Article

Bitcoin Analysis and Forecasting through Fuzzy Transform

Maria Letizia Guerra[1,*], Laerte Sorini[2], Luciano Stefanini[3]

[1] *Department of Statistical Sciences "Paolo Fortunati"*, University of Bologna, Italy. (*e-mail*: mletizia.guerra@unibo.it)
[2] *Department of Economics, Society, Politics*, University of Urbino Carlo Bo, Urbino, Italy. (*e-mail*: laerte.sorini@uniurb.it)
[3] *Department of Economics, Society, Politics*, University of Urbino Carlo Bo, Urbino, Italy. (*e-mail*: luciano.stefanini@uniurb.it)
* *Correspondence*: mletizia.guerra@unibo.it

Abstract: Sentiment analysis to characterize properties of Bitcoin prices and their forecasting is here developed thanks to the capability of the Fuzzy transform to capture stylized facts and mutual connections between time series having different nature. Six years of daily Bitcoin Prices and Google Trends are analyzed in order to establish new perspectives in the management of their dynamics.

Keywords: F-transform, BITCOIN, Clustering, Sentiment Analysis

1. Introduction

The goal of the present paper is to analyze the connection between Bitcoin returns and the level of interest in the world wide web through two types of regressions: quantile and expectile through inverse and direct F-transform. The Bitcoin observed time series is modelled through fuzzy-valued functions, whose level-cuts can be interpreted in the setting of expectile and quantile fuzzy regressions; these last are introduced in ([1], [2]) as non-parametric smoothing methodologies and are constructed by defining fuzzy-valued expectile (L_2-norm) and quantile (L_1-norm) extensions of the F-transforms. We recall that F-transform has been introduced by [3] (see also [4], [5], [6], [7])

Quantile regression is also applied in [8] to show that Bitcoin reacts positively to uncertainty at both higher quantiles and shorter frequency movements of Bitcoin returns.

Following recent research on financial time series, where the properties of quantile and expectile modelling are discussed with respect to coherent and elicitable risk measures (see [9], [10]), expectile methods seem to compete favourably with quantiles; furthermore, some recent papers (e.g., [11]) suggest to adopt L_p-norm based procedures with p between 1 and 2 (e.g. $p \in \{1.25, 1.5, 1.75\}$) in order to adjust for tail behaviour according to robust Extreme Value Theory.

Literature about Bitcoin has hugely grown in the last years and some papers deserve a citation. An exhaustive analysis of Bitcoin and its statistical properties are explored in [12] by a comparison with standard currencies dynamics. Yermack in [13] shows that Bitcoin does not satisfy the three main properties called medium of exchange, unit of account and store of value, concluding that it is not a currency but rather a speculative asset. However, a debate is still open about the nature of Bitcoin and many hints can be found in [14], [15] and [16]. A shared property by financial instruments is the day-of-the-week effect and in [17] the same effect is proved for Bitcoin returns and volatility through OLS and GARCH model. The possibility of global economic policy uncertainty to produce valid information to improve the prediction of returns and volatility in the Bitcoin market is detailed in [18].

Research on possible forecasting models to be used as decision support tools in investment strategies is more recent; in [19], monthly data are considered and it is shown that the predictive ability of the internet-based economic uncertainty related queries index is statistically stronger than the measure of uncertainty derived from newspapers in predicting Bitcoin returns. The nexus between Bitcoin prices and market sentiment is further studied in many papers: in [20] sentiment is shown to explain about 2.5% to 5% of the unusual level of price clustering in Bitcoin. In [21] the cross-correlations between Google Trends and Bitcoin market is analyzed through the Multifractal Detrended Cross-correlation Analysis (MF-DCCA) method and in [22], within the more general context of Dow Jones Industrial Average, it is shown that Google searches are power-law correlated with Hurst exponents between 0.8 and 1.1 and that there is no universal relationship between the online search queries and some financial measures. In [23] investor sentiment regarding Bitcoin is introduced because of its significant information for explaining changes in Bitcoin volatility for future periods; on this basis Bitcoin is proved to be an investment asset with high volatility and dependence on investor sentiment rather than a monetary asset.

In a more general framework, in [24], interactions between (mass) media reporting and financial market movements are measured with particular focus on the property of sentiment as a predictors of securities' prices.

A non-parametric forecasting model based on technical analysis is presented in [25] focusing on the presence of predictive local non-linear trends that reflect the speculative nature of crypto-currency trading. In [26] a computational intelligence technique that uses a hybrid Neuro-Fuzzy controller is introduced in order to forecast the direction in the change of the daily price of Bitcoin and its performance is the best when compared with two other computational intelligence models.

Forecasting of Bitcoin risk measures is developed in [27] by comparing predictability of the one-step-ahead volatility with Value-at-Risk using several volatility models.

Many other authors approach general cryptocurrencies properties. For example, in [28] there is evidence that Bitcoin is the most influential among digital coins both as a transmitter toward digital currencies and as a receiver of spillovers from virtual and traditional instruments. An extended analysis is also presented in [29] where the four crypto-currencies Bitcoin, Ethereum, Ripple and Litecoin are predicted through a combination of eight models revealing that a combination of stochastic volatility and a student-t distribution gives the best results. The same topic of Bitcoin-realized volatility forecasting is studied in [30] where conventional regression models are substituted by least squares model-averaging methods and no investor sentiment is modelled.

In [31] and [32] a continuous time model for Bitcoin price dynamics is studied in order to detect bubbles; regarding the existence of a bubble, in [33] it is proved it holds from early 2013 to mid 2014, but, not in late 2017 as it was supposed. Evidence of bubbly Bitcoin behavior, mainly in the 2017–2018 period, is shown in [34], where it is also proved that economic policy uncertainty and stock market volatility play the most important role in bitcoin values.

The evidence for the Bitcoin bubble is confirmed in [35] through the empirical validation of three properties: volume of trading is mainly explained in terms of price dynamics, trading is based exclusively on past prices and the price of Bitcoin is an explosive process.

In [36] an thorough analysis is conducted: several alternative univariate and multivariate models for point and density forecasting of crypto-series are compared, finding statistically significant improvements in point forecasting when using combinations of univariate models, and in density forecasting when relying on the selection of multivariate models.

Various deep learning-based Bitcoin price prediction models are studied in [37] using Bitcoin blockchain information; regression and classification problems are addressed in the sense that the first predicts the future Bitcoin price and the second one predicts whether or not the future price will go up or down.

In the case of Bitcoin prices using high frequency data, in [38] it is shown that it exists a large degree of multi-fractality in all examined time intervals which can be attributed to the high kurtosis and the fat distributional tails of the series returns; in [39] there is evidence about the leverage effect as the most powerful effect in volatility forecasting; volatility is also analysed in [40] in terms of the property of the long memory parameter to be significant and quite stable for both unconditional and conditional volatilities at different time scales. Extending the study to several high frequency cryptocurrencies data, in [41] the investigation on stylized facts is developed in terms of the Hurst exponent of dependence between four different cryptocurrencies.

Also in [42] multi-fractality of Bitcoin time series is investigated, confirming that both temporal correlation and the fat-tailed distribution are the main sources, in addition in [43] a possible use of multi-fractal parameters in Technical Analysis is suggested.

The paper is organized into six sections: preliminary facts on methodology concerning Fuzzy Transform are presented in section two, while the empirical experiments concerning Bitcoin prices are detailed in sections three and four. Possible forecasting techniques are shown in section five thereafter the last one closes with some hints for future research paths.

2. Fuzzy-Transform Smoothing

We introduced in [44] and then we enhance in [1] two non-parametric smoothing methodologies called expectile and quantile Fuzzy-transform; the first one is based on the classical direct F-transform and it is obtained by minimizing a least squares (L$_2$-norm) operator while the second one is based on the L$_1$-type direct F-transform and it is obtained by minimizing an L$_1$-norm operator.

Some preliminary notions compose the research framework: a fuzzy set is a mapping $u : \mathbb{R} \longrightarrow [0,1]$ and a fuzzy interval is a fuzzy set on $\mathbb{R}$ with the properties that the mapping u is *(i)* normal ($\exists \widehat{x} \in \mathbb{R}$ with $u(\widehat{x}) = 1$), *(ii)* upper semi-continuous, *(iii)* fuzzy convex ($u(\lambda x' + (1-\lambda)x'') \geq \min\{u(x'), u(x'')\}$ for all $\lambda \in [0,1]$), *(iv)* $cl\{x|u(x) > 0\}$ is a compact interval. A consequence of (ii) and (iii) is that the α-cuts $[u]_\alpha = \{x|u(x) \geq \alpha\} = [u^-_\alpha, u^+_\alpha]$ are compact intervals for all $\alpha \in]0,1]$. The 1-cut is the core $[u]_1 = \{x|u(x) = 1\}$ of u; the interval $[u]_0 = cl(\{x|u(x) > 0\})$ is the 0-cut of u. A fuzzy interval is a fuzzy number if its core is a singleton $[u]_1 = \{\widehat{u}\}$ with $\widehat{u} \in \mathbb{R}$.

The space of real fuzzy intervals is denoted $\mathbb{R}_\mathcal{F}$ and the mapping $u \in \mathbb{R}_\mathcal{F}$ satisfies what follows:

$$u(x) = \begin{cases} 0 & \text{if} \quad x \notin [u]_0 \\ \sup\{\alpha | x \in [u]_\alpha\} & \text{if} \quad x \in [u]_0. \end{cases}$$

For a given real compact interval $[a,b]$, a generalized r-partition is defined by a triplet $(\mathbb{P}, \mathbb{A}, r)$ where $r \geq 1$ is integer, $\mathbb{P} = \left\{x_j = a + \frac{j-1}{n-1}(b-a); j = 1,2,...,n\right\}$, $n \geq 2$, is a uniform decomposition of $[a,b]$; for simplicity of notation, if $r > 1$ we extend $\mathbb{P}$ by adding $r-1$ points $x_{1-j} = a - j\frac{b-a}{n-1}, j = 1,...,r-1$ on the left of a and $r-1$ points $x_{n+j} = b + j\frac{b-a}{n-1}, j = 1,...,r-1$ on the right of b. The second term of the triplet is a family $\mathbb{A} = \{A_{-r+2},...,A_1,A_2,...,A_n,...,A_{n+r-1}\}$ of $n+2r-2$ continuous fuzzy sets on $\mathbb{R}$, called *basic functions*, that satisfy the following condition for all $x \in [a,b]$

$$\sum_{k=-r+2}^{n+r-1} A_k(x) = r$$

and are such that $A_k(x_k) = 1$, for $k = 2-r,...,1,2,...,n,...,n+r-1$, $A_k(x) = 0$ for all $x \notin]x_{k-r}, x_{k+r}[$.

