



# From regression models to machine learning approaches for long term Bitcoin price forecast

Andrea Caliciotti<sup>1,2</sup> · Marco Corazza<sup>3</sup> · Giovanni Fasano<sup>4</sup>

Accepted: 18 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

We carry on a long term analysis for Bitcoin price, which is currently among the most renowned crypto assets available on markets other than Forex. In the last decade Bitcoin has been under spotlights among traders all world wide, both because of its nature of *pseudo-currency* and for the high volatility its price has frequently experienced. Considering that Bitcoin price has earned over five orders of magnitude since 2009, the interest of investors has been increasingly motivated by the necessity of accurately predicting its value, not to mention that a comparative analysis with other assets as silver and gold has been under investigation, too. This paper reports two approaches for a long term Bitcoin price prediction. The first one follows more standard paradigms from regression and least squares frameworks. Our main contribution in this regard fosters conclusions which are able to justify the cyclic performance of Bitcoin price, in terms of its *Stock-to-Flow*. Our second approach is definitely novel in the literature, and indicates guidelines for long term forecasts of Bitcoin price based on Machine Learning (ML) methods, with a specific reference to Support Vector Machines (SVMs). Both these approaches are inherently data-driven, and the second one does not require any of the assumptions typically needed by solvers for classic regression problems.

**Keywords** Bitcoin · Forecast · Least squares problems · Regression · Support vector machines · Bootstrap

---

✉ Marco Corazza  
corazza@unive.it

Andrea Caliciotti  
caliciotti@diag.uniroma1.it; andrea.caliciotti@enel.com

Giovanni Fasano  
fasano@unive.it

<sup>1</sup> Department of Computer, Control, and Management Engineering 'A.Ruberti', University of Rome "La Sapienza", via Ariosto 25, 00185 Rome, Italy

<sup>2</sup> Enel Green Power S.p.A., Viale Regina Margherita 125, 00198 Rome, Italy

<sup>3</sup> Department of Economics, Ca' Foscari University of Venice, Cannaregio 873, 20121 Venice, Italy

<sup>4</sup> Department of Management, Ca' Foscari University of Venice, Cannaregio 873, 20121 Venice, Italy

## 1 Introduction

In this paper we consider a challenging price forecast problem, associated with a specific asset class, namely the crypto assets. In particular, we focus on one of the most famous crypto assets which is *Bitcoin* (Nakamoto, 2008), inasmuch as it currently corresponds also to the largest market capitalization asset among the crypto ones (see also (Various Authors, 2011; Nakamoto, 2014; Vigna & Casey, 2015)).

Bitcoin was created in 2008 (Nakamoto, 2008) by an anonymous researcher (or possibly a team of people), under the nickname of Satoshi Nakamoto. It represents a digital asset whose implementation release protocol is open–source. What strongly characterizes Bitcoin with respect to fiat currencies (as Dollar, Euro, Yen, Pound, etc.) is its decentralized nature. Indeed, no private bank or national central bank is neither responsible for managing the overall amount of circulating bitcoins nor be able to issue new bitcoins. Bitcoin negotiations need exchanges to finalize transactions. These are special intermediaries who allow the negotiation of Bitcoin vs. the main fiat currencies or vs. other crypto assets. Nevertheless, peer–to–peer movements on the Bitcoin network can be perfectly finalized without the need for intermediaries, too. Transactions among users are validated by network nodes (computers), after solving complex inverse cryptographic problems. Moreover, the transactions cannot be removed from the Bitcoin network, since they are sequentially collected into *blocks* appended to a public distributed ledger called *blockchain*. Newly minted bitcoins are created every time a block is added to the blockchain, by special nodes (computers) of the network, associated with the so called *miners*, who are rewarded for solving the above complex inverse cryptographic problems. Miners' rewarding policy of Bitcoin network changes every four years, identifying events in the history of Bitcoin known as *halvenings*, since they correspond indeed to halven the reward associated to each mined block.

In order to foresee the long term price for Bitcoin, a number of different approaches were considered in the literature (the interested reader can refer to the recent papers (Aggarwal et al., 2020; Sreekanth Reddy & Sriramy, 2020) and therein references), so that an increasing interest in the literature has grown in the last decade. Considering the wide range of stakeholders for crypto assets, ranging from practitioners to investors, researchers and members from private/public institutions, the quality of the literature on Bitcoin price prediction has sometimes been methodologically questionable. However, one of the main difficulties for Bitcoin price prediction relies on the high volatility of this asset (Baur & Dimpfl, 2021), whose price can definitely show large oscillations in a short time period. The main reason of this drawback is that Bitcoin is a relatively recent asset. Thus, considering its market capitalization (which is currently about one tenth of the overall gold capitalization), Bitcoin is often the target of speculations which include highly leveraged transactions on Bitcoin derivatives (i.e. futures and options).

Among the most recent contributions, for the assessment of the price of Bitcoin, we find the recent paper (Aggarwal et al., 2020), that introduces a ML–based approach to provide a quantitative model for Bitcoin price forecast. However, Aggarwal et al. (2020) basically relies on using *intrinsic mode functions* (IMFs), coupled with SVMs, which attempt to capture the natural characteristics of the time series associated with Bitcoin prices. As another contribution based on ML we find the analysis in Sreekanth Reddy and Sriramy (2020), which relies on the optimization method LASSO for ML.

Unlike the cited references, our main approach here is twofold. On one hand we investigate linear Least Squares models, to study the role played by the Stock–to–Flow ratio within regression problems related to Bitcoin price forecast. We recall that the Stock–to–Flow ratio

is defined as the ratio between the current overall stock of Bitcoin on the market, and the quantity of bitcoins minted in a given time period. On the other hand, we show that we can use a ML-based technique (see Sect. 5), combining Mathematical Programming and SVMs, in order to provide a long term measurement system for estimating Bitcoin price. We will show that, under suitable assumptions, the two proposals provide similar results. We remark that our second proposal does not rely on those theoretical assumptions (e.g. the normal distribution of data) which are typically associated with regression formulations. More specifically, in our second proposal we combine a preliminary multiobjective programming approach with an SVM: this represents to our knowledge a completely novel framework in the literature.

Given the foregoing, it is possible to show that to some extent the price of Bitcoin shows a dependency on its *Stock-to-Flow* ratio (SF). For instance, Fig. 1 represents 1295 pairs of Bitcoin price vs. SF, corresponding to the period between January, 2011 and July, 2022,<sup>1</sup> Data has been transformed to allow easy processing, with respect to a logarithmic scale. The (red) bullets represent Bitcoin prices corresponding to SF values, and it is not difficult to realize the high volatility of Bitcoin price. Moreover, the time window length for computing the SF is given by 463 days, as indicated by (Buy Bitcoin Worldwide, 2019).<sup>2</sup> The rationale behind the determination of this value follows guidelines suggested by other scarce assets like, for instance, gold and silver. Briefly, given the average length of the production cycle of bitcoins between two consecutive halvings (about 4 years, that is about 1,460 days), practitioners and professionals have identified within this cycle the following three consecutive market phases: *bull run*, *correction*, and *reversion to the mean*. The lengths of these three phases are estimated approximately equal among them, that is 4/3 years each, hence the (approximate) value of 463 days for the time window length used to compute the Bitcoin SF.<sup>3</sup> However, this choice of the length of the time window may be itself questionable, and represents a key parameter, as detailed in Sect. 4. All this said, the approximate solution of a linear regression problem, to forecast Bitcoin price vs. its SF, through the solution of a linear least squares problem, strongly depends on a number of issues and definitely requires specific cares. For the sake of completeness we recall that the parallel continuous lines in Fig. 1 have equations  $y = mx + q_1$  and  $y = mx + q_2$ , respectively, where  $m$ ,  $q_1$  and  $q_2$  solve the linear programming problem (each pair  $(\bar{x}_i, \bar{y}_i)$  represents the Bitcoin price, i.e.  $\bar{y}_i$ , corresponding to its SF, i.e.  $\bar{x}_i$ )

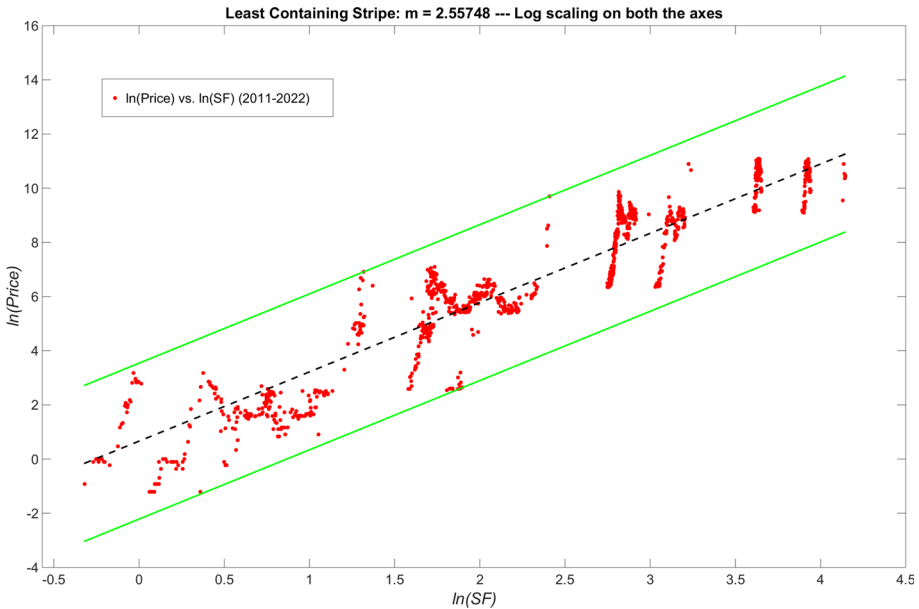
$$\begin{aligned} \min_{m, q_2, q_1} \quad & q_2 - q_1 \\ \bar{y}_i \geq \quad & m\bar{x}_i + q_1, \quad i = 1, \dots, 1295, \\ \bar{y}_i \leq \quad & m\bar{x}_i + q_2, \quad i = 1, \dots, 1295, \end{aligned}$$

that is the area between the two lines identifies the narrowest stripe containing all Bitcoin transactions.

