships between Bitcoin, aggregate commodity and gold prices. *Resources Policy, 57*, 224–235.

Bouri, E., Gupta, R., Tiwari, A. K., & Doubaud, D. (2017b). Does Bitcoin hedge global uncertainty? Evidence from wavelet-based quantile-in-quantile regressions. *Finance Research Letters, 23*, 87–95.

Bouri, E., Shahzad, S. J. H., & Doubaud, D. (2019). Co-explosivity in the cryptocurrency market. *Finance Research Letters* In Press.

Brière, M., Oosterlinc, K., & Szafarz, A. (2015). Virtual currency, tangible return: Portfolio diversification with Bitcoin. *Journal of Asset Management, 16*, 365–373.

Cheah, E-T., & Fry, J. (2015). Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. *Economics Letters, 130*, 32–36.

Demir, E., Gozgor, G., Lau, C. K. M., & Vigne, S. A. (2018). Does economic policy uncertainty predict the Bitcoin returns? An empirical investigation. *Finance Research Letters, 26*, 145–149.

Fama, F. F. (1965). The behavior of stock market prices. *Journal of Business, 38*, 34–105.

Feng, W., Wang, Y., & Zhang, Z. (2018). Informed trading in the Bitcoin market. *Finance Research Letters, 26*, 63–70.

Gallant, S. L. (1993). *Neural network learning and expert systems*. Cambridge, MA: MIT Press.

Gandal, N., Hamrick, J. T., Moore, T., & Oberman, T. (2018). Price manipulation in the Bitcoin ecosystem. *Journal of Monetary Economics, 95*, 86–96.

Georgoula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D., & Giaglis, G. (2015). Using time-series and sentiment analysis to detect the determinants of Bitcoin prices. Available at 10.2139/ssrn.2607167.

Greaves, A., & Au, B.. Using the Bitcoin Transaction graph to predict the price of Bitcoin. Available at. (2015). http://snap.stanford.edu/class/cs224w-2015/projects_2015/.

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*. Available at https://ssrn.com/abstract=2010545.

Hagan, M. T., & Menhaj, M. (1994). Training feed forward networks with the Marquardt algorithm. *IEEE Transactions Neural Networks, 5*, 989–993.

Hayes, A. S. (2017). Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing Bitcoin. *Telematics and Informatics, 34*, 1308–1321.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall.

Hendrickson, J. R., & Luther, W. J. (2017). Banning Bitcoin. *Journal of Economic Behavior & Organization, 141*, 188–195.

Hiebert, P. G. (2008). Transforming worldviews: An anthropological understanding of how people change. Baker Academic.

Jang, J. (Ed.). (1997). *Neuro-fuzzy and soft computing. A computational approach to learning and machine intelligence*. Upper Saddle River, NJ: Prentice Hall.

Jang, S. (1993). ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans-actions on Systems, Man, and Cybernetics, 23*, 665–685.

Kacprzyk, J., & Pedrycz, W. (Eds.). (2015). *Springer handbook of computational intelligence (Part D and G)*. Springer.

Kim, T. (2017). On the transaction cost of Bitcoin. *Finance Research Letters, 23*, 300–305.

Kristoufek, L. (2015). What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. *PLoS ONE, 10*(4), E0123923. doi:10.1371/journal.pone.0123923.

Madan, I., Saluja, S., & Zhao, A. (2015). *Automated Bitcoin trading via machine learning algorithms*. Available at: http://cs229.stanford.edu/projects2014.html.

Makrichoriti, P., & Moratis, G. (2016). BitCoin's roller coaster: systemic risk and market sentiment. Available at SSRN: 10.2139/ssrn.2808096.

Mamdani, E., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies, 7*, 1–13.

Mehdi, K., & Mehdi, B. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing, 11*, 2664–2675.

Norgaard, M., Ravnm, O., & Poulsen, N. K. (2003). *Neural networks for modelling and control dynamic systems*. London: Springer.

Polasik, M., Piotrowska, A. I., Wisniewski, T. P., Kotkowski, R., & Lightfoot, G. (2015). Price fluctuations and the use of Bitcoin: An empirical inquiry. *International Journal of Electronic Commerce, 20*, 9–49.

Pomares, H., Rojas, I., Gonzalez, J., Rojas, F., Damas, M., & Fernandez, F. (2002). A two-stage approach to self-learning direct fuzzy controllers. *International Journal of Approximate Reasoning, 29*, 267–289.

Ross, T. (1995). *Fuzzy Logic with Engineering Applications*. McGraw-Hill Inc.

Ruan, D., & Kerre, E. (Eds.). (2000). *Fuzzy if-then rules in computational intelligence*. Springer.

Rubio, A., Bermudez, J. D., & Vercher, E. (2017). Improving stock index forecasts by using a new weighted fuzzy-trend time series method. *Expert Systems with Applications, 76*, 12–20.

Shah, D., & Zhang, K. (2014). *Bayesian regression and Bitcoin*. Available at: http://hdl.handle.net/1721.1/101044.

Sugeno, M. (1985). *Industrial applications of fuzzy control*. Elsevier Science Pub Co.

Talarposhti, F. M., Sadaei, H. J., Enayatifar, R., Guimaraes, F. G., Mahmud, M., & Eslami, T. (2016). Stock market forecasting by using a hybrid model of exponential fuzzy time series. *International Journal of Approximate Reasoning, 70*, 79–98.

Taskaya, T., & Casey, M. C. C. (2005). A comparative study of autoregressive neural network hybrids. *Neural Networks, 18*, 781–789.

Tiwari, A. K., Jana, R. K., Das, D., & Roubaud, D. (2018). Informational efficiency of Bitcoin – An extension. *Economics Letters, 163*, 106–109.

Yu, L., Wang, S. Y., & Lai, K. K. (2005). A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Computers and Operations Research, 32*, 2523–2541.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–358.

Zhu, B., & Wei, Y. (2013). Carbon price forecasting with a novel hybrid ARIMA and least squares support vector machines methodology. *Omega, 41*, 517–524.

Zhu, Y., Dickinson, D., & Li, J. (2017). Analysis on the influence factors of Bitcoin's price based on VEC model. *Financial Innovation, 3*, 3. doi:10.1186/s40854-017-0054-0.

Zimmerman, H. J. (1991). *Fuzzy set theory and its applications*. Kluwer Academic Publishing.