If $r = 1$, the partition $(\mathbb{P}, \mathbb{A}, 1)$ will be simply denoted by $(\mathbb{P}, \mathbb{A})$.

Families of basic functions can be obtained in terms of increasing shape functions such as rational splines of the form

$$L(t; \beta_0, \beta_1) = \frac{t^2 + \beta_0 t(1-t)}{1 + (\beta_0 + \beta_1 - 2)t(1-t)}, \qquad t \in [0,1] \tag{1}$$

with real parameters $\beta_0 \geq 0$, $\beta_1 \geq 0$; the Hermite-type conditions $L(0) = 0$, $L(1) = 1$, $L'(0) = \beta_0$, $L'(1) = \beta_1$ are satisfied and $L'(t) \geq 0$ for all $t \in [0,1]$. By any pair of non-negative values β_0, β_1, a large number of shape functions can be generated; for example, if $\beta_0 + \beta_1 = 2$ (with $0 \leq \beta_0 \leq 2$, $\beta_1 = 2 - \beta_0$)

we have a quadratic function $L(t) = (1-\beta_0)t^2 + \beta_0 t$, e.g., $L(t;2,0) = 2t - t^2$, $L(t;0,2) = t^2$ and $L(t;1,1) = t$ is linear.

Each basic function A_k, $k = 2-r, ..., 1, 2, ..., n, ..., n+r-1$, increasing on $[x_{k-r}, x_k]$ and decreasing on $[x_k, x_{k+r}]$, is obtained by translating $t \to L(t, \beta_0, \beta_1)$ and $t \to L(1-t, \beta_0, \beta_1)$ from $[0,1]$ onto $[x_{k-r}, x_k]$ and $[x_k, x_{k+r}]$, respectively (each A_k is finally extended to $\mathbb{R}$ by setting $A_k(x) = 0$ on the left of x_{k-r} and on the right of x_{k+r}).

2.1. L_2-norm F-transform in expectile smoothing

We just recall the discrete version of the direct F-transform.

Definition 1. *(from [3]) Given a set of m values* $\mathbf{Y} = \{(t_i, f_i) | \quad t_i \in [a,b], \quad i = 1, ..., m\}$ *of a function* $f : [a,b] \longrightarrow \mathbb{R}$ *and a fuzzy partition* $(\mathbb{P}, \mathbb{A})$ *of* $[a,b]$ *such that each subinterval* $[x_{k-1}, x_{k+1}]$ *contains at least one point* t_i *in its interior (so that* $\sum_{i=1}^{m} A_k(t_i) > 0$ *for all k), then the discrete direct* L_2*-type F-transform of* $\mathbf{Y}$ *with respect to* $(\mathbb{P}, \mathbb{A})$ *is the n-tuple of real numbers* $(F_1, ..., F_n)$ *where each component* F_k *minimizes the function* $\Phi_k(y) = \sum_{i=1}^{m} |f_i - y|^2 A_k(t_i)$, $k = 1, 2, ..., n$. *The associated inverse F-transform function is defined by* $\widehat{f}_{(\mathbb{P},\mathbb{A})}(x) = \sum_{k=1}^{n} F_k A_k(x)$ *for all* $x \in [a,b]$.

More generally, we consider a generalized $r-$partition $(\mathbb{P}, \mathbb{A}, r)$ and substitute the direct F-transform components F_k, with an $(n+2r-2)$-tuple of polynomials of order $q \geq 0$, say $(\varphi_{2-r}(x), ..., \varphi_{n+r-1}(x))$ with $\varphi_k(x) = F_{k,0} + F_{k,1}(x - x_k) + ... + F_{k,q}(x - x_k)^q$, $k = 2-r, ..., n+r-1$. The $q+1$ coefficients $F_{k,j}$, $j = 0, 1, ..., q$ are obtained, for fixed k, by minimizing the function $\Phi_k(y_0, ..., y_q) = \sum_{i=1}^{m} \left| f_i - \left(y_0 + y_1(t_i - x_k) + ... + y_q(t_i - x_k)^q\right)\right|^2 A_k(t_i)$ with respect to the parameters $y_0, ..., y_q$, under the assumption that, for each k, the data points (t_i, f_i) with t_i in the interval $\left[x_{max(a,k-r)}, x_{min(b,k+r)}\right]$ produce a unique optimal solution. The details are shown in [1].

The corresponding inverse F-transform function is given by

$$\widehat{f}^{(q)}_{(\mathbb{P},\mathbb{A},r)}(x) = \frac{1}{r} \sum_{k=2-r}^{n+r-1} A_k(x) \varphi_k(x; F_{k,0}, ..., F_{k,q}) \qquad \textit{for } x \in [a,b]. \tag{2}$$

Consider for simplicity the F-transform of order zero ($q = 0$); the inverse F-transform function becomes

$$\widehat{f}_{(\mathbb{P},\mathbb{A},r)}(x) = \frac{1}{r} \sum_{k=2-r}^{n+r-1} F_k A_k(x) \qquad \textit{for } x \in [a,b] \tag{3}$$

with the $(n+2r-2)$-tuple of the direct F-transform $(F_{2-r}, ..., F_{n+r-1})$.

We recall from [1] that the *expectile direct F-transform* components are defined to be the minimizers of the strictly convex functions, for $k = 2-r, ..., 1, ..., n, ..., n+r-1$ and $\omega \in]0,1]$,

$$\Phi_{k,\omega}(\mu) = \sum_{j=1}^{m} w_j(\omega; \mu) \left(f_j - \mu\right)^2 A_k(t_j)$$

where

$$w_j(\omega; \mu) = \begin{cases} \omega & \text{if} \quad f_j > \mu \\ 1 - \omega & \text{if} \quad f_j \leq \mu \end{cases} .$$

The value $\mu_k = \mu_k(\omega)$ (depending on ω) is the expectile for the asymmetry parameter $\omega \in]0,1]$ and if $\omega = \frac{1}{2}$ we obtain the direct F-transform component F_k in equation (3).

Proposition 1. *([1])Given the set of minimizers $\{\mu_k(\omega)\,|\omega\in]0,1[\}$ of $\Phi_{k,\omega}(\mu)$, consider $\alpha\in[0,1]$; then the compact intervals*

$$U_{k,\alpha}=\begin{cases}\left\{\mu_k\left(\frac{1}{2}\right)\right\} & \text{if} \quad \alpha=1\\ \left[\mu_k\left(\frac{\alpha}{2}\right),\mu_k\left(1-\frac{\alpha}{2}\right)\right] & \text{if} \quad \alpha\in]0,1[\\ cl\left(\bigcup\limits_{\beta>0}U_{k,\beta}\right) & \text{if} \quad \alpha=0\end{cases} \tag{4}$$

define the α-cuts of a fuzzy number $u_k\in\mathbb{R}_{\mathcal{F}}$ having membership function

$$u_k(x)=\begin{cases}\sup\{\alpha|x\in U_{k,\alpha}\} & \text{if} \quad x\in U_{k,0}\\ 0 & \text{if} \quad x\notin U_{k,0}\end{cases}.$$

Definition 2. *Given a set of m points $Y=\{(t_i,f_i);\,t_i\in[a,b],\,i=1,...,m\}$ and a fuzzy r-partition $(\mathbb{P},\mathbb{A},r)$ of $[a,b]$, the $(n+2r-2)$-vector of fuzzy numbers*

$$\mathbf{F}_{(\mathbb{P},\mathbb{A},r)}=(F_{2-r},...,F_{n+r-1}),$$

where each fuzzy interval F_k has α-cuts $U_{k,\alpha}$ given by (4) in Proposition 1, is called the discrete direct expectile fuzzy-valued F-transform *of f with respect to $(\mathbb{P},\mathbb{A},r)$, based on the data-set Y. The corresponding* inverse expectile fuzzy-valued iF-transform *is the fuzzy-valued function defined by*

$$\widehat{f}_{(\mathbb{P},\mathbb{A},r)}(x)=\frac{1}{r}\sum_{k=2-r}^{n+r-1}F_kA_k(x) \text{ for } x\in[a,b]. \tag{5}$$

The fuzzy-valued function $\widehat{f}_{(\mathbb{P},\mathbb{A},r)}(x)$ is well defined as indeed each basic function A_k has non-negative values for each $x\in[a,b]$. The α-cuts $U_{k,\alpha}$ of F_k will be denoted by

$$[F_k]_\alpha=\left[F^-_{k,\alpha},F^+_{k,\alpha}\right],\quad k=2-r,...,n+r-1,\quad \alpha\in[0,1]$$

and the α-cuts of the fuzzy-valued function $\widehat{f}_{(\mathbb{P},\mathbb{A},r)}(x)$, $x\in[a,b]$, will be given by

$$\left[\widehat{f}_{(\mathbb{P},\mathbb{A},r)}(x)\right]_\alpha=\left[\frac{1}{r}\sum_{k=2-r}^{n+r-1}F^-_{k,\alpha}A_k(x)\;,\;\frac{1}{r}\sum_{k=2-r}^{n+r-1}F^+_{k,\alpha}A_k(x)\right],\quad \alpha\in[0,1].$$

When $\alpha=1$ we obtain the standard direct $F-$transform and the standard $iF-$transform function, corresponding to the core of the fuzzy-valued iF-transform.

2.2. L_1-norm F-transform in quantile smoothing

The L_1-norm direct and inverse F-transform are defined as follows.

Definition 3. *Given a set of m values $\mathbf{Y}=\{(t_i,f_i)|\quad t_i\in[a,b],\quad i=1,...,m\}$ of a function $f:[a,b]\longrightarrow\mathbb{R}$ and a fuzzy partition $(\mathbb{P},\mathbb{A})$ of $[a,b]$ such that each subinterval $[x_{k-1},x_{k+1}]$ contains at least one point t_i in its interior, then the discrete direct L_1-type F-transform of $\mathbf{Y}$ with respect to $(\mathbb{P},\mathbb{A})$ is the n-tuple of real numbers $(G_1,...,G_n)$ where each component G_k minimizes the function $\Psi_k(y)=\sum\limits_{i=1}^{m}|f_i-y|\,A_k(t_i)$, $k=1,2,...,n$. The associated inverse F-transform function is defined by $\widetilde{f}_{(\mathbb{P},\mathbb{A})}(x)=\sum\limits_{k=1}^{n}G_kA_k(x)$ for all $x\in[a,b]$.*

Also in this case, we consider a generalized $r-$partition $(\mathbb{P},\mathbb{A},r)$ and substitute the direct F-transform components G_k, with an $(n+2r-2)$-tuple of polynomials of order $q\geq 0$, say $(\psi_{2-r}(x),...,\psi_{n+r-1}(x))$ with $\psi_k(x)=G_{k,0}+G_{k,1}(x-x_k)+...+G_{k,q}(x-x_k)^q$, $k=2-r,...,n+r-1$.