<sup>1</sup> Note that reliable data in the early years of Bitcoin history may be hardly retrieved, because in 2009–2010 there were not yet observers in charge for accurate data collection. Hence, we decided to completely revise and update our database including more recent data, but also discarding the pairs corresponding to the years 2009–2010. In the attempt to collect more reliable data we downloaded and compared it from the websites: <https://www.blockchain.com/charts/total-bitcoins> <https://www.cryptocurrencychart.com/>, <https://datahub.io/cryptocurrency/bitcoin>, <https://www.investing.com/crypto/bitcoin/historical-data>, <https://finance.yahoo.com/cryptocurrencies>.

<sup>2</sup> In particular, see the webpage [www.buybitcoinworldwide.com/stats/stock-to-flow/](http://www.buybitcoinworldwide.com/stats/stock-to-flow/).

<sup>3</sup> Note that 463 days is about 95% of 4/3 years..



**Fig. 1** The price of Bitcoin vs. its SF. Continuous green lines (*support lines*) delimit the narrowest stripe containing all Bitcoin transactions. The dashed black line is the mid-line between the support lines and is expected to give a trend for Bitcoin price, with respect to its SF. The natural logarithm is used for scaling both the axes, and the reference time window for computing SF is 463 days [see also Sect. 4 and (Buy Bitcoin Worldwide, 2019)]

Note that the linear model suggested by Fig. 1, that is

$$\ln(\text{Price}_t) = \hat{m} \ln(\text{SF}_t) + \hat{q},$$

with  $\hat{m}$  and  $\hat{q}$  appropriate estimates of the slope and the intercept respectively, is very popular among practitioners and professionals (see for instance (Buy Bitcoin Worldwide, 2019; PlanB, 2019) and the really huge amount of more or less authoritative contributions in the so called *Socials*). However, these investigations are generally not well founded, with possible negative effects when used for professional trading purposes. One of the goals of this paper consists exactly in providing the essentials of such foundations (see Sects. 3 and 4).

The remainder of this paper is organized as follows. In Sect. 2 we remind the reader some details about the relationship between regression problems and least squares optimization approaches. Section 3 describes linear least squares schemes, where scaling on variables is suitably investigated in view of the analysis for Bitcoin price. In Sect. 4 we give indications about the computation of the SF ratio for Bitcoin, while Sect. 5 reports theoretical contributions and numerical results on long term Bitcoin price analyses which are based on Support Vector Machines (SVMs) and SVMs coupled with the bootstrap method. Finally, Sect. 6 reports some conclusions and guidelines for future work.

As regards the symbols used in this paper, we indicate with  $\mathbb{E}[v]$  the expected value of the real random variable/vector  $v$ . With  $A > 0$  [ $A \geq 0$ ] we indicate that the matrix  $A$  is positive definite [positive semidefinite]. Finally,  $|B|$  denotes the cardinality of the set  $B$ .

## 2 Regression problems and least squares optimization

To better explain the reliability of solving linear least squares problems in the context of linear regression formulations, assume we are given the  $1 + p$  random variables  $Y$  and  $\{X_i\}$ , being

$$Y = \sum_{i=1}^p \beta_i X_i + u_i, \quad \beta_i \in \mathbb{R}, \quad i = 1, \dots, p, \quad (2.1)$$

where  $Y$  is the dependent variable and  $\{X_i\}$  are the independent unknowns. Moreover,  $u_i$  represents a statistical error, for any  $i$ , and satisfies  $\mathbb{E}(u_i | X_1, \dots, X_p) = 0$ . Then, if  $Y, X_1, \dots, X_p$  are *independent and identically distributed* (i.i.d.) and a few mild assumptions are fulfilled, the solution of the Linear Regression problem

$$\min_{\hat{a}, \hat{b}} \mathbb{E} \left[ Y - \left( \hat{b} + \sum_{i=1}^p \hat{a}_i X_i \right) \right], \quad \hat{a} \in \mathbb{R}^p, \hat{b} \in \mathbb{R} \quad (2.2)$$

can be equivalently obtained by solving the Linear Least Squares problem

$$\min_{a, b} \sum_{j=1}^N \left[ Y^{(j)} - \left( b + \sum_{i=1}^p a_i X_i^{(j)} \right) \right]^2, \quad a \in \mathbb{R}^p, b \in \mathbb{R} \quad (2.3)$$

where  $N$  represents the number of available samples for the random variables  $(Y, X_1, \dots, X_p)$ . We strongly remark that the solution of the minimization problem (2.3) is definitely appealing, since it is an unconstrained convex quadratic model. However, we also highlight that the solutions of (2.2) and (2.3) might strongly differ in case the theoretical assumptions on the quantities  $Y, X_1, \dots, X_p, u_1, \dots, u_p$  were not fulfilled. A typical example where we experience the last drawback is the case in which the samples do not follow a normal distribution. Conversely, in case the random variables  $u_1, \dots, u_p$  admit the joint normal distribution  $N(0, \sigma^2 I)$ , with zero expected value and the same variance for all the variables, then the solutions of (2.2) and (2.3) coincide.

Typically in applied sciences (2.3) is often solved assuming the fulfillment of indispensable theoretical assumptions, which unfortunately are often not satisfied, as specified above. Thus, a test on the reliability of the solutions of (2.3) is usually sought (e.g. the  $R^2$  and the  $p$ -value indicators).

In this paper we are interested about estimating the price (i.e.  $Y$ ) of Bitcoin vs. its SF (i.e.  $X$ ), being  $Y = aX + u$ , with  $a \in \mathbb{R}$ ; however, the error  $u$  is not normally distributed, so that the solution of the linear least squares problem (2.3) might possibly represent a poor estimator [see also the practical analysis on Graybill and Iyer (1994)].

## 3 Our setting for data scaling in Least Squares problems

On the guidelines of Sect. 2, we detail here the linear least squares setting we make reference to. Note that in this section we propose some novel theoretical results, in order to address and give foundation to claims and questions raised by practitioners, within the literature on Bitcoin [see for instance Buy Bitcoin Worldwide (2019); PlanB (2019)]. Generally, these results are simple from a mathematical standpoint (mainly, they refer to the classical Box-Cox transformation approach Box and Cox (1964)) but are meaningful from the point of

view of the Bitcoin price forecast. For these reasons, below we consider a couple of different linear regression problem formulations, as general as possible, in order to take into account a number of possible parameters that can affect Bitcoin price forecast.

Let us consider the  $N$  training pairs  $(\bar{x}_i, \bar{y}_i) \in \mathbb{R}^2$ ,  $i = 1, \dots, N$ , and let us consider the following least squares problem, where we assume that the training pairs are possibly subject to a log–transformation, being  $\alpha$  and  $\beta$  suitable positive parameters not equal to zero or one

$$\min_{m, q \in \mathbb{R}} \sum_{i=1}^N [\log_{\alpha}(\omega \bar{y}_i) - m \log_{\beta}(\bar{x}_i) - q]^2. \quad (3.4)$$

In particular, in the pair  $(\bar{x}_i, \bar{y}_i)$  the quantity  $\bar{x}_i$  is associated with Bitcoin SF value, while  $\bar{y}_i$  represents the corresponding price of Bitcoin. The motivation for introducing in (3.4) the parameter  $\omega$  will be clear later on, and further specific motivations to recur to the log–scaled expression (3.4) are given in Buy Bitcoin Worldwide (2019). They basically reduce to the fact that, assimilating the pair  $(\bar{x}_i, \bar{y}_i)$  to a determination of a two–dimensional random variable  $(x, y)$ , then  $(x, y)$  has not necessarily a normal distribution. In this regard, a log–transformation, with respect to at least one of the random variables, may possibly yield the pairs  $\{(\bar{x}_i, \bar{y}_i)\}$  to better resemble a normal distribution.

**Proposition 1** *Given the problem (3.4) with  $\alpha, \beta \notin \{0, 1\}$  and positive, let*

$$\begin{aligned} A &= \sum_{i=1}^N [\log_{\beta}(\bar{x}_i)]^2, & B &= \sum_{i=1}^N [\log_{\alpha}(\bar{y}_i)]^2, \\ C &= 2 \sum_{i=1}^N \log_{\beta}(\bar{x}_i), & D &= 2 \sum_{i=1}^N \log_{\alpha}(\bar{y}_i), \\ E &= 2 \sum_{i=1}^N \log_{\beta}(\bar{x}_i) \log_{\alpha}(\bar{y}_i). \end{aligned} \quad (3.5)$$

*Assume without loss of generality that  $B \neq 0$  and the sequence  $\{\log_{\beta}(\bar{x}_i)\}$  contains at least two non–coincident entries. Then the optimal solution  $m^*, q^*$  to (3.4) satisfies the following properties:*

1.  $m^*$  is independent of  $\omega$ , being

$$m^* = \frac{2NE - CD}{4NA - C^2};$$

2.  $q^*$  is not independent of  $\omega$ , being

$$q^* = \log_{\alpha}(\omega) - \frac{CE - 2AD}{4NA - C^2}.$$

**Proof** We rewrite (3.4) as

$$\min_{m, q \in \mathbb{R}} \sum_{i=1}^N [\log_{\alpha}(\omega) + \log_{\alpha}(\bar{y}_i) - m \log_{\beta}(\bar{x}_i) - q]^2,$$

so that after computing the square in the sum and collecting the terms we equivalently obtain the problem

$$\min_{m, q \in \mathbb{R}} N [\log_{\alpha}(\omega)]^2 + Am^2 + Nq^2 + B - C \log_{\alpha}(\omega)m - 2N \log_{\alpha}(\omega)q + D \log_{\alpha}(\omega) + Cmq - Em - Dq$$

and after a few arrangements

$$\min_{m,q \in \mathbb{R}} N \left\{ [\log_\alpha(\omega)]^2 + q^2 - 2 \log_\alpha(\omega)q \right\} + Am^2 - Cm [\log_\alpha(\omega) - q] + D [\log_\alpha(\omega) - q] - Em + B$$

or equivalently

$$\min_{m,q \in \mathbb{R}} N [\log_\alpha(\omega) - q]^2 + Am^2 + (D - Cm) [\log_\alpha(\omega) - q] - Em + B.$$

