HYBRID ARDL-MIDAS-TRANSFORMER TIME-SERIES REGRESSIONS FOR MULTI-TOPIC CRYPTO MARKET SENTIMENT DRIVEN BY PRICE AND TECHNOLOGY FACTORS

Ioannis Chalkiadakis
School of Mathematical and Computer Sciences,
Heriot-Watt University
ic14@hw.ac.uk

Gareth W. Peters
Department of Statistics & Applied Probability
University of California Santa Barbara
garethpeters@ucsb.edu

Matthew Ames
ResilientML
Melbourne, Australia
matt.ames@resilientml.com

September 2, 2021

ABSTRACT

This paper develops a novel hybrid Autoregressive Distributed Lag Mixed Data Sampling (ARDL-MIDAS) model that integrates both deep neural network multi-head attention Transformer mechanisms, and a number of covariates, including sophisticated stochastic text time-series features, into a mixed-frequency time-series regression model with long memory structure. In doing so, we demonstrate how the resulting class of ARDL-MIDAS-Transformer models allows one to maintain the interpretability of the time-series models whilst exploiting the deep neural network attention architectures. The latter may be used for higher-order interaction analysis, or, as in our use case, for design of Instrumental Variables to reduce bias in the estimation of the infinite lag ARDL-MIDAS model. Our approach produces an accurate, interpretable forecasting framework that allows one to forecast end-of-day sentiment intra-daily, with readily attainable time-series regressors.

In this regard, we conduct a statistical time-series analysis on mixed data frequencies to discover and study the relationships between sentiment from our custom stochastic text time-series sentiment framework, alternative popular sentiment extraction frameworks (BERT and VADER), and technology factors, as well as to investigate the role that price discovery has on retail cryptocurrency investors' sentiment (crypto sentiment). This is an interesting time-series modelling challenge as it involves working with time-series regression models in which the time-series response process, and the regression time-series covariates, are observed at different time scales.

Specifically, a detailed real data study is conducted where we explore the relationship between daily crypto market sentiment (of positive, negative and neutral polarity) and the intra-daily (hourly) price log-return dynamics of crypto markets. The sentiment indices constructed for a variety of "topics" and news sources are produced as a collection of time-series capturing the daily sentiment polarity signals for each "topic", namely each particular market or crypto asset. Different sentiment methods are developed in a time-series context, and utilised in the proposed hybrid regression framework. Furthermore, technology time-series factors are introduced to capture network effects, such as the hash rate which is an important aspect of money supply relating to mining of new crypto assets, and block hashing for transaction verification. Throughout our real data study, we provide guidance and insights on how to use our hybrid model to combine - in a transparent, non black-box way - covariates obtained with different time resolutions, understand the arising dynamics between these covariates, potentially under the presence of long memory structure, and, finally, successfully leverage these in forecasting applications.

Mixed-Data Sampling time-series regression (MIDAS) Transformer deep neural network multi-scale resolution data; natural language processing (NLP) text sentiment NLP time-series modeling Gegenbauer long memory

1 Introduction

Cryptocurrency markets are emerging into mainstream finance with their adoption by institutions taking place on a regular basis, and regulatory frameworks coming into place to better manage the new classes of assets, markets and risks emerging from the re-envisaging of financial models through the Decentralised Finance (DeFi) movement. However, still the overwhelming majority of investors in the crypto space are comprised of retail investors rather than institutions. They are actors in the crypto markets for a variety of reasons; they may be seeking alternatives to the traditional fiat systems that could be overly oppressive in some countries, or may be seeking safe haven from hyper inflationary domestic currencies. Note that the latter may be due to the potential offering of a reliable, censorship-resistant scarce store of wealth, speculation of price activity, convenience of efficient financial transactions, or significantly greater interest bearing accounts from lending and staking platforms. However, there is no denying the fact that the crypto markets are still significantly influenced by sentiment as an entire industry on news and analytics has arisen in the crypto space that mirrors the mainstream financial media outlets such as MSNBC, Reuters and the like. In the online and print media, this includes news brands that have been servicing the crypto community for at least 5 to 6 years consistently at this stage, and includes well-known news brands in the crypto space such as: Cointelegraph, cryptonews, CoinDesk, Bitcoin Magazine, Crypto Reddit, CryptoSlate, CryptoPotato, Coinmarketcap and Cryptoscoop, to name a few of the more widely followed news feeds.