The $q+1$ coefficients $G_{k,j}, j = 0, 1, ..., q$ are obtained, for fixed k, by minimizing the function $\Psi_k(y_0, ..., y_q) = \sum_{i=1}^{m} \left| f_i - (y_0 + y_1(t_i - x_k) + ... + y_q(t_i - x_k)^q) \right| A_k(t_i)$ with respect to the parameters $y_0, ..., y_q$ (see details in [1]).

The corresponding L_1-type inverse F-transform function is given by

$$\widetilde{f}^{(q)}_{(\mathbb{P},\mathbb{A},r)}(x) = \frac{1}{r}\sum_{k=2-r}^{n+r-1} A_k(x)\psi_k(x; G_{k,0}, ..., G_{k,q}) \quad \textit{for } x \in [a,b]. \tag{6}$$

The inverse F-transform function of order zero ($p = 0$) becomes

$$\widetilde{f}_{(\mathbb{P},\mathbb{A},r)}(x) = \frac{1}{r}\sum_{k=2-r}^{n+r-1} G_k A_k(x) \quad \textit{for } x \in [a,b] \tag{7}$$

with the $(n+2r-2)$-tuple of the L_1-type direct F-transform $(G_{2-r}, ..., G_{n+r-1})$.

We recall from [1] that the *quantile direct F-transform* is defined in terms of the minimizers of the convex functions, for $k = 1, ..., n$ and $\omega \in]0,1]$,

$$\Psi_{k,\omega}(\eta) = \sum_{j=1}^{m} w_j(\omega; \eta) \left| f_j - \eta \right| A_k(t_j)$$

where

$$w_j(\omega;\eta) = \begin{cases} \omega & \text{if} \quad f_j > \eta \\ 1-\omega & \text{if} \quad f_j \le \eta \end{cases}.$$

As detailed in [1], the minimization of $\Psi_{k,\omega}(\eta)$ produces the family of compact intervals, for $\alpha \in [0,1]$ and $\omega \in \left\{\frac{\alpha}{2}, 1 - \frac{\alpha}{2}\right\}$,

$$V_{k,\alpha} = \begin{cases} \left\{\eta_k\left(\frac{1}{2}\right)\right\} & \text{if} \quad \alpha = 1 \\ \left[\eta_k\left(\frac{\alpha}{2}\right), \eta_k\left(1-\frac{\alpha}{2}\right)\right] & \text{if} \quad \alpha \in]0,1[\\ cl\left(\bigcup\limits_{\beta>0} V_{k,\beta}\right) & \text{if} \quad \alpha = 0; \end{cases} \tag{8}$$

and we obtain the α-cuts of a fuzzy number $v_k \in \mathbb{R}_{\mathcal{F}}$ having membership function

$$v_k(x) = \begin{cases} \sup\{\alpha | x \in V_{k,\alpha}\} & \text{if} \quad x \in V_{k,0} \\ 0 & \text{if} \quad x \notin V_{k,0}. \end{cases}$$

Definition 4. *Given a set of m points $Y = \{(t_i, f_i); t_i \in [a,b], i = 1, ..., m\}$ and a fuzzy r-partition $(\mathbb{P}, \mathbb{A}, r)$ of $[a,b]$, the $(n+2r-2)$-vector of fuzzy numbers*

$$\mathbf{G}_{(\mathbb{P},\mathbb{A},r)} = (G_{2-r}, ..., G_{n+r-1}),$$

where each fuzzy interval G_k has α-cuts $V_{k,\alpha}$, is called the discrete direct quantile fuzzy transform *of Y with respect to $(\mathbb{P}, \mathbb{A}, r)$.*

The corresponding *inverse quantile fuzzy transform* of f is the fuzzy-valued function defined by

$$\widetilde{f}_{(\mathbb{P},\mathbb{A},r)}(x) = \frac{1}{r}\sum_{k=2-r}^{n+r-1} G_k A_k(x) \quad \text{for } x \in [a,b].$$

Denoting the α-cuts $V_{k,\alpha}$ of G_k by

$$[G_k]_\alpha = \left[G^-_{k,\alpha}, G^+_{k,\alpha}\right], \quad k = 2-r, ..., n+r-1, \quad \alpha \in [0,1],$$

then, the α-cuts of the corresponding fuzzy-valued function $\widetilde{f}_{(\mathbb{P},\mathbb{A},r)}(x)$, $x \in [a,b]$, will be given by

$$\left[\widetilde{f}_{(\mathbb{P},\mathbb{A},r)}(x)\right]_\alpha = \left[\frac{1}{r}\sum_{k=2-r}^{n+r-1} G^-_{k,\alpha}A_k(x)\ , \frac{1}{r}\sum_{k=2-r}^{n+r-1} G^+_{k,\alpha}A_k(x)\right], \quad \alpha \in [0,1].$$

2.3. General L_p-norm based discrete F-transform

The general L_p-norm based F-transform has been analysed in detail in [45] for the continuous case. Its interest in time series applications is motivated by recent literature on tail behaviour of economic and financial time series (see, e.g., [11]) and in modelling risk measures ([9], [10]): L_p-norm estimation with $1 < p < 2$ has been suggested to balance robustness and fitting properties.

For a data-set of m values $\mathbf{Y} = \{(t_i, f_i) | \quad t_i \in [a,b], \quad i = 1, ..., m\}$ and a generalized $r-$partition $(\mathbb{P},\mathbb{A},r)$, the L_p-norm direct F-transform is an $(n+2r-2)$-tuple of polynomials of order $q \geq 0$ $(\vartheta_{2-r}(x), ..., \vartheta_{n+r-1}(x))$ with $\vartheta_k(x) = \theta_{k,0} + \theta_{k,1}(x-x_k) + ... + \theta_{k,q}(x-x_k)^q$, $k = 2-r, ..., n+r-1$, where, this time, the $q+1$ coefficients $\theta_{k,j}, j = 0,1,...,q$, are obtained by minimizing the functions $\Theta^{(p)}_k(\theta_0, ..., \theta_q) = \sum\limits_{i=1}^{m} \left|f_i - (\theta_0 + \theta_1(t_i - x_k) + ... + \theta_q(t_i-x_k)^q)\right|^p A_k(t_i)$ with respect to the parameters $\theta_0, ..., \theta_q$.

If $q = 0$, the direct fuzzy-valued F-transform components, for a fixed $p \in]1,2[$, are obtained by minimizing the strictly convex functions, for $k = 2-r, ..., 1, ..., n, ..., n+r-1$ and $\omega \in]0,1]$,

$$\Theta^{(p)}_{k,\omega}(\theta) = \sum_{j=1}^{m} w_j(\omega;\theta)\left|f_j - \theta\right|^p A_k(t_j)$$

where

$$w_j(\omega;\theta) = \begin{cases} \omega & \text{if} \quad f_j > \theta \\ 1-\omega & \text{if} \quad f_j \leq \theta \end{cases}.$$

The minimizers $\vartheta_k = \vartheta_k(\omega)$ (depending on ω) define the L_p-norm direct F-transform components for the asymmetry parameter $\omega \in]0,1]$. We have (the proof is similar to the case $p = 2$ in [1])

Proposition 2. *Given the set of minimizers $\{\vartheta_k(\omega) | \omega \in]0,1[\}$ of $\Theta^{(p)}_{k,\omega}(\theta)$, consider $\alpha \in [0,1]$; then, the compact intervals*

$$W_{k,\alpha} = \begin{cases} \left\{\theta_k\left(\frac{1}{2}\right)\right\} & \textit{if} \quad \alpha = 1 \\ \left[\theta_k\left(\frac{\alpha}{2}\right), \theta_k\left(1-\frac{\alpha}{2}\right)\right] & \textit{if} \quad \alpha \in]0,1[\\ cl\left(\bigcup\limits_{\beta>0} W_{k,\beta}\right) & \textit{if} \quad \alpha = 0 \end{cases} \tag{9}$$

define the α-cuts of a fuzzy number $w_k \in \mathbb{R}_\mathcal{F}$ having membership function

$$w_k(x) = \begin{cases} \sup\{\alpha | x \in W_{k,\alpha}\} & \textit{if} \quad x \in W_{k,0} \\ 0 & \textit{if} \quad x \notin W_{k,0} \end{cases}.$$

Definition 5. *Given a set of m points $Y = \{(t_i, f_i); t_i \in [a,b], i = 1, ..., m\}$ and a fuzzy r-partition $(\mathbb{P},\mathbb{A},r)$ of $[a,b]$, the $(n+2r-2)$-vector of fuzzy numbers*

$$\mathbf{H}_{(\mathbb{P},\mathbb{A},r)} = (H_{2-r}, ..., H_{n+r-1}),$$

where each fuzzy interval H_k has α-cuts $W_{k,\alpha}$ given by (9) in Proposition 2, is called the discrete direct L_p-norm fuzzy-valued F-transform *with respect to* $(\mathbb{P}, \mathbb{A}, r)$*, based on the data-set* Y*. The corresponding* inverse L_p-norm fuzzy-valued iF-transform *is the fuzzy-valued function defined by*

$$f^{(p)}_{(\mathbb{P},\mathbb{A},r)}(x) = \frac{1}{r} \sum_{k=2-r}^{n+r-1} H_k A_k(x) \quad \textit{for } x \in [a, b]. \tag{10}$$

The α-cuts $W_{k,\alpha}$ of H_k will be denoted by

$$[H_k]_\alpha = \left[H^-_{k,\alpha}, H^+_{k,\alpha}\right], \quad k = 2-r, ..., n+r-1, \quad \alpha \in [0, 1]$$

and the α-cuts of the fuzzy-valued function $f^{(p)}_{(\mathbb{P},\mathbb{A},r)}(x)$, $x \in [a, b]$, are

$$\left[f^{(p)}_{(\mathbb{P},\mathbb{A},r)}(x)\right]_\alpha = \left[\frac{1}{r} \sum_{k=2-r}^{n+r-1} H^-_{k,\alpha} A_k(x) \ , \frac{1}{r} \sum_{k=2-r}^{n+r-1} H^+_{k,\alpha} A_k(x)\right], \quad \alpha \in [0, 1].$$

When $\alpha = 1$ we obtain the L_p-norm direct $F-$transform and inverse $iF-$transform function, corresponding to the core of the fuzzy-valued L_p-norm iF-transform.