Thus, after setting  $s = \log_\alpha(\omega) - q$  we equivalently have to solve the problem

$$\min_{m,s \in \mathbb{R}} \psi(s, m) \equiv Ns^2 + Am^2 + (D - Cm)s - Em + B.$$

Now observe that the Hessian matrix of the last quadratic function is constant and is given by

$$\nabla^2 \psi(s, m) = \begin{pmatrix} 2N & -C \\ -C & 2A \end{pmatrix},$$

so that it is positive definite. Indeed  $N, A \geq 0$  and  $\det [\nabla^2 \psi(s, m)] = 4NA - C^2$ , with  $|C| \leq 2\|v\|_1$  and

$$v = \begin{bmatrix} \log_\beta(\bar{x}_1) \\ \vdots \\ \log_\beta(\bar{x}_N) \end{bmatrix}.$$

Hence,  $C^2 \leq 4\|v\|_1^2$  so that

$$\det [\nabla^2 \psi(s, m)] = 4NA - C^2 \geq 4N\|v\|_2^2 - 4\|v\|_1^2 = 4(\sqrt{N}\|v\|_2 + \|v\|_1)(\sqrt{N}\|v\|_2 - \|v\|_1).$$

Since  $\|v\|_1 \leq \sqrt{N}\|v\|_2$  (see Fact 9.8.12 in Bernstein,2009), then  $\det [\nabla^2 \psi(s, m)] \geq 0$ . Moreover, since at least two entries of  $v$  are non coincident, then  $\psi(s, m)$  is a strictly convex function in  $\mathbb{R}^2$ , inasmuch as  $4NA - C^2 > 0$ . Now, first order stationarity conditions applied to  $\psi(s, m)$  yield

$$\begin{cases} \frac{\partial \psi(s^*, m^*)}{\partial s} = 2Ns^* + (D - Cm^*) = 0 \\ \frac{\partial \psi(s^*, m^*)}{\partial m} = 2Am^* - Cs^* - E = 0, \end{cases}$$

so that

$$\begin{cases} m^* = \frac{2NE - CD}{4NA - C^2} \\ s^* = \frac{CE - 2AD}{4NA - C^2}, \end{cases} \tag{3.6}$$

i.e. the statement of the proposition definitely holds.  $\square$

Similarly to the previous proposition we also have the following result, where the scaling parameter  $\omega$  is applied to the sequence  $\{\bar{x}_i\}$ , i.e. now we refer to the case where problem (3.4) becomes

$$\min_{m, q \in \mathbb{R}} \sum_{i=1}^N [\log_{\alpha}(\bar{y}_i) - m \log_{\beta}(\omega \bar{x}_i) - q]^2. \quad (3.7)$$

**Proposition 2** *Given the problem (3.7) with  $\alpha, \beta \notin \{0, 1\}$  and positive, let us consider the positions (3.5). Assume without loss of generality that  $B \neq 0$ ,  $A + C \log_{\beta}(\omega) > 0$  and the sequence  $\{\log_{\beta}(\bar{x}_i)\}$  contains at least two non-coincident entries. Then the optimal solution  $m^*, q^*$  to (3.7) satisfies the following properties:*

1.  $m^*$  is independent of  $\omega$ , being

$$m^* = \frac{2NE - CD}{4NA - C^2};$$

2.  $q^*$  is not independent of  $\omega$ , being

$$q^* = -\frac{2NE - CD}{4NA - C^2} \log_{\beta}(\omega) - \frac{CE - 2AD}{4NA - C^2}.$$

**Proof** Following the guidelines of Proposition 1 we can equivalently rewrite (3.7) as

$$\min_{m, q \in \mathbb{R}} \sum_{i=1}^N [\log_{\alpha}(\bar{y}_i) - m [\log_{\beta}(\omega) + \log_{\beta}(\bar{x}_i)] - q]^2,$$

i.e.

$$\begin{aligned} \min_{m, q \in \mathbb{R}} \psi(m, q) \equiv & B + Nm^2 [\log_{\beta}(\omega)]^2 + Am^2 + Cm^2 \log_{\beta}(\omega) + Nq^2 - Dm \log_{\beta}(\omega) \\ & - Em - Dq + 2Nm q \log_{\beta}(\omega) + Cmq. \end{aligned}$$

Observe that by the assumptions the Hessian matrix

$$\nabla^2 \psi(m, q) = \begin{pmatrix} 2N [\log_{\beta}(\omega)]^2 + 2A + 2C \log_{\beta}(\omega) & 2N \log_{\beta}(\omega) \\ 2N \log_{\beta}(\omega) & 2N \end{pmatrix}$$

is positive definite (indeed the assumption  $A + C \log_{\beta}(\omega) > 0$  implies that both its determinant and its trace are positive), so that  $\psi(m, q)$  is strictly convex on  $\mathbb{R}^2$ .

Now, first order stationarity conditions applied to  $\psi(m, q)$  yield

$$\begin{cases} \frac{\partial \psi(m^*, q^*)}{\partial m} = 2Nm^* [\log_{\beta}(\omega)]^2 + 2Am^* + (2Cm^* - D) \log_{\beta}(\omega) - E \\ \quad + 2Nq^* \log_{\beta}(\omega) + Cq^* = 0 \\ \frac{\partial \psi(m^*, q^*)}{\partial q} = 2Nq^* - D + 2Nm^* \log_{\beta}(\omega) + Cm^* = 0, \end{cases}$$

so that from the second equation

$$q^* = \frac{D - m^* [2N \log_{\beta}(\omega) + C]}{2N}$$



and replacing into the first equation we have after some computations

$$m^* = \frac{2NE - CD}{4NA - C^2}.$$

Note also that, similarly to Proposition 1, the assumptions ensure that the denominator  $4NA - C^2$  of  $q^*$  is strictly positive. Finally, we also obtain after some computations

$$\begin{aligned} q^* &= \frac{D - m^* [2N \log_\beta(\omega) + C]}{2N} = -m^* \log_\beta(\omega) + \frac{D - m^*C}{2N} \\ &= -\frac{2NE - CD}{4NA - C^2} \log_\beta(\omega) - \frac{CE - 2AD}{4NA - C^2}, \end{aligned}$$

which completes the proof. □

**Observation 3.1** *We highlight that whatever the value of  $\omega$  in the linear regression problems (3.4) and (3.7), Propositions 1 and 2 give the same optimal  $m^*$ s, and that in case  $\omega = 1$  in (3.4) and (3.7), then evidently the optimal  $q^*$ s from Propositions 1 and 2 (as expected) coincide. Moreover, the assumption  $A + C \log_\beta(\omega) > 0$  in Proposition 2 is not particularly restrictive for Bitcoin, as detailed in Sect. 4.*

**Observation 3.2** *We stress that the outcomes according to which both (3.4) and (3.7) have optimal  $m^*$ s independent of  $\omega$  imply that the value of  $\omega$  does not affect the forecast of the Bitcoin price variation.*

As for the roles played by the bases of the logarithms  $\alpha$  and  $\beta$ , with respect to the natural one, we provide the following results.

**Corollary 1** *Let us consider the problem (3.4) with  $\alpha = \alpha', \beta = \beta' \notin \{0, 1\}$  and positive, and let us consider the same problem with  $\alpha = \beta = e$ . Then the problem using the natural base has the optimal intercept  $q_e^*$  such that:*

$$q_e^* = \frac{1}{\log_{\alpha'}(e)} q^*.$$

**Proof** The proof directly holds performing a straightforward computation. □

**Corollary 2** *Let us consider the problem (3.7) with  $\alpha = \alpha', \beta = \beta' \notin \{0, 1\}$  and positive, and let us consider the same problem with  $\alpha = \beta = e$ . Then the problem using the natural base has the optimal intercept  $q_e^*$  such that:*

$$q_e^* = \frac{1}{\log_{\alpha'}(e)} q^*.$$

**Proof** Again the proof directly holds performing a straightforward computation. □

**Corollary 3** *Let us consider the problems (3.4) and (3.7) with  $\alpha = \alpha', \beta = \beta' \notin \{0, 1\}$  and positive, and let us consider the same problems with  $\alpha = \beta = e$ . Then both the problems using the natural base have the same optimal slope  $m_e^*$  such that:*

$$m_e^* = \frac{\log_{\beta'}(e)}{\log_{\alpha'}(e)} m^*.$$

**Proof** The proof follows the guidelines for Corollaries 1 and 2. □

**Observation 3.3** *We remark that in case  $\alpha' = \beta'$  with  $\alpha', \beta' \notin \{0, 1\}$  and positive, then  $m_e^* = m^*$ . This sheds some lights on possible doubts raised within the literature for practitioners, about the possible dependency of Bitcoin price forecast on the bases  $\alpha, \beta$  in (3.4) and (3.7) (see PlanB (2019)).*

## 4 The assessment of the SF for Bitcoin

As reported in Sect. 1, several authors in the literature have observed a relationship between the price of Bitcoin and its SF value (see e.g. Buy Bitcoin Worldwide (2019), among the first). Here we want to better investigate the last conclusion, in view of Propositions 1 and 2. In particular, we want to show that the SF value for Bitcoin strongly relies on the time window we use to compute it: this motivates the introduction of the parameter  $\omega$  in the formulation (3.7), in case  $\bar{y}_i$  represents the  $i$ -th price of Bitcoin and  $\bar{x}_i$  is associated with the  $i$ -th value of the SF.