Sentiment analysis, or opinion mining, is an active area of study in the field of natural language processing that analyses people's opinions, sentiments, evaluations, attitudes, and emotions via the computational treatment of subjectivity in text, for instance, see detailed overviews of the field of sentiment analysis in [1] and [2].

In this study, we aim to explore the relationship between crypto market sentiment and intra-daily price. In particular, we seek to study the time-series relationship between the daily sentiment time-series of two major crypto assets, i.e. Bitcoin and Ethereum, to the intra-daily time-series for the price of leading crypto assets and the volatility dynamics of crypto currency markets, on an hourly time resolution. We construct the sentiment time-series based on a collection of curated news articles about Bitcoin and Ethereum that span the last 3 years, which we have collected from widely-read crypto news sources. A range of crypto news sentiment perspectives is then captured by sentiment signals of positive, negative and neutral sentiment polarities for a variety of different cryptocurrency markets and news sources.

To achieve this, we introduce a new approach to cryptocurrency sentiment that is tailored to the crypto space context, and we compare and contrast our proposed methodology with existing sentiment extraction methods such as BERT (an attention-based Transformer sentiment model) and VADER (a rule-based model for online social media text sentiment analysis).

The studies we performed extract sentiment on particular cryptoassets, incorporating in the process the market opinion sentiment, crypto regulation sentiment and Decentralised Finance sentiment into a common sentiment index. In addition, we incorporate technology factors related to network and mining efficiency, as well as transaction costs. We finally combine this information into two classes of econometric models. First, we relate sentiment extracted by our rigorous crypto-specific framework, to other less interpretable crypto sentiment methods, using an Autoregressive Distributed Lag (ARDL) modelling framework. Having shown the utility of our proposed sentiment time-series methodology, we next adopt a time-series regression framework that will accommodate the different time scales of the response time-series of daily crypto sentiment, and the covariate time-series of hourly crypto asset prices, volatility and network effects, such as the hash rate.

We will focus our analysis on currency pair markets for the asset exchange rates of BTC/USDT and ETH/USDT extracted from an aggregate price, and obtained from CoinGecko, a leading price aggregator from a variety of centralised cryptocurrency exchanges and distributed DEXs.

1.1 Statistical Modelling and Application Contributions

In order to undertake the proposed cryptocurrency studies we have adopted and extended a class of time-series regression models known as the Mixed Data Sampling (MIDAS) models which were recently made popular in the econometrics community. They allow one to parametrically accommodate Autoregressive Distributed Lag (ARDL) models where the response time-series is sampled at a different frequency to the covariate time-series. In this work, we explore and extend the class of MIDAS models to accommodate a few additional key structures:

- First, we incorporate within the MIDAS-ARDL model structure an infinite-lag structure that is transformed to a finite lag MIDAS-ARDL model via use of the MIDAS-modified classical Koyck transform. We call this the Koyck-MIDAS transform. We study the calibration of these models using Instrumental Variables (IV) for the sentiment signal, constructed based on VADER and deep learning solutions such as BERT. This in turn produces a hybrid time-series model that is denoted as the ARDL-MIDAS-Transformer model, which is an illustration of a class of ARDL-MIDAS-NeuralNet time-series regression models in which the neural network is a Transformer model that combines attention mechanisms with a Feed-Forward Neural-Network. This is used to construct the Instrumental Variables in order to reduce bias in the estimation of the infinite-lag Koyck-transformed ARDL-MIDAS model.
- Second, we incorporate a long memory structure into the MIDAS-ARDL model class creating a form of MIDAS-GARDL model where the G stands for the class of long memory structures we incorporate, known as the Gegenbauer long memory polynomial filtertaps.
- Third, we study the combination of MIDAS exponential Almon weight functions, Koyck-transformed geometric weight decay structures, and the Gegenbauer polynomial weight function generator in the case studies undertaken in the crypto space.