3. Analysis of Bitcoin prices and Google trends by F-transform

To focus on the strength of fuzzy-valued L_p-norm F-transform smoothing, we will apply the proposed models to the time series of Bitcoin prices, that has received many attentions by regulators and investors in the last decade and because it deserves more in-depth analysis to capture its intrinsic nature.

Bitcoin was released at the beginning of 2009 as one of the first digital currencies in the market, it remained under \$0.20 for three years, it began to increase during the first quarter of 2013. By the end of 2017, Bitcoin was valued at nearly \$18,000 per "coin". In 2018, the price plummeted \$4000 and it grew again in 2019.

The second dataset we consider is Google Trends, the search index that measures what people are currently interested in and curious about. In particular, we consider Google Trends with value 100 out of 100 meaning that trend (word Bitcoin) is on its peak in the considered time period.

Here, we work on two daily time series, as in Fig. 1, from April 2013 to June 2019: Bitcoin prices (from www.blockchain.info) and Google Trends GT100 (from https://trends.google.com).

Figure 1. Daily Bitcoin prices (blue color, left scale) and daily GT100 series (red color, right scale) from 28.04.2013 to 17.06.2019.

Remark that the F-transform (direct and inverse) is linear with respect to the data set, in particular it is homogeneous and scale invariant: we can normalize the two time series and the direct F-transform components (or the iF-transform function) are multiplied by the same factor. In this way, we can compare the F-transform results for the two series in terms of the obtained smoothing effect and by visualizing the scatter plots of the of each series and the obtained iF-transform reconstructions.

The *degree of smoothness* of a given time series f_t, $t = 1, 2, ..., M$ is measured in terms of its (average) absolute variation, given by

$$V(f) = \frac{\sum\limits_{t=1}^{M-1} |f_{t+1} - f_t|}{M-1};$$

on the other hand, it is well known that the inverse L_p-norm F-transform function, for a fixed r-partition $(\mathbb{P}, \mathbb{A}, r)$, allows to compute the smoothing values

$$f_t^{(r)} = \frac{1}{r} \sum_{k=2-r}^{n+r-1} \vartheta_k(t) A_k(t) \qquad \text{for } t = 1, 2, ..., M$$

corresponding to the estimated (local) polynomials of order q, $\vartheta_k(t) = \theta_{k,0} + \theta_{k,1}(t - x_k) + ... + \theta_{k,q}(t - x_k)^q$, $k = 2 - r, ..., n + r - 1$. The corresponding absolute variation, given by

$$V(f^{(r)}) = \frac{\sum\limits_{t=1}^{M-1} |f_{t+1}^{(r)} - f_t^{(r)}|}{M - 1},$$

is in general smaller than $V(f)$; the ratio

$$L(f, f^{(r)}) = \frac{V(f^{(r)})}{V(f)}$$

represents the proportion of absolute variation which remains in the smoothed time series $f^{(r)}$ with respect to the original data f, while $1 - L(f, f^{(r)})$ is the amount of removed variation.

Table 1. **L_p-norm F-transform of BitCoin and GT100 time series**
For five values p and three orders $q \in \{0, 1, 3\}$ the table shows ratios $L(f, f^{(r)})$ with $f \in \{BitCoin, GT100\}$.

p	q	BitCoin	GT100
1	0	0.3526	0.1646
1	1	0.4298	0.2643
1	3	0.5813	0.4922
1.25	0	0.3493	0.1628
1.25	1	0.4315	0.2677
1.25	3	0.5693	0.4813
1.5	0	0.3465	0.1649
1.5	1	0.4282	0.2702
1.5	3	0.5558	0.4785
1.75	0	0.3448	0.1777
1.75	1	0.4250	0.2783
1.75	3	0.5483	0.4798
2	0	0.3428	0.1715
2	1	0.4221	0.2859
2	3	0.5432	0.4815

We have computed the L_p-norm F-transforms for different values of $p \in \{1, 1.25, 1.5, 1.75, 2\}$ and orders $q \in \{0, 1, 3\}$. The excellent performances of the smoothing based on F-transform are strongly confirmed for these two special time series, as summarized in Table 1 and Table 2.

All computations are performed with a 1-partition ($r = 1$), a decomposition $\mathbb{P}$ of $[1, M]$ into $n = 326$ equispaced nodes and basic functions obtained from $L(\tau) = \dfrac{\tau^2}{1 - 2\tau(1 - \tau)}$, $\tau \in [0, 1]$. For both series, covering the same time period, we have $M = 2276$ so that each subinterval $[x_k, x_{k+1}]$, $k = 1, ..., n - 1$ of $\mathbb{P}$ has exactly 8 observations (13 observations internal to intervals $]x_{k-1}, x_{k+1}[$, $k = 2, ..., n - 1$ on which basic function A_k is non-zero); remark that the two series are observed all days of the week, including holidays, so that all internal nodes x_k of $\mathbb{P}$ correspond to the same day in the week.

For the results of Table 1, the observed values are normalized in the range $[0, 1000]$; on this common range, the computed absolute variations are $V(BitCoin) = 4.6354$ and $V(GT100) = 6.9986$ and BitCoin is 66.23% less *fluctuating* (on average) than GT100. For all values of p, the reduction in total variation expressed by the ratios $L(f, f^{(r)})$, for both series, is more depending on the order q than on the used norm L_p; this is not surprising, because increasing the degree of local polynomials will reduce the average fitting errors but increase their variation.

In Table 2 the smoothing and fitting of the time series by L_p-norm F-transforms are compared in terms of three well known indices: the mean square error MSE, the mean absolute percentage error $\%MAE$ and the Kendall τ rank correlation.

Table 2. Approximation indices for L_p-norm F-transform of BitCoin and GT100
For five values $p \in \{1, 1.25, 1.5, 1.75, 2\}$ and three orders $q \in \{0, 1, 3\}$ the table shows indices MSE, $\%MAE$, and Kendall τ correlation for the pairs $(f, f^{(r)})$ with $f \in \{BitCoin, GT100\}$.

			BitCoin			GT100	
p	q	MSE	$\%MAE$	τ	MSE	$\%MAE$	τ
1	0	363.98	2.5173	0.9766	2.7670	7.8219	0.8897
1	1	201.66	2.0152	0.9815	1.8610	6.7139	0.9049
1	3	148.75	1.4739	0.9860	1.5614	4.8293	0.9331
1.25	0	219.72	2.5171	0.9768	2.3939	7.9307	0.8911
1.25	1	191.58	2.0315	0.9817	1.8211	6.8342	0.9074
1.25	3	139.34	1.4874	0.9863	1.4556	4.9210	0.9345
1.5	0	217.61	2.5641	0.9765	2.3339	8.1374	0.8910
1.5	1	190.20	2.0805	0.9814	1.8027	7.0240	0.9068
1.5	3	137.72	1.5284	0.9861	1.4169	5.1172	0.9331
1.75	0	217.79	2.6232	0.9762	2.3020	8.3932	0.8896
1.75	1	191.26	2.1298	0.9810	1.7920	7.2505	0.9057
1.75	3	137.75	1.5734	0.9858	1.4031	5.3430	0.9311
2	0	219.41	2.6859	0.9757	2.2855	8.6858	0.8877
2	1	192.74	2.1778	0.9708	1.7964	7.4879	0.9038
2	3	138.46	1.6155	0.9856	1.4040	5.5762	0.9289

Here the series are not normalized (in particular, the value of index MSE depends on the scale of the series). In all cases, the F-transform fitting for Bitcoin series has significantly smaller errors than for GT100, as is evidenced by indices $\%MAE$ and τ (the Kendall τ is always significantly positive with p-value less than 10^{-8}).

For the fitting F-transform functions obtained by L_p-norm with $p = 1.5$ and $q = 0$, the scatter plots of time series (f_t, f_t^r) are pictured in Figure 2; Figure 3 plots f_t and f_t^r with respect to time.

Figure 2. Scatter plots of (f_t, f_t^r) for daily Bitcoin prices (left picture) and daily GT100 trends (right), from april 2013 to june 2019.

Figure 3. L_p-based smoothing for BitCoin (top picture, blue points) and GT100 (bottom, blue points) series, obtained with $p = 1.5$ and $q = 0$. The green curves plot the observed series.

The research question we want now to investigate is the following: the higher is GT100 about the word Bitcoin, the higher Bitcoin returns can be expected?

First of all, a scatter plot of the pairs is pictured in Figure 4 (the two time series are normalized in the range $[0, 1000]$ and GT100 appears in horizontal axis); observations in our data-set cover the cited time period 2013-2019, are concentrated on the bottom-left part of the positive quadrant and have rare points with big values in both series. At a first look, no evident functional relationships emerge from the data; they simply show a tendency to be co-monotonic, but the points are very sparse.

Figure 4. Scatterplot $(GT100_t, BitCoin_t)$ of daily Bitcoin prices vs daily GT100 series. For this visualization, the two time series are normalized to the common range $[0, 1000]$.

For a deeper analysis, we propose a model based on F-transform, relating the pairs of data $(GT100_t, BitCoin_t)$. F-transform is thus used to model Bitcoin as a function of GT100. The L_1-norm and L_2-norm based inverse iF-transforms of the data-set $(GT100_t, BitCoin_t), t = 1, 2, ..., M$ are computed; taking into account the sparsity of the values $GT100_t$ (in particular above the threshold 40), we use a non-uniform 1-partition $(\mathbb{P}, \mathbb{A})$ of the range $[0, 100]$ of the observed $GT100_t$, namely the set of 25 nodes $\{0, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 35, 40, 45, 50, 60, 100\}$, as pictured in Figure 5. The two curves give the predominant relationship between $BitCoin$ and $GT100$. It is evident that both iF-transforms of $BitCoin_t$ are not increasing on the whole range of $GT100$ (in particular, they decrease when GT100 is around $7 - 8$, around 20 and around 40). On the other hand, observing the dispersion of points in Figure 4, both iF-transforms are not good fitting of the complete data set and it appears that the points can be "clustered", e.g., for different levels of Bitcoin prices, and better sub-fittings can be attempted.

Figure 5. The figure shows (inverse) L_1-norm (on top) and L_2-norm (on bottom) iF-transform functions obtained for the observations $(GT100_t, BitCoin_t)$; here, the daily Bitcoin prices are considered as functions in the domain $[0, 100]$ of GT100.