As a preliminary example, recalling that the SF is the ratio between the current stock of an asset and its flow within a given time window, assume the flow is computed in a time interval of one year (i.e. 365 days as in Fig. 1). Then, SF essentially represents the number of years, at the current annual production rate, that are necessary to obtain its current stock. Hence, the higher the SF the scarcer the asset. Now, let us consider in our example the pairs *price* vs. SF of Bitcoin, from January 1st, 2011, to July 15th, 2022<sup>4</sup>. Then, the formula to be adopted for the computation of the SF for Bitcoin should be (as an example the stock in the formula refers to the end of September, 2021)

$$SF_{Bitcoin} \approx \frac{18,700,000}{\frac{24 \cdot 60}{10} \cdot 463 \cdot 6.25} \approx 44.9. \quad (4.8)$$

The quantities in the last computation duly keep into consideration the following facts:

- every day about  $24 \cdot 60/10$  newly mined blocks are added to Bitcoin blockchain (i.e. around one block every 10 min);
- for each newly added block to the blockchain of Bitcoin, exactly 6.25 bitcoins are minted and rewarded to miners, so that they can be possibly negotiated (i.e. they are potentially available) on the market. Moreover, the value 6.25 will be successively halved in  $2020 + 4k$ , for any  $k = 1, 2, \dots$ , up to about 2140;
- the time window for computing the stock is different with respect to 365 days (being indeed *463 days* as suggested in Buy Bitcoin Worldwide (2019)), so that the corresponding value is used in the denominator of (4.8), in place of 365.

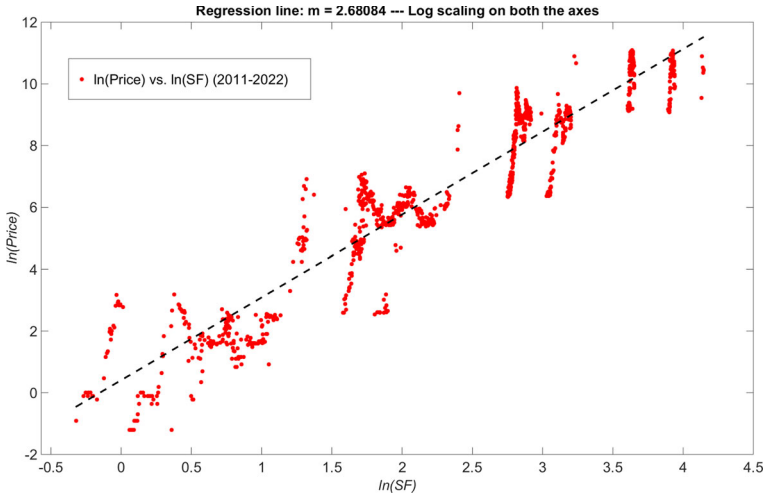
We comparatively observe that the current SF of the gold is a bit larger than 60, so that we conclude that the scarcity of Bitcoin to some extent compares with that of the gold (hence the widely used nickname of *digital gold* for Bitcoin). Figure 2 reports a plot of Bitcoin price vs. its SF: the dashed line is a regression line obtained from solving either the problems (3.4) or (3.7), after setting  $\omega = 1$ . The slope of the resulting regression line is  $m^* \approx 2.68084$  and  $q^*$  amounts to  $\approx 0.40596$ , while red points represent the pairs  $(\ln(\text{price}), \ln(SF))$ . Thus, relation

$$\ln(\text{price}) = 2.680 \ln(SF) + 0.405 \quad (4.9)$$

immediately yield the Bitcoin price forecast. For instance, considering the value of the SF in (4.8) (i.e. at the end of September, 2021), we had

$$\text{price} = SF^{2.680} \cdot e^{0.405} = 44.9^{2.680} \cdot e^{0.405} \approx 40,171\$.$$

<sup>4</sup> For further information on data for Bitcoin prices and the number of minted bitcoins, the reader may refer to the footnote at page 2.



**Fig. 2** The price of Bitcoin vs. its SF (the flow is computed with respect to a time window of 463 days), where data refers to the period between January 1st, 2011 and July 15th, 2022. The black dashed line represents a regression line from solving either the problems (3.4) or (3.7), after setting  $\omega = 1$ . The natural logarithm (i.e.  $\alpha = \beta = e$ ) is used for scaling both the axes. The slope of the regression line is  $m^* \approx 2.68084$

For the sake of completeness observe that  $R^2$  and  $p$ -value, associated with the computation of the linear regression model (4.9), are given by

$$R^2 = 0.639, \quad p \ll 0.05,$$

which shows that the linear regression model is only partially reliable. Furthermore, we remark that there is no discrepancy between the outcomes of Figs. 1 and 2, since in both the figures the flow, in the SF for Bitcoin, is computed with respect to the same time window (i.e. 463 days). This conclusion highlights the importance of introducing the parameter  $\omega$  in (3.7), inasmuch as the following observations hold:

- when in (3.7)  $\bar{x}_i$  represents a value of Bitcoin SF and we set  $\omega = 1$ , then it means we are adopting *exactly* the formula (4.8) for the SF;
- when in (3.7)  $\bar{x}_i$  represents a value of Bitcoin SF and we set  $\omega = 365/463$ , then it means we are adopting the formula (4.8), with the number 365 in place of 463.

Hence, the parameter  $\omega$  duly takes into account the time window adopted for computing the SF of Bitcoin, and according with Proposition 2 it does not influence the slope  $m^*$  of the corresponding regression line. This is to our knowledge a remarkable novel result in the literature.

**Observation 4.1** *We highlight that the choice of the value of  $\omega$  affects the values of the optimal  $q^*$ s given by Propositions 1 and 2, and consequently influences the Bitcoin price forecast. Therefore, in order to improve the quality of the forecast, a more general optimization process might be considered, also taking into consideration  $\omega$  itself as unknown, in addition to  $q$  and  $m$ . It will be a point of our future research on this topic. Nevertheless, in general, for trading purposes what really matters is possibly not the forecast of the asset price, but rather the forecast of the asset price variation. In this regard, as stressed in Observation 3.2, the value of  $\omega$  does not affect the Bitcoin price variation forecast. So that, from an operational point of view, the choice of such parameter might be not so crucial.*

Lastly, given that the linear model is inherently data-driven, we point out that the assumptions needed by the solver to address the corresponding regression problem has to be satisfied. In this regard, we recall that one of the main assumptions underlying the considered regression problem is that the probability distribution of  $Y$  conditional on  $(X_1, \dots, X_i, \dots, X_p)$  is normal (see (2.1)).

In order to verify this assumption, we carried out the test of Jarque and Bera (see 1987) as follows

$$\begin{cases} H_0: & \text{the probability distribution of } u \text{ is normal} \\ H_1: & \text{the probability distribution of } u \text{ is not normal} \end{cases}$$

where  $u \doteq (u_1, \dots, u_N)$  is the vector of the residuals coming from the regression problem

$$\min_{m,q} \sum_{j=1}^N \left[ \ln(\text{price})^{(j)} - m \ln(\text{SF})^{(j)} - q \right]^2, \quad m, q \in \mathbb{R}.$$

In particular, we performed the test for the overall period considered in the example presented in this section, that is from January 1st, 2011 to July 15th, 2022, and for each of the thirteen time-windows considered in Sect. 5.2. For all such time periods, the null hypothesis has been accepted at the 5% significance level, with  $p$ -values ranging in the interval  $[0.1170, 0.1666]$ . Hence, given this matter of fact, we decided to admit the use of the linear regression approach when reporting the comparative results in Sect. 5.2.

## 5 Our second approach for long term Bitcoin price prediction

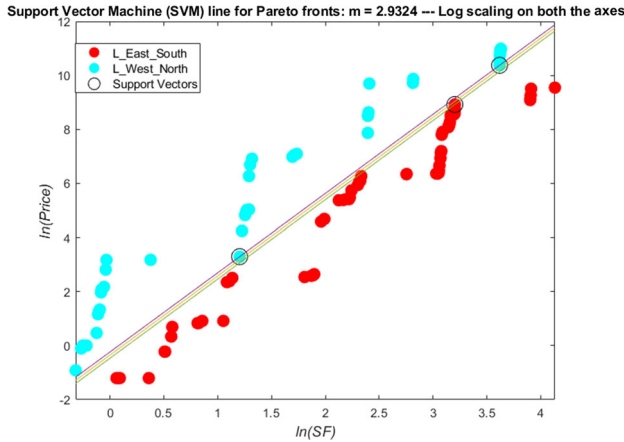
In this section we describe our second approach for the long term Bitcoin price forecast. In particular, first, in Sect. 5.1, we show that combining a Multiobjective Technique (MT), from Nonlinear Programming, with a SVM, from ML (see also Cristianini and Shawe-Taylor, 2000; Vapnik, 1995; 1998; Hastie et al., 2008; Deng et al., 2013; Marsland, 2015), we can recover results similar to those from the first approach, without requiring any assumption typically needed by regression frameworks. Then, in Sect. 5.2 we couple the above SVM-based approach with a bootstrap method, in order to possibly improve the quality of its performance. Finally, we apply both such developed methodologies to the long term forecast of Bitcoin price.

### 5.1 The algorithm MT-SVM

In order to combine MTs with SVMs, let  $(\bar{x}_i, \bar{y}_i)$ ,  $i = 1, \dots, N$ , be the pairs of SF and Bitcoin price, within a given time interval. Then, we preliminarily identify the two subsets  $L_{\max}$  and  $L_{\min}$  of  $\{(\bar{x}_i, \bar{y}_i)\}$ , each associated with a different weak Pareto front. In particular, consider Fig. 3. We compute the sets  $L_{\max}$  and  $L_{\min}$ , being now respectively  $L_{\max}$  indicated by  $L_{West-North}$  and  $L_{\min}$  indicated by  $L_{East-South}$ , as

- $L_{East-South}$ : the weak Pareto front (red points) associated with both the maximization of the stock-to-flow SF and the minimization of Bitcoin price;
- $L_{West-North}$ : the weak Pareto front (cyan points) associated with both the minimization of the stock-to-flow SF and the maximization of Bitcoin price.