From an application perspective we make the following additional practical contributions:

- In terms of statistical modelling, we develop an approach to constructing time-series formulations of sentiment in order to quantify in a single index the market sentiment that is extracted from multiple collections of daily sets of news articles. We propose a novel way to construct the sentiment index, and provide a combining rule to obtain a single index, which is important to summarise the sentiment content from different news sources, thus facilitating sentiment incorporation into a time-series framework. Details on the construction and text sentiment usage in the cryptospace are presented in Section 4.
- We demonstrate that our approach to constructing sentiment time-series is distinct from those that can be derived by popular deep learning, Transformer solutions such as BERT, but also rule-based approaches such as VADER. This is achieved using ARDL time-series regressions methods and formal statistical tests (Section 6.1).
- We analyse the relationships between financial intra-day price signals, technology and network factors related to money supply in cryptocurrencies, and the daily sentiment time-series signal. The goal of the analysis is to enhance understanding of the evolving dynamics between covariates and responses of different time scales, thus facilitating in-sample fitting and out-of-sample forecasting applications (Section 6.2).

2 ARDL-MIDAS Long-Memory Time-Series Regressions

In this section we present the modelling framework that incorporates four working components: Koyck-transformed infinite-lag Autoregressive Distributed Lag time-series regressions; Mixed Data Sampling (MIDAS) multi-time resolution time-series regressions; natural language text component obtained both via crypto-tailored text processing and Transformer deep neural network architectures; and Gegenbauer long memory structure to parametrically capture persistence. The logic of combining these models is to exploit the ability of deep learning architectures to learn higher-order feature structures that can act as generative model inputs to interpretable regression time-series models. We will begin with an overview of the overall regression structure before exploring each component.

The context of this study naturally allows one to explore a range of both Autoregressive Distributed Lag models, as well as MIDAS regression models. There is a subtle difference between such classes of models as explained in detail in [3]. One can not strictly classify a MIDAS model as an autoregressive model in the standard sense as they involve regressors with different sampling frequencies. This can be understood as a consequence of the fact that autoregressive structures implicitly assume that data are sampled at the same frequency in the past. Instead, MIDAS regressions share some features with distributed lag models but also have unique features we will adopt for part of this study.

The following notation conventions are adopted to accommodate the various time scales considered in the MIDAS structures. The low frequency time scale is indexed by t for the regression response process $\{y_t,\ t \in \mathbb{Z}\}$ and the higher frequency time scale denoted by m for the regression covariate time-series processes $\{x_t^{(m)},\ t \in \mathbb{Z}\}$ which is observed m-times faster than time scale t, such that for each low frequency period t one has m values of $x_{t-1+1/m_i}^{(m_i)}, x_{t-1+2/m_i}, \ldots, x_t$. Here, m_i will denote the i-th high frequency time scale and there may be numerous higher frequencies time scales used depending on the covariates utilised. Two lag operators are utilised:

- a low frequency lag operator, which is denoted by L and will be applied as: $LY_t = Y_{t-1}$; and

- a high frequency lag operator $L^{1/m}$ which will apply to time-series observed m-times faster than the t time scale, and which, when applied, produces $L^{1/m}X_t^{(m)} = X_{t-1/m}^{(m)}$.

From this we will define the following characteristic polynomials for the autoregressive (AR) and distributed lag (DL) time-series components:

$$\begin{aligned}
\Phi_p(L) &= 1 - \sum_{j=1}^{p} \phi_j L^j, \\
\boldsymbol{x}_t^{(m)} &= \left[x_{t-1+1/m}, x_{t-1+2/m}, \ldots, x_t\right], \\
\boldsymbol{\beta}_k(L^{1/m}) &= \sum_{j=1}^{k} \boldsymbol{\beta}_j L^{j/m}, \;\; \boldsymbol{\beta}_j = [\beta_{j,0}, \ldots, \beta_{j,m}]^T, \\
L^{j/m}\boldsymbol{x}_t^{(m)} &= \left[L^{j/m}x_{t-1+1/m}, L^{j/m}x_{t-1+2/m}, \ldots, L^{j/m}x_t\right]^T.
\end{aligned} \tag{1}$$