This can be better analysed by applying the quantile and the expectile F-transforms to our data set $(GT100_t, BitCoin_t)$, pictured in Figure 6, which shows the fuzzy-valued expectile F-transform of Bitcoin as a function of GT100 for five different α-cuts corresponding to $\alpha \in \{0.01, 0.25, 0.5, 0.75, 1\}$ (i.e. ten values $\omega \in \{\frac{\alpha}{2}, 1 - \frac{\alpha}{2}; all\ \alpha\}$ for the asymmetry expectile parameter ω).

Figure 6. Fuzzy-valued Quantile (top picture) and Espectile (bottom picture) F-transform functions of the data set $(BitCoin_t, GT100_t)$; the five computed α-cuts correspond to $\alpha \in \{0.01, 0.25, 0.5, 0.75, 1\}$.

We see that, corresponding to different values of α, i.e., corresponding to subintervals in the range of Bitcoin prices, the relationship between our time series changes significantly.

This suggests that, possibly, the clustering of the data into subsets may significantly improve the quality of fitting.

3.1. F-transform of clustered subsets of observed data

Clearly, there are several procedures and criteria to cluster the observed data $(GT100_t, BitCoin_t)$; we use the well-known *k-means* method and the number of clusters is selected according to the silhouette measures available in MATLAB R2018b. We have performed three types of clusters, the first on the basis of variable $BitBoin$, the second using the pair $(BitCoin, GT100)$ and the third using observations $(BitCoin, GT100, \Delta BitCoin, \Delta GT100)$ where Δ is the first difference operator $\Delta f_t = f_t - f_{t-1}$. The clustering measure is the standard euclidean distance.

Let nCl denote the number of clusters and let $j = 1, 2, ..., nCl$ be the labels of cluster S_j. Each observation $(BitCoin_t, GT100_t)$ is assigned to cluster $S_{c(t)}$, i.e., the observation t is assigned to cluster labelled $c(t) \in \{1, 2, ..., nCl\}$.

For a given clustering, identified by clusters $S_1, S_2, ..., S_{nCl}$, the L_2-norm based F-transform is applied (independently) on each subset of data, for $j = 1, 2, ..., nCl$,

$$S_j = \{(BitCoin_t, GT100_t) | c(t) = j\} .$$

Finally, for the observations of each cluster, the inverse iF-transform is computed and the fitted values, for each cluster, are obtained and recomposed to obtain the fitted values for the whole data set.

If all the data are collected in a unique cluster and the F-transform is applied to the whole data set, we obtain the fitted $BitCoin$ series pictured in Figure 7: the green points give the observed $BitCoin_t$, the red points are the $GT100_t$ series and in blue is the fitting of $Bitcoin$. We see that the fitting preserves the qualitative (gross) form of the observed Bitcoin prices, but in several portions of time period the fitting is not good.

Figure 7. The fitted $BitCoin$ time series (blue color) is obtained from L_2-norm iF-transform function applied to the data set $(GT100_t, BitCoin_t)$; the observed $BitCoin_t$ values are green and the observed $GT100$ values are red.

Significant improvements are obtained by adopting the three described pre-clustering, denoted respectively by labels A, B and C. *The computations are performed on the second half portion of the time-period, starting with observation time* t_{1121}*; the first part is less interesting because, from observation* t_{350} *to* t_{1220} *both time series have small variations and relatively flat curves. Without performing pre-clustering, the* L_2*-norm F-transform reconstruction (of order 1) of* $BitCoin$ *in terms of* $GT100$ *has Kendall correlation* $\tau = 0.6049$ *and Spearman correlation* $\rho = 0.7830$. We will compare τ and ρ indices as preliminary evaluation of the effect of pre-clustering on the fitting quality.

Clustering A. Clusters are based on variable $BitCoin$: the number of clusters is $nCl = 20$. The L_2-norm F-transform reconstruction (of order 1) of $BitCoin$ in terms of $GT100$ with pre-clustering A has much higher Kendall correlation $\tau = 0.9457$ and Spearman correlation $\rho = 0.9956$.

Figure 8. Clustering A: the fitted $BitCoin$ time series (blue color) is obtained from L_2-norm iF-transform function applied to the data set $(GT100_t, BitCoin_t)$; the observed $BitCoin_t$ values are green and the observed $GT100$ values are red.

Figure 9. The blue points correspond to the observations $(GT100_t, BitCoin_t)$ assigned to each cluster by clustering A. The red points give the centroid values of variable $BitCoin$.

Figure 10. Clustering A: for each cluster, the fitted $BitCoin$ subseries (blue colours) are obtained from L_2-norm iF-transform function applied to the subset of data assigned to each cluster; the observed $(GT100_t, BitCoin_t)$ values are green.

In Figure 8 we plot the observed and fitted Bitcoin series for the second half of observations, with evidence that clustering A allows a much better fitting. The 20 clusters are pictured (blue colours) in Figure 9 and expanded in 10 where also the sub-fittings are visible.

Clustering B. Clusters are based on both variables $(BitCoin, GT100)$: the number of clusters is $nCl = 21$. The L_2-norm F-transform reconstruction (of order 1) of $BitCoin$ in terms of $GT100$ with pre-clustering B has high Kendall correlation $\tau = 0.9447$ and Spearman correlation $\rho = 0.9954$, similar to clustering A.

Figure 11. Clustering B: the fitted $BitCoin$ time series (blue color) is obtained from L_2-norm iF-transform function applied to the data set $(GT100_t, BitCoin_t)$; the observed $BitCoin_t$ values are green and the observed $GT100$ values are red.

Figure 12. The blue points correspond to the observations $(GT100_t, BitCoin_t)$ assigned to each cluster by clustering B. The red points give the centroid values of variables $(GT100, BitCoin$.

Figure 13. Clustering B: the fitted $BitCoin$ time series is in blue color; the observed $BitCoin_t$ values are green.

In Figure 11 we plot the observed and fitted Bitcoin series for the second half of observations, with evidence that clustering B allows a good fitting. The 21 clusters are pictured (blue colours) in Figure 12 and expanded in 13 where also the sub-fittings are visible.

Clustering C. Clusters are based on variables $(BitCoin, GT100, \Delta BitCoin, \Delta GT100)$; the number of clusters is $nCl = 24$. The L_2-norm F-transform reconstruction (of order 1) of $BitCoin$ in terms of $GT100$ with this pre-clustering has high Kendall correlation $\tau = 0.9183$ and Spearman correlation $\rho = 0.9905$, similar but not better than pre-clustering A and B.

Figure 14. Clustering C: the fitted $BitCoin$ time series (blue color) is obtained from L_2-norm iF-transform function applied to the data set $(GT100_t, BitCoin_t)$; the observed $BitCoin_t$ values are green and the observed $GT100$ values are red.

Figure 15. The blue points correspond to the observations $(GT100_t, BitCoin_t)$ assigned to each cluster by clustering C. The red points give the centroid values relative to two variables $BitCoin$ and $BitCoin$.

Figure 16. The fitted $BitCoin$ time series (blue color) is obtained from L_2-norm iF-transform function applied to the data set $(GT100_t, BitCoin_t)$; the observed $BitCoin_t$ values are green and the observed $GT100$ values are red.

In Figure 14 we plot the observed and fitted Bitcoin series for the second half of observations, with evidence that also clustering C allows a good fitting. The 24 clusters are pictured (blue colours) in Figure 15 and expanded in 16 where also the sub-fittings are pictured.

The overall result is that pre-clustering of the data, even based on very simple clustering strategies and a relatively small number of clusters (from 20 to 24) significantly improves the fitting ability of F-transform.

It is also interesting to see that the form of relationships between $BitCoin$ and $GT100$ is very different for each cluster; this has important consequences on the analysis and modelling of Bitcoin time series as, in particular, it follows different paths in various sub-periods of time and in cases of *rare* values of the data (e.g., big values and/or big absolute changes).

4. Stylized facts identified by F-transform components

In this section we will analyse the local F-transform components, in particular the form of polynomials φ_k of L_2-norm F-transform, of orders $q = 1$ and $q = 2$ described in section 2.1, applied to the Bitcoin time series. We have selected the number m of daily observations such that $m - 1$ is a multiple of 7 (Bitcoin and GT100 are observed all the days in the year): in this way, the available data are $m = 2276$.

Consider an r-partition $(\mathbb{P}, \mathbb{A}, r)$ with nodes $x_k, k = 1, 2, ..., n$; denoting the time points of observations simply by $t = 1, 2, ..., m$ (or $t_j = j$ for $j = 1, ..., m$) we consider two uniform $(\mathbb{P}, \mathbb{A}, r)$:

(P_a) - a dense partition with $n = m$ and $x_k = k$ (i.e., each observation is a node) and the bandwidth r is chosen such that each open interval $I_k^{(r)} =]x_{k-r}, x_{k+r}[$ contains (internally) a prescribed sufficient number of data. E.g., with $r = 4$ each $I_k^{(r)}$ contains all data of the week centred at x_k; with $r = 7$, $I_k^{(r)}$ contains the two weeks ending and starting with x_k.

(P_b) - a sparse partition with $n = 326$ and $x_k = 1 + 7(k - 1)$ (i.e., there is a node every 7 observations) and the bandwidth $r = 3$ is chosen. The (direct) F-transform components will span 21 observed values on each side (left or right) of the nodes.

4.1. F-transform fitting with dense r-partition

In the dense partition case, we obtain the best L_2-norm F-transform components $\varphi_k(t)$ associated to all observations (indeed, $x_k = k$ corresponds to all observed times $k = 1, ..., n = m$); in this way, we are able to estimate the local trend around every observation and se can follow the time evolution of trends by plotting $\varphi_k(t)$ around x_k on subintervals $I_k^{(r)}$ (see Figures 17 and 18).

On the other hand, if we translate vertically the polynomials $\varphi_k(t)$, the polynomials $\widehat{\varphi}_k(t) = \varphi_k(t) - \phi_{k,0}$ are such that $\widehat{\varphi}_k(x_k) = 0$ for all k and, if $q > 0$, we can cluster the $\widehat{\varphi}_k$ by clustering the set of vectors $(\phi_{1,k}, ..., \phi_{q,k})$ of the estimated coefficients. If $q = 1$ we obtain a set of lines through the origin with different slopes (in terms of a single *variable* $\phi_{1,k}$); if $q = 2$ we obtain a set pf parabolic functions through the origin, in terms of two variables $\phi_{1,k}$ and $\phi_{2,k}$.