For a more formal definition of these last sets of points, the reader can consider that the point  $(\bar{x}_i, \bar{y}_i)$  will be classified as a point in  $L_{\max}$  if it satisfies the properties (*non-dominated*



**Fig. 3** The fronts  $L_{East-South}$  and  $L_{West-North}$  among the pairs Bitcoin price vs. its SF. Data is reported using  $\ln()$  transformation

point with respect to maximization of  $\bar{y}$  and minimization of  $\bar{x}$ )

$$(\bar{x}_i, \bar{y}_i) \in L_{\max} \text{ if } \nexists j \in \{1, \dots, N\}, \text{ with } (\bar{x}_j, \bar{y}_j) \neq (\bar{x}_i, \bar{y}_i), \\ \text{s.t. } \bar{x}_j < \bar{x}_i \text{ and } \bar{y}_j > \bar{y}_i.$$

Similarly, the point with coordinates  $(\bar{x}_i, \bar{y}_i)$  will be classified as a point in  $L_{\min}$  if it satisfies the properties (*dominated point* with respect to minimization of  $\bar{y}$  and maximization of  $\bar{x}$ )

$$(\bar{x}_i, \bar{y}_i) \in L_{\max} \text{ if } \nexists j \in \{1, \dots, N\}, \text{ with } (\bar{x}_j, \bar{y}_j) \neq (\bar{x}_i, \bar{y}_i), \\ \text{s.t. } \bar{x}_j > \bar{x}_i \text{ and } \bar{y}_j < \bar{y}_i.$$

Broadly speaking, the front  $L_{\max}$  includes (desirable) points with high price performance for Bitcoin vs. its SF. On the contrary,  $L_{\min}$  contains (undesirable) points with poor price performance for Bitcoin vs. its SF. Therefore, the sets  $L_{\max}$  and  $L_{\min}$  include points which may be associated with *extreme opposite performances* of Bitcoin price, and can be used in our SVM-based classification framework to possibly model long term Bitcoin price.

In the remaining part of this section we propose an iterative procedure which relies on the following definition (all the theoretical results in the current section refer to the set  $\{(\bar{x}_i, \bar{y}_i)\}$ , with  $\bar{x}_i \in \mathbb{R}$  and  $\bar{y}_i \in \mathbb{R}$ . Nevertheless, in Pontiggia and Fasano (2021) the authors proved that they can be immediately extended to the set  $\{(\bar{x}_i, \bar{y}_i)\}$ , with  $\bar{x}_i \in \mathbb{R}^p, p \geq 2, \bar{y}_i \in \mathbb{R}$ , too).

**Definition 1** Given the points  $(\bar{x}_i, \bar{y}_i) \in \mathbb{R}^2, i = 1, \dots, N$ , and the values  $z_i \in \{-1, +1\}, i = 1, \dots, N$ , let us define the nonempty sets  $A = \{(\bar{x}_i, \bar{y}_i) \mid z_i = +1\}$  and  $B = \{(\bar{x}_i, \bar{y}_i) \mid z_i = -1\}$ . Then, we say that  $A$  and  $B$  are linearly separable in  $\mathbb{R}^2$  if there exists a line  $H(\beta, \beta_0; x, y) = 0$ , with coefficients  $\beta, \beta_0 \in \mathbb{R}$ , such that

$$\begin{cases} H(\beta, \beta_0; \bar{x}_i, \bar{y}_i) > 0, & \forall i : z_i = +1 \\ H(\beta, \beta_0; \bar{x}_i, \bar{y}_i) < 0, & \forall i : z_i = -1. \end{cases} \tag{5.10}$$

The procedure in our proposal starts by setting  $A_0 = L_{\max}$  and  $B_0 = L_{\min}$ ; then, we generate the sequences of sets  $\{A_k\}$  and  $\{B_k\}$ , with  $k = 0, 1, 2, \dots$ , according with the next two distinct phases:

- first, we solve an SVM classification problem that computes the line

$$H\left(\beta^{(k)}, \beta_0^{(k)}; x, y\right) = 0$$

which linearly separates (see Definition 1) the sets  $A_k$  and  $B_k$ , and is maximally distant (maximum *margin*) from their points;

- second, a point  $\left(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}\right)$  in  $\{(\bar{x}_i, \bar{y}_i), i = 1, \dots, N\} \setminus \{A_k \cup B_k\}$  is identified and assigned with a label  $z_k \in \{-1, +1\}$ . In case  $z_k = +1$  we generate the novel sets  $A_{k+1} = A_k \cup \left\{\left(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}\right)\right\}$  and  $B_{k+1} = B_k$ , otherwise we set  $A_{k+1} = A_k$  and  $B_{k+1} = B_k \cup \left\{\left(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}\right)\right\}$ .

More formally, our proposal is summarized in the *Algorithm MT-SVM* of Table 1, whose steps are briefly commented as follows.

In a preliminary initialization we set  $\chi = \{(\bar{x}_i, \bar{y}_i), i = 1, \dots, N\}$  and set  $A_0$  and  $B_0$ , using  $L_{\max}$  and  $L_{\min}$  defined above. Then, we compute the *best* (i.e. the one maximizing the *margin*) separating line  $H\left(\beta^{(0)}, \beta_0^{(0)}; x, y\right) = 0$  between  $A_0$  and  $B_0$ , whose parameters are given by  $\beta^{(0)}$  and  $\beta_0^{(0)}$ , using an SVM method. Thus,  $H\left(\beta^{(0)}, \beta_0^{(0)}; x, y\right) = 0$  is the maximally distant line with respect to both the sets  $A_0$  and  $B_0$ . In particular, in *Algorithm MT-SVM* we indicate with  $SVM(A_k, B_k)$  the solution of the SVM problem which computes the best (maximally distant) separating line between  $A_k$  and  $B_k$ . Note that the SVM problem reduces to a convex linearly constrained quadratic minimization problem which always admits solution (see Cristianini and Shawe-Taylor, 2000).

Therefore, at any step  $k$  we first pick the point  $\left(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}\right)$  in  $\chi \setminus \{A_k \cup B_k\}$  with the largest distance from  $H\left(\beta^{(k)}, \beta_0^{(k)}; x, y\right) = 0$ . Moreover, depending on the half space where  $\left(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}\right)$  is located, with respect to the line  $H\left(\beta^{(k)}, \beta_0^{(k)}; x, y\right) = 0$ , we update the novel sets  $A_{k+1}, B_{k+1}$  starting from the pair  $A_k, B_k$ . In the end, we increase the step and iterate the procedure. The next results can be proved, which establishes theoretical properties for *Algorithm MT-SVM* (see also Pontiggia and Fasano, 2021).

**Lemma 1** Consider the set  $\chi \subset \mathbb{R}^2$ , with  $|\chi| < +\infty$ . Let  $L_{\max}, L_{\min} \subseteq \chi$ . Then, the Algorithm MT-SVM in Table 1 provides the pair of sets  $A_m, B_m$  after  $m$  steps, with

$$m = |\chi \setminus \{L_{\max} \cup L_{\min}\}|,$$

such that

$$\begin{cases} A_m \cup B_m = \chi \\ A_m \cap B_m = \emptyset. \end{cases}$$

**Proof** By Table 1, recalling that  $|\chi|$  is finite, the index  $k$  ranges from 0 to  $|\chi \setminus \{A_0 \cup B_0\}| \leq |\chi| < +\infty$ . Moreover, since by construction  $|A_k \cup B_k| = |A_{k-1} \cup B_{k-1}| + 1$ , then  $m$  is exactly given by the number of points in  $\chi$  which are neither present in  $L_{\max}$  nor in  $L_{\min}$ .  $\square$

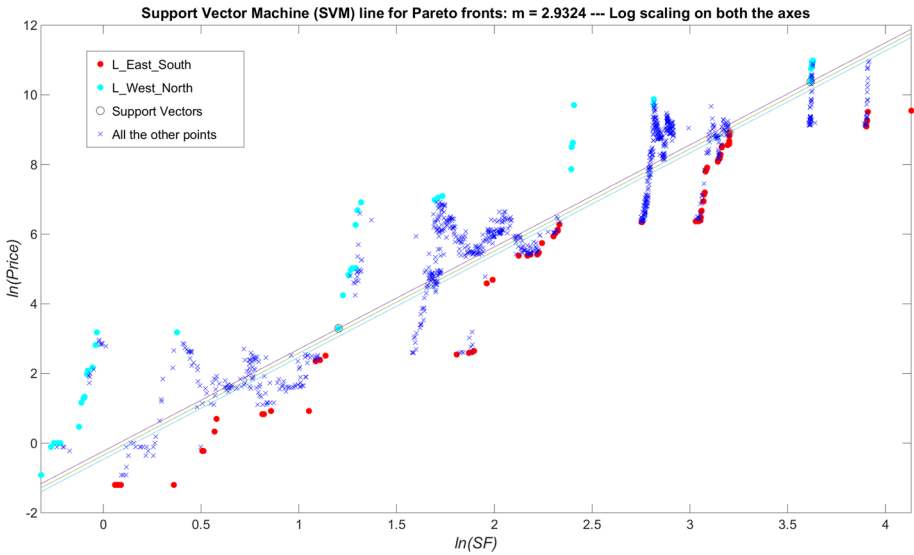
**Proposition 3** Let be given the nonempty sets  $L_{\max}, L_{\min} \subseteq \chi$ , and consider the Algorithm MT-SVM in Table 1. If  $L_{\max}$  and  $L_{\min}$  are linearly separable, as by Definition 1, then the sets  $A_k$  and  $B_k$  are linearly separable, for any  $k \geq 0$ .