The standard multiple ARDL-MIDAS model would then be given by a regression structure

$$\Phi_p(L)Y_t = \sum_{j=1}^{J} \boldsymbol{\beta}_k^{(j)}(L^{1/m_j})\boldsymbol{X}_t^{(m_j)} + \epsilon_t, \tag{2}$$

with $\Phi_p(L)$ the standard AR characteristic polynomial expressed in lag operator L at time scale t, and $\beta(L^{1/m_j})$ is the j-th time-series covariate's characteristic polynomial with MIDAS weight function expressed in lag operator $L^{1/m}$, namely at time scale m_j times faster than t. This MIDAS polynomial is applied to the covariate time-series observed at the time scale of m_j times faster than t. Note that in this notation we have a vector covariate at time t constructed from the m_j sub-time steps.

We wish to extend this model in three important ways:

1. Considering an infinite-lag structure at time scale m_j with $\boldsymbol{\beta}_\infty(L^{1/m})$ and developing an ARDL-MIDAS Koyck Transform.
2. Considering a fractional integration of the Gegenbauer form, to capture potential for long memory structure in the regression relationship, where we add the fractional difference operator $\left(1 - 2uL^{1/m_j} + L^{2/m_j}\right)^{-d}$.
3. Adding a generative embedding model for construction of high-order covariate feature interactions that can be combined within the multiple ARDL-MIDAS-Gegenbauer time-series regression and includes:
    - Transformer regression structures based on deep neural network multi-head attention architectures;
    - Natural Language crypto-specific covariate feature generation, which may range from time-series on semantics and word distributions to Context-Free Grammar parsing higher-order features.

2.1 Infinite-Lag Autoregressive Distributed Lag (ARDL) Regressions

In this section we briefly recall the basic framework of the ARDL regression modelling structure that will be adopted in the studies performed. The concept of distributed lag models is widely studied in econometrics and time-series literature and dates back to early works in the former, e.g. see the following papers on the estimation of such models [3], [4] and [5].

A stylised distributed lag model is a time-series model in which the effect of a covariate on an outcome variable occurs over time. By adding autoregressive lags to a simple distributed lag model, a new model is formed, called an autoregressive distributed lag model (ARDL). A general ARDL(p,k) model is defined as follows either in expanded form or in terms of characteristic AR and DL polynomials:

$$Y_t = \mu + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{k} \beta_j X_{t-j} + \varepsilon_t,$$

where ε_t is a stationary white noise error term, $\Phi(L)$ and $\beta(L)$ are respectively order-p and order-k characteristic polynomials for the AR and DL components expressed with regard to the backshift operator L.

In many cases, it can be challenging to determine the appropriate choice of lag structure for k, so one may set up an infinite-lag model structure by setting $k = \infty$. This model is very similar to an ARMA model, except that the infinite-

lag polynomial is applied to the explanatory variable rather than the error term as would be the case in an ARMA structure. As such, this class of models is termed an infinite ARDL model:

$$Y_t = \mu + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{\infty} \beta_j X_{t-j} + \varepsilon_t. \tag{3}$$

It can be immediately recognised that it is impossible to estimate the coefficients of equation (3) since the number of unknown parameters is infinite, therefore further assumptions are required to make the problem tractable.

One popular way to solve the infinite-lag distributed models estimation problem is to impose a parametrisation on the relationship of the infinite lag coefficients to map the problem back to a finite parameter space. One may use for instance a geometric distributed lag model. As indicated by the name, these models are based on the geometric distribution. One of the most popular geometric distributed lag models is the Koyck model [6]. Under Koyck's approach one can assume that all the coefficients of equation (3) have the same sign and decline geometrically, with a specified rate of decay. Then, by taking advantage of the geometric series convergence, the Koyck model turns the infinite coefficients of equation (3) into an equation that includes a finite number of unknown parameters.