Figure 17. L_2-norm F-transform components of order 1 for Bitcoin series. The fuzzy r-partition is P_a with $r = 4$ (top picture) and $r = 7$ (bottom).

Figure 18. L_2-norm F-transform components of order 2 for Bitcoin series. The fuzzy r-partition is P_a with $r = 4$ (top picture) and $r = 7$ (bottom).

Using the k-means clustering method with the Euclidean distance and testing the number of clusters using the silouhette values, we have that the best number of clusters is $nCl = 9$ when $q = 1$ and $nCl = 15$ when $q = 2$.

The interpretation of $nCl = 9$ clusters (characterised by variable $\phi_{1,k}$) is interesting, because we have a central cluster of local trends with slope around 0 and other eight clusters characterized by slopes ranging from very negative values (cluster 1) to intermediate negative values (cluster 3) up to intermediate positive (cluster 7) to very positive slopes (cluster 9).

Analogous interpretation is applied to the case of $nCl = 15$ clusters (and $q = 2$), where again cluster 1 corresponds to the most negative slope, cluster 8 to almost zero slope and cluster 15 to the most positive slope; clearly, also the degrees of concavity and convexity (represented by the second variable $\phi_{2,k}$) are taken into account in this case.

Figure 19 pictures, for each cluster $S_c^{(1)}$, $c = 1, 2, ..., 9$, the first order shifted polynomials $\widehat{\varphi}_k(t) = \phi_{1,k}(t - x_k)$ assigned to $S_c^{(1)}$ (red colours) and the centroid polynomial (blue colour) obtained by averaging the parameters $\phi_{1,k}$.

Similarly, Figure 20 pictures, for each cluster $S_c^{(2)}$, $c = 1, 2, ..., 15$, the second order shifted polynomials $\phi_{1,k}(t - x_k) + \phi_{2,k}(t - x_k)^2$ assigned to $S_c^{(2)}$ (red colours) and the centroid polynomial (blue colour) obtained by averaging the pairs of parameters $(\phi_{1,k}, \phi_{2,k})$.

As we have said, the nCl centroid polynomials, identified by averaging the parameters of all elements assigned to each cluster $S_c^{(1)}$ or $S_c^{(2)}$, can be considered as the *stylized forms* of the local trends. If we identify each estimated trend by the centroid of its cluster, we then have nCl stylized forms, one for each cluster, that form the possible typical trends around the observed points of the time series.

Figure 19. Clustering of L_2-norm F-transform components (polynomials) of order 1 for Bitcoin series. The clusters correspond to the fuzzy r-partition P_a with $r = 4$ (bottom picture) and $r = 7$ (top).

Figure 20. Clustering of L_2-norm F-transform components (polynomials) of order 2 for Bitcoin series. The clusters correspond to the fuzzy r-partition P_a with $r = 4$ (bottom picture) and $r = 7$ (top).

As a last step, we can produce a simple analysis of how good the stylized trends represent the effective observations: let's denote by $f^q_{(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ the standard inverse iF-transform values at times t_j obtained with estimated direct F-transform (polynomial) components $\varphi_k(t)$, i.e.,

$$f^q_{(\mathbb{P}_a,\mathbb{A},r)}(t) = \frac{1}{r}\sum_{k=2-r}^{n+r-1} \varphi_k(t)A_k(t); \tag{11}$$

denote by $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ the analogous expression obtained by substituting each local polynomial $\varphi_k(t)$ by $\widehat{\varphi}_k(t) = \phi_{0,k} + \widehat{\phi}_{1,k}(t-x_k) + ... + \widehat{\varphi}_{q,k}(t-x_k)^q$ (here $q = 1$ or $q = 2$), where the parameters $\widehat{\phi}_{1,k}$ or the pairs $(\widehat{\phi}_{1,k}, \widehat{\phi}_{2,k})$ are the ones that identify the centroid of cluster containing time k, i.e., if an observation belongs to cluster S_c we substitute the computed local trend with the local trend of the corresponding centroid:

$$f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t) = \frac{1}{r}\sum_{k=2-r}^{n+r-1} \widehat{\varphi}_k(t)A_k(t). \tag{12}$$

Essentially, we identify the elements of each cluster by its centroid and we estimate its goodness in terms of the vicinity between the modified version $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t)$ at times $t_j = 1, ..., M$ and the observed data $f_j = BitCoin_j$. In Figures 21 and 22 the data f_j (green colours) and $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ (black colours) are plotted for all the data with $r = 4$ and $r = 7$, respectively.

Scatter plots of the same values for the cases $r = 4$, $r = 7$ are pictured in Figures 23 and 24; remark in particular that the iF-transform values and the modified values have a very high correlation and the two values are very near to each-other on the whole range of small and big values of observed prices.

Figure 21. L_2-norm modified iF-transform values $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ (top:$q = 1, bottom$:q=2) for Bitcoin series. The fuzzy r-partition is P_a with $r = 4$.

Figure 22. L_2-norm modified iF-transform values $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ (top:$q = 1, bottom$:q=2) for Bitcoin series. The fuzzy r-partition is P_a with $r = 7$.

Figure 23. Pairwise scatter-plots of values f_j, $f^q_{(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ and $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ using L_2-norm F-transform smoothing of order 1 (top) and order 2 (bottom) for Bitcoin series. The fuzzy r-partition is P_a with $r = 4$.

Figure 24. Pairwise scatter-plots of values f_j, $f^q_{(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ and $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ using L_2-norm F-transform smoothing of order 1 (top) and order 2 (bottom) for Bitcoin series. The fuzzy r-partition is P_a with $r = 7$.

Finally, it is interesting to observe the time evolution of the different clusters (see Figure 25).

Figure 25. Time-evolution of clusters obtained with the L_2-norm F-transform smoothing of order 1 for Bitcoin series. The fuzzy r-partition is P_a with $r = 4$.

Remark that in the first part of the time series, the data (i.e., the local trends) persist into the quasi-zero slope (quasi-constant time series); after observation 1400, frequent changes in the local trends appear evident, but changes seem to be gradual from one class to a near one and only rarely the local trends jump from a form to a very different one. This appears clearly from the transition matrix $P = [prob(i,j)]$, $i,j = 1,2,...,9$ given below ($prob(i,j)$ is the probability that local trend of cluster i moves to class j): we see that matrix P is essentially tridiagonal.

$$P = \begin{array}{c|c|c|c|c|c|c|c|c|c|}
 & \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & \mathit{5} & \mathit{6} & \mathit{7} & \mathit{8} & \mathit{9} \\ \hline
\mathit{1} & \mathbf{0.57} & 0.29 & 0.14 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
\mathit{2} & 0.08 & \mathbf{0.44} & 0.44 & 0 & 0.04 & 0 & 0 & 0 & 0 \\ \hline
\mathit{3} & 0.02 & 0.12 & \mathbf{0.39} & 0.25 & 0.12 & 0.03 & 0.05 & 0.02 & 0 \\ \hline
\mathit{4} & 0 & 0.02 & 0.10 & \mathbf{0.50} & 0.31 & 0.04 & 0.01 & 0.01 & 0 \\ \hline
\mathit{5} & 0 & 0 & 0 & 0.03 & \mathbf{0.93} & 0.03 & 0 & 0 & 0 \\ \hline
\mathit{6} & 0 & 0 & 0.1 & 0.04 & 0.32 & \mathbf{0.53} & 0.10 & 0.01 & 0 \\ \hline
\mathit{7} & 0 & 0 & 0 & 0.03 & 0.04 & 0.27 & \mathbf{0.56} & 0.10 & 0 \\ \hline
\mathit{8} & 0 & 0.05 & 0.09 & 0.05 & 0 & 0.09 & 0.18 & \mathbf{0.50} & 0.05 \\ \hline
\mathit{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.50 & \mathbf{0.50} \\ \hline
\end{array}$$

4.2. F-transform fitting with sparse r-partition

The results obtained with the dense r-partition $\mathbb{P}_a$ are confirmed when using the sparse r-partition $\mathbb{P}_b$. The computations are performed only for the bandwidth $r = 3$, corresponding to $n = 326$ nodes of $\mathbb{P}_b$ and $42 = 2 * 7 * 3$ data belonging to each interval $[x_{k-r}, xk + r]$.

Figures 26 and 27 plot the local trends of orders $q = 1$ and $q = 2$, respectively (We have plotted only the second part of the Bitcoin time series).

Figure 26. L_2-norm F-transform components of order 1 for Bitcoin series. The fuzzy r-partition is P_b with $r = 3$.

Figure 27. L_2-norm F-transform components of order 2 for Bitcoin series. The fuzzy r-partition is P_b with $r = 3$.

The local trends are clustered into $nCl = 9$ groups when $q = 1$ and $nCl = 15$ groups when $q = 2$, pictured in Figures 28 and 29. It appears that , for $q = 1$, clusters 8 and 9 (and clusters 1, 13, 14, 15, when $q = 2$) contain very few elements and possibly, in this sparse case, the number of clusters should be reduced to 7 and 11.

Figure 28. Clustering of L_2-norm F-transform components (polynomials) of order 1 for Bitcoin series. The fuzzy r-partition is P_b with $r = 3$.

Figure 29. Clustering of L_2-norm F-transform components (polynomials) of order 2 for Bitcoin series. The fuzzy r-partition is P_b with $r = 3$.

The substitution of estimated local trends with the ones obtained from centroids of each cluster, produces the smooth reconstructions represented in figures 30 and 31, respectively; the scatter plots of the data and their smoothing with standard F-transform and the modified versions, are plotted in Figures 32 and 33. We remark that, corresponding to a stronger smoothing effect obtained with a smaller number of nodes in the decomposition (now it is sparse) the quality of the fitting is reduced.

Figure 30. L_2-norm modified iF-transform values $f^{q}_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ for Bitcoin series. Here, $q = 1$ and the fuzzy r-partition is P_b with $r = 3$.

Figure 31. L_2-norm modified iF-transform values $f^q_{c,(\mathbb{P}_a,\mathbb{A},r)}(t_j)$ for Bitcoin series. Here, $q = 2$ and the fuzzy r-partition is P_b with $r = 3$.

Figure 32. L_2-norm F-transform components (polynomial of orders 1 and 2) for Bitcoin series. The fuzzy r-partition is P_b with $r = 3$.

Figure 33. L_2-norm F-transform components (polynomial of orders 1 and 2) for Bitcoin series. The fuzzy r-partition is P_b with $r = 3$.