**Proof** The proof can be found in Pontiggia and Fasano (2021).  $\square$

**Table 1** Description of our SVM-based procedure applied to the points  $\{(\bar{x}_i, \bar{y}_i), i = 1, \dots, N\}$  in the set  $\chi$ **Algorithm MT-SVM****Data:**  $\chi \equiv \{(\bar{x}_i, \bar{y}_i), i = 1, \dots, N\}$ **Initialization:** Set  $k \leftarrow 0$ ,  $choice \in \{TRUE, FALSE\}$ ,  $A_0 \leftarrow L_{\max}$ ,  $B_0 \leftarrow L_{\min}$ **Step k:** **While**  $(|\chi \setminus \{A_k \cup B_k\}| \neq 0)$     Compute the parameters  $[\beta^{(k)}, \beta_0^{(k)}] = SVM(A_k, B_k)$     Set  $d_{\max}^{(k)} \leftarrow 0$     **For**  $i = 1 : |\chi \setminus \{A_k \cup B_k\}|$         Extract  $(\bar{x}_i, \bar{y}_i)$  from  $\chi \setminus \{A_k \cup B_k\}$         Compute the distance  $d_i = d[(\bar{x}_i, \bar{y}_i), H(\beta^{(k)}, \beta_0^{(k)}; x, y)]$         **If**  $(d_i \geq d_{\max}^{(k)})$  **Then**            Set  $d_{\max}^{(k)} \leftarrow d_i$             Set  $(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}) \leftarrow (\bar{x}_i, \bar{y}_i)$         **End If**    **End For**    **If**  $(d_{\max}^{(k)} > 0)$  **Then**        **If**  $(H(\beta^{(k)}, \beta_0^{(k)}; \bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}) > 0)$  **Then**            Set  $A_{k+1} \leftarrow A_k \cup \{(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)})\}$             Set  $B_{k+1} \leftarrow B_k$         **Else** (i.e.  $H(\beta^{(k)}, \beta_0^{(k)}; \bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)}) < 0$ )            Set  $A_{k+1} \leftarrow A_k$             Set  $B_{k+1} \leftarrow B_k \cup \{(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)})\}$         **End If**    **Else** (i.e.  $d_{\max}^{(k)} = 0$ )        **If** ( $choice = FALSE$ ) **Then**            Set  $A_{k+1} \leftarrow A_k \cup \{(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)})\}$             Set  $B_{k+1} \leftarrow B_k$         **Else**            Set  $A_{k+1} \leftarrow A_k$             Set  $B_{k+1} \leftarrow B_k \cup \{(\bar{x}_{\max}^{(k)}, \bar{y}_{\max}^{(k)})\}$         **End If**    **End If**     $k \leftarrow k + 1$ **End While**

We can also give a couple of additional results, to better clarify the properties of the solution provided by *Algorithm MT-SVM*, either in case  $L_{\max}$  and  $L_{\min}$  are / are not linearly separable.

**Lemma 2** *Let be given the nonempty sets  $L_{\max}, L_{\min} \subseteq \chi$ , and consider the Algorithm MT-SVM in Table 1. If  $L_{\max}$  and  $L_{\min}$  are linearly separable, then*



**Fig. 4** The fronts  $L_{East-South}$  and  $L_{West-North}$ , for the problem of forecasting Bitcoin price vs. its SF ratio. Applying *Algorithm MT-SVM* and stopping right after iteration  $k = 0$  we obtain the circled points (support vectors) and the side lines (support lines). (Blue) crosses identify all the pairs Bitcoin price vs. its SF. Data is reported using  $\ln(\cdot)$  transformation

- for any  $k \geq 1$  the margin  $W^{(k)}$  of the SVM problem  $SVM(A_k, B_k)$  satisfies

$$W^{(k)} = \min \left\{ W^{(k-1)}, 2d_{\max}^{(k)} \right\}; \tag{5.11}$$

- the sequence  $\{W^{(k)}\}$  is monotonically nonincreasing, with

$$W^{(k)} \leq 2d_{\max}^{(j)}, \quad j = 0, \dots, k. \tag{5.12}$$

Moreover, assume that at step  $k$  of the *Algorithm MT-SVM* we set  $d_{\max}^{(k)} \leftarrow \hat{d}$ , being  $\hat{d} = d[(\hat{x}, \hat{y}), H(\beta^{(k)}, \beta_0^{(k)}; x, y)]$  with

$$\hat{d} \notin \arg \max_{(x_i, y_i) \in \chi \setminus \{A_k \cup B_k\}} \left\{ d[(x_i, y_i), H(\beta^{(k)}, \beta_0^{(k)}; x, y)] \right\}.$$

Then, we have  $W^{(k)} \leq 2\hat{d}$ .

**Proof** The proof can be found in Pontiggia and Fasano (2021). □

**Observation 5.1** Let be given the nonempty sets  $L_{\max}, L_{\min} \subseteq \chi$ , and consider the *Algorithm MT-SVM* in Table 1. Then, it can be also proved (see Pontiggia and Fasano, 2021) that under mild assumptions the misclassified points when solving  $SVM(A_{k+1}, B_{k+1})$  are a subset of the misclassified points when solving  $SVM(A_k, B_k)$ . This remarks once more the importance of selecting in our procedure  $A_0 = L_{\max}$  and  $B_0 = L_{\min}$ , being  $L_{\max}$  and  $L_{\min}$  likely separable in practice.

Figure 4 shows the overall outcome of *Algorithm MT-SVM* for  $k = 0$ . In particular, we have  $A_0 = L_{\max} = L_{West-North}$  and  $B_0 = L_{\min} = L_{East-South}$ , being  $A_0$  and  $B_0$  linearly separable as in Definition 1. The picture also reports three parallel lines: one central line and



two side lines. The central line represents the separating line computed by  $SVM(A_0, B_0)$ , i.e. the line of equation  $H(\beta^{(0)}, \beta_0^{(0)}; x, y) = 0$ . Conversely, the two side lines delimit the largest region (stripe) where none of the points in  $A_0 \cup B_0$  is included. Finally, the circled points in the picture represent so called *support vectors* (see Cristianini and Shawe-Taylor, 2000), i.e. those points in  $A_0 \cup B_0$  which are the closest to the central line of equation  $H(\beta^{(0)}, \beta_0^{(0)}; x, y) = 0$ . Of course, applying the procedure in the *Algorithm MT-SVM* also for  $k \geq 1$ , the stripe delimited by the two side lines will reduce its thickness, so that the three lines tend to become closer and closer as  $k$  increases. Their slope identifies a possible trend-line (which is expected to change with  $k$ ) for Bitcoin price vs. its SF. As by Figs. 3 and 4 the common slope of the three lines when  $k = 0$  is  $m^* \approx 2.9324$ , which is not pretty close to the value 2.680 obtained with our first proposal in Sect. 4. In particular, similarly to (4.9) we have now the line

$$\ln(\text{price}) = 2.9324 \ln(SF) - 0.35052,$$

so that considering for instance the value of the SF in (4.8) (i.e. at the end of September, 2021), we obtain

$$\text{price} = SF^{2.9324} \cdot e^{-0.35052} = 44.9^{2.9324} \cdot e^{-0.35052} \approx 49, 297\$.$$

Observe that this last Bitcoin price forecast is appreciably different with respect to the one obtained using (4.9). Moreover, observing historical data of Bitcoin prices we can immediately realize that 40, 171\$ was a bit closer to the actual value (i.e.  $\approx 41, 500$  \$) we experienced in practice, with respect to the value 49, 297\$ obtained with our second approach. Figure 4 includes the same information of Fig. 3, and additionally reports also the (remaining) points of the set  $\{(\bar{x}_i, \bar{y}_i), i = 1, \dots, N\}$  which are not in  $L_{\max} \cup L_{\min}$ .

Observing the parallel lines in Figs. 3 and 4, our SVM-based approach suggests also that the two side lines (i.e. the support hyperplanes following the taxonomy of SVMs) have respectively the equations

$$\begin{cases} \ln(\text{price}) = 2.9324 \ln(SF) + (-0.35052 + 0.11629) = 2.9324 \ln(SF) - 0.23423, \\ \ln(\text{price}) = 2.9324 \ln(SF) + (-0.35052 - 0.11629) = 2.9324 \ln(SF) - 0.46681, \end{cases}$$

so that considering again the value of the SF in (4.8), we obtain for Bitcoin price forecast at the end of September 2021 the range of possible values

$$\text{price} \in [44.9^{2.9324} e^{-0.46681}, 44.9^{2.9324} e^{-0.23423}] \equiv [43, 884.8\$ , 55, 376.1\$]. \quad (5.13)$$

This last result reveals that the actual price of 41,500 \$ for Bitcoin, at the end of September 2021, was not included in the interval indicated by (5.13), though the relative error  $\epsilon_{SVM} = [43, 884.8 - 41, 500]/41, 500$ , using our SVM-based approach (lowest extreme in the interval in (5.13)), and the relative error  $\epsilon_{LS} = [40, 171 - 41, 500]/41, 500$ , using a more standard regression approach, are not distant. This indeed explains why we considered to report our numerical experience with reference to the end of September 2021: *it was indeed the case in which our second proposal performs most poorly with respect to the regression analysis*. A couple of final considerations should also be highlighted:

- our second approach in the current section does not require any of the assumptions typically needed when solving linear least squares problems. This implies that our second approach is not subject to any specific validation test;

- as any approach based on linear regression, our first proposal is capable of producing both a point relation between the Bitcoin price and its SF (see (4.9)) and a confidence interval for the price itself. Nevertheless, the calculation of this interval is easy only under suitable theoretical assumptions on the probability distribution of the data (e.g. the normal one which is typically associated with regression formulations). Conversely, our second approach always gives an interval of reliability for the pair of Bitcoin price vs. its SF, identified by the stripe delimited by the side lines in Fig. 3, regardless of the data probability distribution. Thus, in this regard our second proposal shows to some extent higher versatility and possibly robustness of the outcomes. Moreover, it provides a tentative information about long term Bitcoin price, that possibly expert investors may decide to refine or even integrate with their own trusted methods.