Based on the Koyck transformation method, in equation (3) we make the substitution of a geometric decaying coefficient relationship given by $\beta_j = \mu\gamma^j$ where $0 < \gamma < 1$. Using the convergence of geometric series, equation (3) can now be rewritten as follows:

$$Y_t = \mu' + \phi' Y_{t-1} + \phi'' Y_{t-p-1} + \sum_{i=2}^{p} \varphi_i Y_{t-i} + \mu X_t + \varepsilon'_t, \tag{4}$$

where $\mu' = (1-\gamma)\mu$, $\phi' = \phi_1 + \gamma$, $\phi'' = -\gamma\phi_p$, $\varphi_i = \phi_i - \gamma\phi_{i-1}$, and $(\varepsilon_t - \gamma\varepsilon_{t-1}) = \varepsilon'_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. The operation of the Koyck model to decrease the number of parameters in the model is appreciable; however, whilst the problem of an infinite number of parameters is resolved by this assumed functional parametrisation, the reformulated model then produces a challenge for parameter estimation. In equation (4), the error term ε'_t and Y_{t-1} are not independent anymore. Hence, this renewed equation cannot be efficiently solved using conventional techniques that solve the regression models in an unbiased manner.

One common way to resolve this challenge is to introduce the concept of an instrumental variable. In our model, Y_{t-1} should be replaced by an instrumental variable which is independent of ε'_t, whilst capturing the basic dynamical structure of Y_{t-1}. If such an IV can be constructed then the new model can be solved using the conventional regression model estimation techniques, see [7]. One must be careful with the introduction of an IV to resolve this challenge, as, whilst the estimation can be performed, the properties of the resulting estimators will depend on the performance of the constructed IV. In later sections, we will demonstrate how to use deep neural network Transformer methods to construct such instrumental variables.

2.2 Fractionally Integrated Mixed Data Sampling (MIDAS) Regressions for Long Memory Regressions

In the MIDAS regression model context, the objective is to accommodate such distributed lag structures but the time-series of observations and regressors acting as lagged predictors are no longer sampled or observed at the same temporal resolution. It is assumed that the covariate variables of the ARDL-type model are now observed at higher frequency than the covariates in an AR-type regression.

In [8] and [9], such classes of problems were studied and the resulting modelling framework was named the MIDAS class of distributed lag regressions. The MIDAS structures have been extended broadly by others such as [10], [11], and [12]. This is a suitable model structure with established estimation techniques that is capable of treating mixed frequency ARDL-type regression models.

Let the variable Y_t represent the crypto sentiment constructed at a daily sampling frequency and let X_t represent an explanatory factor such as crypto asset prices, technology factors such as hash rate which are sampled m times faster than Y_t (as an example, when Y_t is daily, $X_t^{(m)}$ is sampled hourly in the 24hr, 7 day per week crypto markets giving $m = 24$). Suppose that Y_t is available once between $t-1$ and t. Using the MIDAS model, we want to project Y_t onto a history of lagged observations of $X^m_{t-j/m}$. A simple MIDAS regression model is defined as below:

$$Y_t = \mu + \beta_1 B(L^{1/m}; \psi) X_t^{(m)} + \varepsilon_t^{(m)}, \tag{5}$$

where $B(L^{1/m};\psi) = \sum_{k=0}^{K} B(k;\psi)L^{k/m}$ and $L^{1/m}$ is a lag operator on fractional time scale such that $L^{1/m}X_t^{(m)} = X_{t-1/m}^{(m)}$, μ and β_1 are the unknown parameters of the model, and $\varepsilon_t^{(m)}$ is the error term. In equation (5), the lag coefficients in $B(k;\psi)$ are parametrised as a function of a low dimensional vector of the parameters ψ.

One of the concepts in introducing the MIDAS models is to take advantage of the lag polynomials. In [8], the authors use lag polynomials to avoid the parameter proliferation problem, and to reduce the cost of estimation a variety of finite basis models are considered - we will focus on the Exponential-Almon family. By applying some modifications to the Almon lag, [8] introduce the Exponential Almon lag defined as:

$$B(k;\psi) = \frac{\exp(\psi_1 k + \cdots + \psi_Q k^Q)}{\sum_{k=1}^{K}\exp(\psi_1 k + \cdots + \psi_Q k^Q)},$$

where K is the number of lags required in equation (5).