5. Forecasting Bitcoin Prices with GT100 index

An attempt for forecasting is based on a local polynomial (of other functional form) of F-transform components, used to model the Bitcoin Prices, such as

$$\vartheta_k(t) = \theta_{k,0} + \theta_{k,1}(t - x_k) + ... + \theta_{k,q}(t - x_k)^q + \theta_{k,q+1}g_1(t) + \theta_{k,q+2}g_2(t) + ... + \theta_{k,q+s}g_s(t) \qquad (13)$$

where $\vartheta_k, k = ..., N-2, N-1, N$ is the k-th local trend function (k-th direct F-transform component) and $g_1(t), ..., g_s(t)$ are delayed versions of the Google Trends $GT100(t)$ and/or $BitCoin(t)$ series.

Suppose we are interested to a forecasting model by which, having available observations of $BitCoin(t)$ and $GT100(t)$ at times $t = ..., T-2, T-1, T$ up to time T, we like to construct a forecast of $BitCoin(T+l)$ for l steps ahead. To do this, we estimate the direct F-transform components (13) with appropriate q, s and values $g_1(t-l), ..., g_s(t-l)$, obtained from observed values of $BitCoin(t-l)$ and/or $GT100(t-l)$ for $t = ..., T-2, T-1, T$; then, e.g., using the last estimated trend function $\vartheta_N(t)$ we approximate $BitCoin(T+l)$ with $\vartheta_N(T+l)$. This is always possible if the fuzzy r-partition is such that $T + l \in]T = x_N, x_{N+r}[$. Alternatively, if $r > 1$, we can approximate $BitCoin(T+l)$ by computing the inverse F-transform at time $T+l$ as the combination of local trends $\vartheta_k(T+l)$ that have positive weights $A_k(T+l)$, the basic functions active at time $T+l$.

Clearly, this construction may be good and reasonable only for short-term forecasting. In our experiments we have used the first approximation only to forecast Bitcoin prices $BitCoin(T+l)$ with $l \in \{1,2,3,4,5\}$. The reported results are given for the last 1200 values of the available time series. We have used $q \in \{0,1,2\}$, $r \in \{1,2\}$ and two cases of functions $g_j(t)$:

Model A: $s = 1$ and $g_1(t) = GT100(t)$;

Model B: $s = 2$ and $g_1(t) = GT100(t)$, $g_2(t) = BitCoin(t-1)$, i.e., by adding an autoregressive term.

Our simple forecasting model is obtained by the following three steps:

Step 1. We start with one of the fitting model obtained in the analysis in previous sections and we chose the pair of values (r, m_r) where $r \geq 1$ is the bandwidth of the r-partition and the associated value m_r is the number of time observations used to estimate the parameters of local trend functions $\vartheta_k(t)$. We have found that m_r observations on each subinterval of the partition, in the range $m_r = 11, ..., 25$, produces in general the best fitting results: two or three weeks of data are sufficient to obtain the forecast.

Step 2. The parameters in $\vartheta_k(t)$ are estimated using the L$_p$-norm based criterion: we assume $p = 1.5$ as a good intermediate value between quantile ($p = 1$) and expectile ($p = 2$) estimators.

Step 3. Each l-steps ahead forecast value f_{T+l} for the last more recent 1200 available observations ending at time $T_{final} = 2242$ is obtained from $\vartheta_N(t)$, where $N = 2 + r$ is the number on intervals in the partition covering m_r data that terminate at time T: forecast f_{T+l} is then estimated from data $BitCoin_t, t = T - m_r + 1, ..., T$, $GT100_t, t = T - m_r - l + 1, ..., T - l$ for $g_1(t-l)$ and, in Model B, $BitCoin_t, t = T - m_r - l + 1, ..., T - l$ for $g_1(t-l)$. In this way, we can compute $f_{T+l} = \vartheta_N(T+l)$ as the needed values $g_1(T+l) = GT100_T$ and $g_2(T+l) = BitCoin_T$ are available from observations.

For the fitting (i.e., with $l = 0$) and forecasting models (with $l > 0$) we report the mean square error ($MSE = \frac{1}{m}\sum_{i=1}^{m}(f_i - f_i^r)^2$), the mean absolute percentage error ($\%MAE = \frac{1}{m}\sum_{i=1}^{m}\frac{\left|f_i - f_i^r\right|}{f_i}$) and the well known Kendall τ rank correlation (a measure of ordinal association between f_i and f_i^r).

Table 3. Approximation indices for L_p-norm F-transform of BitCoin and GT100
For five values $p \in \{1, 1.25, 1.5, 1.75, 2\}$ and three orders $q \in \{0, 1, 3\}$ the table shows indices MSE, $\%MAE$, and Kendall τ correlation for the pairs $(f, f^{(r)})$ with $f \in \{BitCoin, GT100\}$.

			BitCoin			GT100	
p	q	MSE	$\%MAE$	τ	MSE	$\%MAE$	τ
1	0	363.98	2.5173	0.9766	2.7670	7.8219	0.8897
1	1	201.66	2.0152	0.9815	1.8610	6.7139	0.9049
1	3	148.75	1.4739	0.9860	1.5614	4.8293	0.9331
1.25	0	219.72	2.5171	0.9768	2.3939	7.9307	0.8911
1.25	1	191.58	2.0315	0.9817	1.8211	6.8342	0.9074
1.25	3	139.34	1.4874	0.9863	1.4556	4.9210	0.9345
1.5	0	217.61	2.5641	0.9765	2.3339	8.1374	0.8910
1.5	1	190.20	2.0805	0.9814	1.8027	7.0240	0.9068
1.5	3	137.72	1.5284	0.9861	1.4169	5.1172	0.9331
1.75	0	217.79	2.6232	0.9762	2.3020	8.3932	0.8896
1.75	1	191.26	2.1298	0.9810	1.7920	7.2505	0.9057
1.75	3	137.75	1.5734	0.9858	1.4031	5.3430	0.9311
2	0	219.41	2.6859	0.9757	2.2855	8.6858	0.8877
2	1	192.74	2.1778	0.9708	1.7964	7.4879	0.9038
2	3	138.46	1.6155	0.9856	1.4040	5.5762	0.9289

We see from Table 3 that the fitting of Bitcoin and GT100 time series become significantly better by increasing the order q; it is also interesting to remark that GT100 fitting is much less precise that Bitcoin fitting (with the same p and q). On the other hand, polynomials of orders $q > 1$ are not useful for extrapolation as they tend to be highly oscillating for a lag $l > 1$.
For these reasons, only three pairs (p, q) as in Tables 4 and 5 are considered and we see that, for forecasting, $q = 1$ gives good results for small lag l while $q = 0$ is better for higher lags.

Table 4. Model A: L_p-norm Fitting and Forecasting results
For $p = 1.5$, the table shows indices $\%MAE$ and Kendall τ correlation for the fitting (columns 4,5 with label fit) and the forecasting (columns 6,7 with label for) corresponding to five lags $l \in \{1, 2, 3, 4, 5\}$ and three pairs (q, r) of order q and bandwidth r.

q	r	l	$\%MAE$-fit	τ-fit	$\%MAE$-for	τ-for
0	1	1	1.65	0.98	3.30	0.96
		2	1.67	0.98	4.41	0.95
		3	1.67	0.98	5.35	0.94
		4	1.58	0.98	6.17	0.93
		5	1.58	0.98	6.88	0.92
1	1	1	0.62	0.99	3.15	0.96
		2	0.59	0.99	5.08	0.94
		3	0.60	0.99	6.59	0.93
		4	0.60	0.99	8.09	0.91
		5	0.60	0.99	10.13	0.89
0	2	1	2.24	0.97	3.70	0.95
		2	2.28	0.97	4.77	0.94
		3	2.31	0.97	5.65	0.93
		4	2.22	0.97	6.43	0.92
		5	2.21	0.97	7.11	0.92

Table 5. Model B: L_p-norm Fitting and Forecasting results
For $p = 1.5$, the table shows indices $\%MAE$ and Kendall τ correlation for the fitting (columns 4,5 with label fit) and the forecasting (columns 6,7 with label for) corresponding to five lags $l \in \{1,2,3,4,5\}$ and three pairs of order q and bandwidth r.

q	r	l	$\%MAE$-fit	τ-fit	$\%MAE$-for	τ-for
0	1	1	1.56	0.98	3.21	0.96
		2	1.57	0.98	4.33	0.95
		3	1.56	0.98	5.31	0.93
		4	1.57	0.98	6.12	0.93
		5	1.57	0.98	6.83	0.92
1	1	1	1.17	0.98	3.16	0.96
		2	1.20	0.98	4.54	0.94
		3	1.17	0.98	5.84	0.93
		4	1.16	0.98	6.98	0.92
		5	1.18	0.98	7.89	0.91
0	2	1	1.56	0.98	3.21	0.96
		2	1.57	0.98	4.34	0.95
		3	1.56	0.98	5.31	0.93
		4	1.57	0.98	6.12	0.93
		5	1.57	0.98	6.83	0.92

The forecast Bitcoin time series for lags $l = 1,2$ obtained with model A are pictured in Figure 34; for lags $l = 4,5$ and model B are plotted in Figure 35; Figure 36 plots the pairs $(fSer_t, fFor_t)$ of (observed, forecast) time series for lags $l = 1,2,4,5$.

Figure 34. Model A: L_p-norm F-transform forecast (denoted as fFor) for the last 1200 days of available data for Bitcoin series (observed time series id senoted as fSer). Here, $p = 1.5$, $q = 1$ and two lags $l = 1$ (top two pictures), $l = 2$ (bottom). The percent errors $\frac{fSer_t - fFor_t}{fSer_t}$ are also plotted.

Figure 35. Model B: L_p-norm F-transform forecast (denoted as fFor) for the last 1200 days of available data for Bitcoin series (observed time series id senoted as fSer). Here, $p = 1.5$, $q = 1$ and two lags $l = 4$ (top two pictures), $l = 5$ (bottom). The percent errors $\frac{fSer_t - fFor_t}{fSer_t}$ are also plotted.

Figure 36. L_2-norm F-transform components (polynomial of orders 1 and 2) for Bitcoin series. The fuzzy r-partition is P_a with $r = 7$.