## 5.2 Enhancement using bootstrap

Bootstrap represents a widely used technique to infer statistics on a population, by performing re-sampling with replacement of the original dataset associated with the population. More often, after defining a reference measure, bootstrap first implies re-sampling the original dataset so that this measure is recomputed several times. Then, exploiting the Central Limit Theorem (CLT), this measure is treated as a random variable whose simple statistics (i.e. the mean value and the standard deviation) are sought. In this regard we recall that according with the CLT, when i.i.d. random variables are summed up (or averaged), then their properly normalized sum approaches a normal distribution, regardless of the original distribution of the random variables (see Davison and Hinkley, 1997).

Bootstrap can be declined in several practical ways, though it basically reduces to repeatedly selecting a sample in a given population, calculating the statistics associated with some measure, and finally taking the average of the computed statistics. To be more precise, a general bootstrap technique can be summarized as by the following scheme:

- 1 Select the number  $NS$  of re-samplings and the sample size  $SS$  to perform;
- 2 For any  $i = 1$  to  $NS$   
     Compute the  $i$ -th sample (of size  $SS$ ) with replacement of the population  
     Compute  $k \geq 1$  quantities related to the  $i$ -th sample
- 3 Compute the statistics associated with the  $k$  quantities.

Note that such a bootstrap technique is in some sense basic as it does not account for possible presence of serial dependence in the considered time series (in that case, so-called *moving blocks bootstrap* methods should be used). Anyway, the choice to use a plain bootstrap technique is deliberate in order to stress the robustness of our *Algorithm MT-SVM* with respect to specific features of the time series data.

When applying a ML model, whose outcomes yield random quantities, then it is desirable that the results are provided with confidence intervals. We highlight that widely used techniques within ML, like cross-validation, are unable to immediately give confidence intervals for the quantities they report. In our SVM-based ML procedure we used bootstrap to estimate the final mean value and the standard deviation of Bitcoin price, with reference to the following 13 different time-windows for the pairs Bitcoin price vs. its SF:

Time–window 1	January 1st, 2011 – March 31st, 2021
Time–window 2	January 1st, 2011 – April 30th, 2021
Time–window 3	January 1st, 2011 – May 31st, 2021
Time–window 4	January 1st, 2011 – June 30th, 2021
Time–window 5	January 1st, 2011 – July 31st, 2021
Time–window 6	January 1st, 2011 – August 31st, 2021
Time–window 7	January 1st, 2011 – September 30th, 2021
Time–window 8	January 1st, 2011 – October 31st, 2021
Time–window 9	January 1st, 2011 – November 30th, 2021
Time–window 10	January 1st, 2011 – December 31st, 2021
Time–window 11	January 1st, 2011 – January 31st, 2022
Time–window 12	January 1st, 2011 – February 28th, 2022
Time–window 13	January 1st, 2011 – March 31st, 2022

**Table 2** Forecasts of the value of the logarithm for Bitcoin price, along with some related statistics, considering 1 month as prediction horizon

$\ln(P_{t+1 \text{ month}})$	SVM	$\mu_{\text{SVM+B}}$	$\sigma_{\text{SVM+B}}$	$[\cdot, \cdot]_{95\%}$	$\epsilon_{\text{SVM}}$	$\epsilon_{\text{SVM+B}}$
10.9122	10.2932	10.2689	0.1501	[9.9748, 10.5630]	-0.0567	-0.0590
10.4512	11.1474	11.1068	0.2241	[10.6676, 11.5460]	0.0666	0.0627
10.4639	12.2532	11.9440	0.3593	[11.2397, 12.6483]	0.1710	0.1414
10.6347	10.6400	10.4055	0.3344	[9.7500, 11.0610]*	0.0005	-0.0216
10.7975	10.6513	10.4186	0.3657	[9.7018, 11.1354]*	-0.0135	-0.0351
10.6879	10.6566	10.4195	0.3012	[9.8291, 11.0099]*	-0.0029	-0.0251
10.0388	10.6550	10.4042	0.2740	[9.8671, 10.9413]*	0.0614	0.0364
10.9559	11.5767	11.3024	0.2867	[10.7405, 11.8643]*	0.0567	0.0316
10.7605	10.6781	10.4266	0.2814	[9.8751, 10.9781]*	-0.0077	-0.0310
10.5386	10.6764	10.4009	0.3112	[9.7910, 11.0108]*	0.0131	-0.0131
10.5371	10.6619	10.4448	0.3456	[9.7674, 11.1222]*	0.0118	-0.0088
10.7674	10.6795	10.4476	0.3258	[9.8091, 11.0861]*	-0.0082	-0.0297
10.5903	10.6843	10.4560	0.2868	[9.8939, 11.0181]*	0.0089	-0.0127

The asterisk in Column 5 indicates that the true value of Bitcoin price falls within the confidence interval

The dataset associated with each time–window is used to apply the *Algorithm MT – SVM*, in order to generate the parameters of the central line in Fig. 3. Then, the forecast of Bitcoin price is computed in the subsequent three months (e.g. for Time–window 1 we used the data in the interval January 1st, 2011 – March 31st, 2021, and computed a forecast for Bitcoin price at the end of April 2021, May 2021 and June 2021). Finally, this last scheme is repeated for  $NS$  re–samplings of data in the same time–window, so that a statistics using bootstrap will be available. Following standard guidelines from the literature, we also set in our bootstrap framework: number of re–samplings  $NS = 250$ ; size of each sample  $SS =$  the size of dataset in the current time–window.

In Tables 2, 3 and 4 we present the 13 forecasts achieved respectively for each of the considered prediction horizons, i.e. after 1 month (Table 2), after 2 months (Table 3), and after 3 months (Table 4). In particular, in each table: Column 1 reports the true value of Bitcoin price; Columns 2 gives the value of the forecast obtained by the *Algorithm MT–SVM*; Column 3 and Column 4 respectively provide the average value and the standard deviation of the forecast obtained by the *Algorithm MT–SVM* coupled with the bootstrap method;

**Table 3** Forecasts of the value of the logarithm for Bitcoin price, along with some related statistics, considering 2 months as prediction horizon

$\ln(P_{t+2 \text{ months}})$	SVM	$\mu_{\text{SVM+B}}$	$\sigma_{\text{SVM+B}}$	$[\cdot, \cdot]_{95\%}$	$\epsilon_{\text{SVM}}$	$\epsilon_{\text{SVM+B}}$
10.4512	11.1474	11.1156	0.1649	[ 10.7923, 11.4389 ]	0.0666	0.0636
10.4639	11.8058	11.7555	0.2399	[ 11.2853, 12.2257 ]	0.1282	0.1234
10.6347	10.6400	10.4208	0.3014	[ 9.8300, 11.0116 ]*	0.0005	-0.0201
10.7975	10.6513	10.4161	0.3348	[ 9.7598, 11.0724 ]*	-0.0135	-0.0353
10.6879	10.6566	10.4236	0.3659	[ 9.7064, 11.1408 ]*	-0.0029	-0.0247
10.0388	10.6550	10.4180	0.3012	[ 9.8277, 11.0083 ]*	0.0614	0.0378
10.9559	11.5767	11.2721	0.3034	[ 10.6774, 11.8668 ]*	0.0567	0.0289
10.7605	10.6781	10.4547	0.2610	[ 9.9431, 10.9663 ]*	-0.0077	-0.0284
11.5386	10.6764	10.4250	0.2813	[ 9.8736, 10.9764 ]	0.0131	-0.0108
10.5371	10.6619	10.3872	0.3107	[ 9.7783, 10.9961 ]*	0.0118	-0.0142
10.7674	10.6795	10.4615	0.3463	[ 9.7828, 11.1402 ]*	-0.0082	-0.0284
10.5903	10.6843	10.4521	0.3259	[ 9.8133, 11.0909 ]*	0.0089	-0.0130
10.2760	10.6619	10.4348	0.2861	[ 9.8741, 10.9955 ]*	0.0376	0.0155

The asterisk in Column 5 indicates that the true value of Bitcoin price falls within the confidence interval

**Table 4** Forecasts of the value of the logarithm for Bitcoin price, along with some related statistics, considering 3 months as prediction horizon

$\ln(P_{t+3 \text{ months}})$	SVM	$\mu_{\text{SVM+B}}$	$\sigma_{\text{SVM+B}}$	$[\cdot, \cdot]_{95\%}$	$\epsilon_{\text{SVM}}$	$\epsilon_{\text{SVM+B}}$
10.4639	11.8058	11.7681	0.1772	[ 11.4208, 12.1154 ]*	0.1282	0.1246
10.6347	10.3003	10.2722	0.2050	[ 9.8704, 10.6740 ]*	-0.0314	-0.0341
10.7975	10.6513	10.4315	0.3018	[ 9.8399, 11.0231 ]*	-0.0135	-0.0339
10.6879	10.6566	10.4211	0.3350	[ 9.7644, 11.0778 ]*	-0.0029	-0.0250
10.0388	10.6550	10.4221	0.3659	[ 9.7050, 11.1392 ]*	0.0614	0.0382
10.9559	11.5767	11.2867	0.3333	[ 10.6333, 11.9401 ]*	0.0567	0.0302
10.7605	10.6781	10.4259	0.2747	[ 9.8874, 10.9644 ]*	-0.0077	-0.0311
10.5386	10.6764	10.4531	0.2610	[ 9.9416, 10.9646 ]*	0.0131	-0.0081
10.5371	10.6619	10.4113	0.2809	[ 9.8608, 10.9618 ]*	0.0118	-0.0119
10.7674	10.6795	10.4038	0.3113	[ 9.7937, 11.0139 ]*	-0.0082	-0.0338
10.5903	10.6843	10.4660	0.3465	[ 9.7869, 11.1451 ]*	0.0089	-0.0117
10.2760	10.6619	10.4309	0.3251	[ 9.7937, 11.0681 ]*	0.0376	0.0151
9.9173	10.6795	10.4515	0.2866	[ 9.8897, 11.0133 ]*	0.0769	0.0539

The asterisk in Column 5 indicates that the true value of Bitcoin price falls within the confidence interval

Column 5 shows the 95% confidence interval computed using the average value and standard deviation in Column 3 and Column 4; Columns 6 and 7 present the relative errors respectively associated to the forecasts from the *Algorithm MT-SVM* and from the *Algorithm MT-SVM* coupled with the bootstrap method.