In this work, we are interested in working with the potential for a strong persistence in the ARDL-MIDAS regression structure, which we will demonstrate can be achieved through introduction of a long memory component in the model.

A stationary time-series process $\boldsymbol{Y} \equiv \{Y_t\}_{t=1:T}$ is said to to be a long memory stationary process if the following condition [13] holds in terms of the divergence of the autocorrelation function for Y_t and Y_{t+j} at lag j:

$$\lim_{n\to\infty}\sum_{j=-n}^{n}|\rho(j)| \to \infty, \tag{6}$$

where

$$\rho(j) = \frac{\text{Cov}(Y_t, Y_{t+j})}{\sqrt{\text{Var}(Y_t), \text{Var}(Y_t + j)}}. \tag{7}$$

We will parametrise processes with this property into the ARDL-MIDAS model through a fractional difference operator that will admit an infinite-lag Gegenbauer functional polynomial series generator that can be combined within the ARDL-MIDAS models previously presented. We introduce, for the first time we believe, this new class of lag basis functions that incorporate long memory features to the MIDAS structure to produce a family of fractional-MIDAS basis functions that can produce long memory regression effects in the distributed lag factors at the high-frequency time scale m.

Definition 1 (Gegenbauer MIDAS Basis)**.** Consider the fractional long-memory Gegenbauer weight functional form given by

$$B(L^{1/m};\psi) = \sum_{j=0}^{\infty}\psi_j L^{j/m} = (1 - 2uL^{1/m} + L^{2/m})^{-d},$$

with ψ_j given by Gegenbauer polynomial functions

$$\psi_j = \sum_{q=0}^{[j/2]}\frac{(-1)^q(2u)^{j-2q}\Gamma(d-q+j)}{q!(j-2q)!\Gamma(d)}, \tag{8}$$

where $|u| < 1$, $d \in (0, 1/2)$ and $[j/2]$ represents the integer part of $j/2$. Furthermore, the Gegenbauer polynomials satisfy the recursive calculation given by

$$\psi_j = 2u\left(\frac{d-1}{j} + 1\right)\psi_{j-1} - \left(2\frac{d-1}{j} + 1\right)\psi_{j-2}, \tag{9}$$

where $\psi_0 = 1$, $\psi_1 = 2du$ and $\psi_2 = -d + 2d(1+d)u^2$.

Remark 1. The following remarks characterise this class of Gegenbauer-MIDAS coefficient functions that we introduce:

- The class of fractional Gegenbauer-MIDAS basis functions allows one to control the strength of the long-memory in the process through selection of d and u.
- If $u = 1$ one produces an ACF function that is strictly positive and decays with a hyperbolic decay rate.
- For $|u| < 1$ the ACF will oscillate between positive and negative with period dictated by d and a hyperbolic envelope decay rate.
- As $d \uparrow 0.5$ the strength of long-memory increases, the slower the hyperbolic decay of the coefficients in the MIDAS weight function will decay and therefore the longer the past of $X_t^{(m)}$ will influence the current regression response.

We will work with the extended MIDAS framework often termed in the econometrics literature as the Multi-MIDAS regression structure. In this model, one may adopt multiple covariates with different time scales as follows, for the d-variate case:

$$Y_t = \mu + \sum_{i=1}^{J} \beta_{1,i} B(L^{1/m_i}; \psi_i) X_{t,i}^{(m_i)} + \sum_{i=1}^{d} \varepsilon_t^{(m_i)}, \tag{10}$$

where one has J covariates each sampled at time scales $\{m_1, m_2, \ldots, m_J\}$ with associated driving white noise processes for each time scale. We will consider in the real data case studies the situation where $J = 2$ and $m_1 = 24$ and $m_2 = 1$. Often we will assume that we subsume all the driving noise processes for the regression, in the case of i.i.d. Gaussian errors, into one driving noise process given by $\epsilon_t := \sum_{i=1}^{J} \varepsilon_t^{(m_i)}$.