To conclude this section on forecasting Bitcoin time series, we shortly explore the statistical significance of the proposed models A and B. Following some ideas in [25], we compare our forecast estimates with the so called random walk model, assumed as a benchmark for single step forecasting, i.e., with lag $l = 1$. Defining the return series as $fRet_t = log(\frac{fSer_t}{fSer_{t-1}})$, the random walk forecast $\widehat{r}_T$ of the returns at time T is defined in terms of a fixed time horizon of s observations $fRet_t$, $t = T - s + 1, ..., T$ ending at time T by

$$Ret_t = \widehat{r}_T + \varepsilon_t, \qquad \varepsilon_t \sim ID(0, \sigma_T), \qquad t = T - s + 1, ..., T.$$

Then, considering that the return values $fRet_t$ are known up to time T, the random walk forecast of $fRet$ at time $T+1$ is simply the average $\widehat{r}_T = \frac{1}{s} \sum_{t=T-s+1}^{T} Ret_t$, we can estimate the random walk forecast of $fSer_{T+1}$ from the definition of return at time $T+1$ and obtain $f^{rw}_{T+1} = exp(\widehat{r}_T)fSer_T$. This calculations are performed for T being each of the last 730 available observations (two years).

As in [25], the Diebold-Mariano (DM) test statistics is applied to test the significance of the MSE measures for our forecasts (with models A and B) in comparison with the random walk forecast. The results are reported in Table 6.

Table 6. Diebold-Mariano test statistics for L_p-norm forecasting results
The table shows indexes $sMSE = \sqrt{MSE}$ and Kendall rank correlation τ, corresponding to lag $l = 1$ forecasts for Models A and B, obtained with $p = 1.5$, $q = 1$ and bandwidth $r = 1$.

	$sMSE$	DM-stat	P-value	τ
Model A	488.2	-2.9468	0.0016	0.9265
Model B	487.0	-3.8301	0.000064	0.9186
Random Walk	585.1	-	-	0.9078

Table 6 (in particular the small P-value) confirm that Bitcoin prices can be forecasted using F-transform of order $q = 1$, as indeed both (simple) short term models A and B outperform the random walk model.

6. Final comments and conclusions

In this paper we develop a sentiment analysis model in order to evaluate the relationship between Google Trends and Bitcoin prices. Thanks to the high flexibility of smoothing techniques and forecasting models based on Fuzzy transform we show that the great interest in Bitcoin phenomenon can produce an increase in Bitcoin prices while the contrary is weak; we contribute to a research topic that has rapidly increased in the last years and however deserves more investigations.

Our results confirm the general hypothesis that short-term local trends characterize the Bitcoin time series and the possibility to forecast its values at least for small steps ahead, in the range 1 to 5.

Possibly, improved forecasting models are possible by identifying specific forms of local trends such as, e.g., parametric exponential functions, more suitable and stable for extrapolation than polynomials.

1. M.L. Guerra, L. Sorini, L. Stefanini, Quantile and Expectile Smoothing based on L_1-norm and L_2-norm F-transforms, International Journal of Approximate Reasoning, 107 (2019) 17 - 43.
2. M.L. Guerra, L. Sorini, L. Stefanini, On the approximation of a membership function by empirical quantile functions, International Journal of Approximate Reasoning, 124 (2020) 133-146.
3. I. Perfilieva, Fuzzy Transforms: Theory and Applications, Fuzzy Sets and Systems, 157 (2006) 993-1023.
4. I. Perfilieva, Fuzzy Transforms: A Challenge to conventional transforms. In P.W. Hawkes (Ed.), Advances in Images and Electron Physics, 147, Elsevier Academic Press, 2007, 137-196.
5. I. Perfilieva, V. Novák, A. Dvorak, Fuzzy transform in the analysis of data, International Journal of Approximate Reasoning, 48 (2008) 36-46.
6. I. Perfilieva, F-transform, chapter 7 of J. Kacprzyk, W. Pedrycz (Eds), Springer Handbook of Computational Intelligence, 2015, 113-130.
7. V. Kreinovich, O. Kosheleva, S. Sriboonchitta, Why Use a Fuzzy Partition in F-Transform? Axioms, 8 (3): 94 (2019).
8. E. Bouri, R. Gupta, A. K. Tiwari, D. Roubaud, Does Bitcoin hedge global uncertainty? Evidence from wavelet-based quantile-in-quantile regressions, Finance Research Letters 23 (2017) 87-95.
9. F. Bellini, B. Klar, A. Muller, E. Rosazza Gianin, Generalized quantiles as risk measures, Insurance Mathematics and Economics, 54 (2014) 41-48.
10. F. Bellini, V. Bignozzi, On elicitable risk measures, Quantitative Finance, 15/5 (2015) 725-733.
11. A. Daouia, S. Girard, G. Stupfler, Extreme M-quantiles as risk measures: From L1 to Lp optimization. Bernoulli, 25/1 (2019) 264-309.

12. A. F. Bariviera, M. J. Basgall, W. Hasperuéb, M. Naiouf, Some stylized facts of the Bitcoin market, Physica A, 484 (2017) 82-90.
13. D. Yermack, Handbook of Digital Currency, Chapter 2 - Is Bitcoin a Real Currency? An Economic Appraisal (2015) 31-43.
14. E. Demir, G. Gozgor, C. K. M. Lau, S. A. Vigne, Does economic policy uncertainty predict the Bitcoin returns? An empirical investigation, Finance Research Letters, 26 (2018) 145-149.
15. S. Corbet, B. Lucey, A. Urquhart, L. Yarovaya, cryptocurrencies as a financial asset: A systematic analysis, International Review of Financial Analysis, 62 (2019) 182-199.
16. M. Gronwald, Is Bitcoin a Commodity? On price jumps, demand shocks,and certainty of supply, Journal of International Money and Finance 97 (2019) 86–92.
17. D. Y. Aharon, M. Qadan, Bitcoin and the day-of-the-week effect, Finance Research Letters, 31 (2019) 415-424.
18. M. Qin, C. Su, R. Tao, A new basket for eggs?, Economic Modelling (2020), doi.org/10.1016/j.econmod.2020.02.031
19. E. Bouri, R. Gupta, Predicting Bitcoin returns: Comparing the roles of newspaper- and internet search-based measures of uncertainty, Finance Research Letters, doi: 10.1016/j.frl.2019.101398.
20. A. Baiga, B. M. Blaub, N. Sabaha, Price clustering and sentiment in bitcoin, Finance Research Letters, 29 (2019) 111-116.
21. W. Zhang, P. Wanga, X. Li, D.Shen, Quantifying the cross-correlations between online searches and Bitcoin market, Physica A, 509 (2018) 657-672.
22. L. Kristoufek, Power-law correlations in finance-related Google searches and their cross-correlations with volatility and traded volume: Evidence from the Dow Jones Industrial components, Physica A, 428 (2015) 194-205.
23. C. Eom, T. Kaizoji, S. H. Kang, L. Pichl, Bitcoin and investor sentiment: Statistical characteristics and predictability, Physica A, 514 (2019) 511-521.
24. V. Karalevicius, N. Degrande, J. De Weerdt, Using sentiment analysis to predict interday Bitcoin price movements, The Journal of Risk Finance, 19, 1 (2018) 56-75.
25. R. Adcock, N. Gradojevic, Non-fundamental, non-parametric Bitcoin forecasting, Physica A: Statistical Mechanics and its Applications, 531 (2019) 121727.
26. G. S. Atsalakis , I. G. Atsalaki , F. Pasiouras ,C. Zopounidis, Bitcoin price forecasting with neuro-fuzzy techniques, European Journal of Operational Research 276 (2019) 770-780.
27. C. Trucios, Forecasting Bitcoin risk measures: A robust approach, International Journal of Forecasting, 35, 3 (2019) 836-847.
28. N. A. Kyriazis, A Survey on Empirical Findings about Spillovers in cryptocurrency Markets, Journal of Risk and Financial Management, 12, 4 (2019) 170.
29. R. Bohte, L. Rossini, Comparing the Forecasting of cryptocurrencies by Bayesian Time-Varying Volatility Models, J. Risk Financial Manag, 12 (2019) 150.
30. T. Xie, Forecast Bitcoin Volatility with Least Squares Model Averaging, Econometrics, 7 (2019) 40.
31. A. Cretarola, G. Figà-Talamanca, M. Patacca. Market attention and Bitcoin price modeling: theory, estimation and option pricing. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3042029, 2017.
32. A. Cretarola, G. Figà-Talamanca, M. Patacca. A continuous time model for bitcoin price dynamics. In Mathematical and Statistical Methods for Actuarial Sciences and Finance, Springer, (2018) 273-277.
33. P. Chaim, M. P. Laurini, Is Bitcoin a bubble?, Physica A, 517 (2019) 222-232.
34. T. Panagiotidis, T. Stengos, O. Vravosinos, A Principal Component-Guided Sparse Regression Approach for the Determination of Bitcoin Returns, Journal of Risk and Financial Management, 13, 2 (2020) 33.
35. I.A., Moosa, The bitcoin: a sparkling bubble or price discovery?, Journal of Industrial and Business Economics, 47 (2020) 93–113.
36. L. Catania, S. Grassi, F. Ravazzolo, Forecasting cryptocurrencies under model and parameter instability, International Journal of Forecasting, 35/2 (2019) 485-501.
37. S. Ji, J. Kim, H. Im, A Comparative Study of Bitcoin Price Prediction Using Deep Learning, Mathematics, 7 (2019) 898.
38. S. Stavroyiannis, V. Babalos, S. Bekiros, S. Lahmiri, G. S. Uddin, The high frequency multifractal properties of Bitcoin, Physica A, 520 (2019) 62-71.

39. M. Yu, R. Gao, X. Su, X. Jin, H. Zhang, J. Song, Forecasting Bitcoin volatility: The role of leverage effect and uncertainty, Physica A (2019), https://doi.org/10.1016/j.physa.2019.03.072.
40. F. N. Zargar, D.Kumar, Long range dependence in the Bitcoin market: A study based on high-frequency data, Physica A, 515 (2019) 625–640.
41. Y. Zhang, S. Chan, J. Chu, S. Nadarajah, Stylised facts for high frequency cryptocurrency data, Physica A, 513 (2018) 598-612.
42. T. Takaishi, Statistical properties and multifractality of Bitcoin, Physica A: Statistical Mechanics and its Applications, 506 (2018) 507-519.
43. A. C. da Silva Filho, N. D. Maganini, E. Fonseca de Almeida, Multifractal analysis of Bitcoin market, Physica A, 512 (2018) 954-967.
44. M.L. Guerra, L. Stefanini, Expectile smoothing of time series using F-transform, Conference of the European Society for Fuzzy Logic and Technology, Advances in Intelligent Systems Research, 32, ISBN: 978-162993219-4 (2013) 559-564.
45. L. Coroianu, L. Stefanini, Properties of fuzzy transform obtained from Lp-minimization and a connection with Zadeh extension principle, Information Sciences, 478 (2019) 331-354.