**Observation 5.2** *We highlight that for each forecast horizon, most of the true values to predict fall within the confidence interval, namely 10 out of 13 for the one-month horizon, 10 out of 13 for the two-month horizon, and 13 out of 13 for the three-month horizon. Furthermore, the other actual values not falling within this interval are generally close to it. Given the*

high volatility that the Bitcoin price has often experienced, our outcomes can be considered more than satisfactory. Lastly note that, unlike what usually happens when forecasting time series data, the above three ratios 10/13, 10/13 and 11/13 do not decrease as the forecast horizon length increases.

**Observation 5.3** *We stress that for all the forecast horizons, the average of the relative error associated to the Algorithm MT–SVM coupled with the bootstrap method (i.e. the average of the values in the last column of Tables 2, 3 and 4) is lower than the average of the relative error associated to the Algorithm MT–SVM alone (i.e. the average of the values in the second last column of Tables 2, 3 and 4), namely: 0.28% vs. 2.31% for the one-month horizon, 0.72% vs. 2.71% for the two-month horizon, and 0.56% vs. 2.54% for the three-month horizon. We also point out that the standard deviations of both the relative errors are very close between them in each of the three considered forecast horizons.*

All these results give numerical evidence that in this context coupling a bootstrap method with a SVM–based approach can improve the quality of the forecast.

Lastly, as stated in the end of Sect. 4, we consider the linear regression model for comparative purposes with our SVM–based approach enhanced through bootstrap. In particular, similarly to what done in the previous analyses, we applied the linear regression model to predict the future price of Bitcoin with reference to the above same 13 time-windows and for each of the above same forecast horizons.

All the main findings strongly suggest that the *Algorithm MT–SVM* coupled with the bootstrap method performs better than the linear regression model. Figure 5 shows that all the 95% confidence intervals related to the former approach (graphically represented in red) are strictly contained in the corresponding 95% confidence intervals related to the linear regression model (graphically represented in black), this for all the considered time-windows and all the forecast horizons.

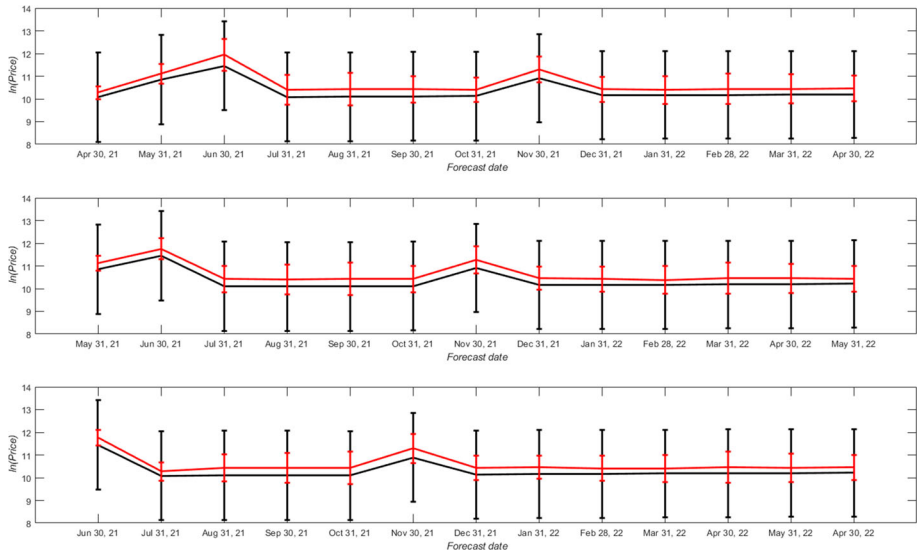
**Observation 5.4** *We highlight that for all the forecast horizons, the average of the standard error associated to the Algorithm MT–SVM coupled with the bootstrap method is lower than the average of the standard error associated to the linear regression model, namely: 0.2959 vs. 0.9935 for the one-month horizon, 0.2941 vs. 0.9935 for the two-month horizon, and 0.2926 vs. 0.9935 for the three-month horizon.*

**Observation 5.5** *We point out that for all the forecast horizons, the average of the relative error associated to the Algorithm MT–SVM enhanced using the bootstrap method is lower (absolute value) than the average of the relative error associated to the linear regression model, namely: 0.28% vs. –2.49% for the one-month horizon, 0.72% vs. –2.00% for the two-month horizon, and 0.56% vs. –2.12% for the three-month horizon. We also stress that the standard deviations of both the relative errors are very close between them in each of the three considered forecast horizons.*

These findings provide numerical evidences that, at least in this forecasting context, coupling a bootstrap method with a SVM–based approach can improve the quality of the performance with respect to the one obtained from the use of a linear regression model.

## 6 Conclusions and future work

This paper contributes to possibly investigate reliable long term models for Bitcoin price forecast. We recall that in the last decade Bitcoin has become an observed digital asset, for



**Fig. 5** 95% confidence intervals related to the *Algorithm MT-SVM* enhanced using the bootstrap method (graphically represented in red) and to the linear regression model (graphically represented in black), considering: 1 month as prediction horizon (upper figure); 2 months as prediction horizon (middle figure); 3 months as prediction horizon (lower figure). Each red confidence interval is strictly contained within the corresponding black confidence interval. The two broken lines in each figure represent the forecasts of  $\ln(\text{price})$  as provided by the two methods, respectively

possible investments by both private and institutional stakeholders. In this paper we have specifically proposed a couple of models, following two different perspectives. The first one was suggested by considering a more standard regression analysis, while the second one is definitely novel in the literature, being obtained by combining a preliminary multiobjective approach with a ML scheme, where a sequence of SVMs is indeed considered.

We are persuaded that several factors strongly contribute to affect long term Bitcoin price, other than the SF. Nevertheless, the dependency of Bitcoin price on its SF was suggested by several authors, and is also considered in the first proposal of this paper. In this regard, a natural future extension for our analysis will have to consider a set of multiple elements encompassing SF, so that our ML-based proposal will have to be enhanced.

**Acknowledgements** Marco Corazza and Giovanni Fasano wish to thank *Istituto Nazionale di Alta Matematica* (INdAM), Giovanni Fasano wishes to thank *Consiglio Nazionale delle Ricerche – Istituto di Ingegneria del Mare (CNR-INM)*, for the support they received. Giovanni Fasano also thanks his three lawyer friends Maria, Maristella and Massimo, whose valuable perspective considerably contributed to inspire the contents of this paper.

## References

- Aggarwal, D., Chandrasekaran, S., & Annamalai, B. (2020). A complete empirical ensemble mode decomposition and support vector machine-based approach to predict Bitcoin prices. *Journal of Behavioral and Experimental Finance*, 27, 100335.
- Baur, D. G., & Dimpfl, T. (2021). The volatility of Bitcoin and its role as a medium of exchange and a store of value. *Empirical Economics*, 61, 2663–2683.

- Bernstein, D. S. (2009). *Matrix mathematics: Theory, facts, and formulas* (2nd ed.). Princeton University Press.
- Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society Series B (Methodological)*, 26(2), 211–252.
- Buy Bitcoin Worldwide (2019). Bitcoin stock to flow model live chart, <https://www.buybitcoinworldwide.com>
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.
- Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press.
- Deng, N., Tian, Y., & Zhang, C. (2013). *Support vector machines - optimization based theory, algorithms, and extensions*. Chapman and Hall or CRC.
- Graybill, F. A., & Iyer, H. K. (1994). *Regression analysis: Concepts and applications*. Duxbury Press.
- Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning* (2nd ed.). Springer. <https://www.blockchain.com/charts/total-bitcoins>
- Jarque, C. M., & Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review*, 55(2), 163–172.
- Marsland, S. (2015). *Machine learning: An algorithmic perspective* (2nd ed.). CRC Press Taylor & Francis Group.
- Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system, <http://www.bitcoin.org/bitcoin.pdf>
- Nakamoto, S., et al. [Anonymous] (2014). Bitcoin source code – amount constraints, <https://github.com/bitcoin/bitcoin>
- PlanB (2019). Modeling Bitcoin's Value with Scarcity, <https://medium.com/@100trillionUSD/modeling-bitcoins-value-with-scarcity-91fa0fc03e25>
- Pontiggia, A., & Fasano, G. (2021). Data analytics and machine learning paradigm to gauge performances combining classification, ranking and sorting for system analysis, Working Paper 05/2021, Department of Management, University Ca' Foscari of Venice.
- Sreekanth Reddy, L., & Sriramya, P. (2020). A research on Bitcoin price prediction using machine learning algorithms. *International Journal of Scientific & Technology Research*, 9(4), 1600–1604.
- Various Authors (2011). Total circulating Bitcoin, <https://www.blockchain.com/charts/total-bitcoins>
- Vapnik, V. (1995). *The nature of the statistical learning theory*. Springer.
- Vapnik, V. (1998). *Statistical learning theory*. Wiley.
- Vigna, P., & Casey, M.J. (2015). *The age of cryptocurrency: How bitcoin and digital money are challenging the global economic order* (1st ed.). St. Martin's Press.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.