2.3 Koyck Infinite-Lag ARDL-MIDAS(p, ∞, K, m) Regressions

An ARDL-MIDAS model can be expressed in numerous ways. In this work, we have built upon the approach of [14] and we have extended it by incorporating the infinite-lag ARDL(p, ∞) Koyck transform model with the MIDAS structure. Using Equation 4 and Equation 10, we produce a variation of the classical ARDL(p, ∞) time-series regression, with the normalised $\sum_k^K B(k; \psi) L^{k/m}$ MIDAS weight function, given in the ARDL-MIDAS(p, ∞, K, m) context as follows:

$$\begin{aligned} Y_t &= \mu + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{\infty} \beta_j B(L^{1/m}; \psi) X_{t-j}^{(m)} + \varepsilon_t^{(m)} \\ &= \mu + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{\infty} \sum_{k=0}^{K} \beta_j B(k; \psi) L^{k/m} X_{t-j}^{(m)} + \varepsilon_t^{(m)} \\ &= \mu + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{\infty} \sum_{k=0}^{K} \beta_j B(k; \psi) X_{t-j-k/m}^{(m)} + \varepsilon_t^{(m)}, \end{aligned} \tag{11}$$

which can be written in compact form as

$$\Phi_p(L) Y_t = \mu + \widetilde{\beta}_\infty(L, L^{1/m}) X_t^{(m)} + \varepsilon_t^{(m)}, \tag{12}$$

with the double polynomial given by

$$\widetilde{\beta}_\infty(L, L^{1/m}) = \sum_{j=0}^{\infty} \sum_{k=0}^{K} \beta_j B(k; \psi) L^{j+k/m}. \tag{13}$$

We will now introduce for this class of models a variation of the classical Koyck transform that we will denote as the MIDAS-Koyck Transform which we will use to refactor this model into a parsimonious parametrisation as detailed in the following proposition.

Proposition 2.1 (MIDAS-Koyck Transform)**.** *Consider the time-series model given by the ARDL-MIDAS(p, ∞, K, m) model specified as follows:*

$$\Phi_p(L) Y_t = \mu + \widetilde{\beta}_\infty(L, L^{1/m}) X_t^{(m)} + \varepsilon_t^{(m)} \tag{14}$$

Then the modified Koyck transform applied to this model uses the modified geometric decay characteristic polynomial at time scale t given as follows:

$$\widetilde{\beta}_\infty(L, L^{1/m}) := \beta_0 B(L^{1/m}; \psi) \sum_{j=1}^{\infty} \gamma^j L^j = \frac{\beta_0 B(L^{1/m}; \psi)}{1 - \gamma L}, \tag{15}$$

for $\gamma \in (0, 1)$, which can transform this ARDL-MIDAS(p, ∞, K, m) into the simplified ARDL-MIDAS$(p, 1, K, m)$ given by

$$Y_t = \beta_0' + \phi' Y_{t-1} + \phi'' Y_{t-1-p} + \sum_{i=2}^{p} \varphi_i Y_{t-i} + \beta_0 \sum_{k=0}^{K} B(k; \psi) X_{t-k/m}^{(m)} + \varepsilon_t^{(m)'} \tag{16}$$

where $\mu' = (1 - \gamma)\mu$, $\phi' = \beta_1 + \gamma$, $\phi'' = -\gamma\beta_p$, $\varphi_i = \beta_i - \gamma\beta_{i-1}$, and $(\varepsilon_t^{(m)} - \gamma\varepsilon_{t-1}^{(m)}) = \varepsilon_t^{(m)'} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.

Proof. The derivation of this MIDAS-Koyck transform is a basic extension of the standard Koyck transform approach with a MIDAS component applied. This proceeds to transform the ARDL-MIDAS(p, ∞, K, m) model as follows:

$$\begin{aligned} Y_t &= \beta_0 + \sum_{i=1}^{p} \beta_i Y_{t-i} + \beta_0 B(L^{1/m}; \psi) \sum_{j=0}^{+\infty} \gamma^j X_{t-j}^{(m)} + \epsilon_t^{(m)}, \\ Y_{t-1} &= \beta_0 + \sum_{i=1}^{p} \beta_i Y_{t-1-i} + \beta_0 B(L^{1/m}; \psi) \sum_{j=0}^{\infty} \gamma^j X_{t-1-j}^{(m)} + \epsilon_{t-1}^{(m)}. \end{aligned} \tag{17}$$

If one then multiplies the second row with the geometric decay rate γ:

$$\gamma Y_{t-1} = \gamma\beta_0 + \gamma \sum_{i=1}^{p} \beta_i Y_{t-1-i} + \beta_0 B(L^{1/m}; \psi) \sum_{j=0}^{+\infty} \gamma^{j+1} X_{t-1-j}^{(m)} + \gamma\epsilon_{t-1}^{(m)}, \tag{18}$$

and subtracting the expressions in equations 17 (for Y_t) and 18 one then obtains:

$$Y_t - \gamma Y_{t-1} = (1-\gamma)\beta_0 + \sum_{i=1}^{p} \beta_i (Y_{t-i} - \gamma Y_{t-1-i}) + \beta_0 B(L^{1/m}; \psi) X_t^{(m)} + \epsilon_t^{(m)} - \gamma\epsilon_{t-1}^{(m)},$$

which results in

$$Y_t - \gamma Y_{t-1} = (1-\gamma)\beta_0 + \beta_1 Y_{t-1} - \beta_p \gamma Y_{t-1-p} + \sum_{i=2}^{p} (\beta_i - \beta_{i-1}\gamma) Y_{t-i} + \beta_0 B(L^{1/m}; \psi) X_t^{(m)} + \epsilon_t^{(m)} - \gamma\epsilon_{t-1}^{(m)},$$

which then gives the desired result after some changes of variable. □

Remark 2. One can make the following remarks about this ARDL-MIDAS(p, ∞, K, m) model transformed via the MIDAS-Koyck Transform to an ARDL-MIDAS$(p, 1, K, m)$ model:

- This specification of the infinite-lag structure allows one to specifically accommodate a type of m-period seasonal structure for the regressors at time scale m which will be consistent with a period t seasonal pattern for the high frequency covariates. This has an advantage over other approaches to constructing infinite-lag structures due to the fact that it does not require assumptions of knowledge of the slower time scale process y_t at times between $t-1$ and t.
- One should consider the use of an Instrumental Variable to replace the term Y_{t-1} in order to attempt to break the correlation that would be present between this variable and the transformed regression error term $\varepsilon_t^{(m)'}$.

Such a structure is also suitable for the application considered, where, if one lines up the time reference t with one of the leading markets morning period, for instance in Europe or Korea, then one would expect a periodic daily structure, when referenced to a US-based timezone. Based on this result, we can then construct the following specialised example corollary models incorporating the Gegenbauer long memory coefficient functions within the infinite-lag ARDL-MIDAS model.

Corollary 2.1 (Gegenbauer-MIDAS Koyck Transform)**.** *Consider the time series model given by the ARDL-Gegenbauer-MIDAS(p, ∞, ∞, m) model specified as follows:*

$$\Phi_p(L) Y_t = \mu + \beta_\infty(L)(1 - 2uL^{1/m} + L^{2/m})^{-d} X_t^{(m)} + \varepsilon_t^{(m)}. \tag{19}$$

Then, the modified Koyck transform applied to this fractionally integrated MIDAS model produces

$$\begin{aligned} Y_t &= \beta_0' + \phi' Y_{t-1} + \phi'' Y_{t-p-1} + \sum_{i=2}^{p} \varphi_i Y_{t-i} + \beta_0 (1 - 2uL^{1/m} + L^{2/m})^{-d} X_t^{(m)} + \varepsilon_t^{(m)'} \\ &= \beta_0' + \phi' Y_{t-1} + \phi'' Y_{t-p-1} + \sum_{i=2}^{p} \varphi_i Y_{t-i} + \beta_0 \sum_{j=0}^{\infty} \sum_{q=0}^{[j/2]} \frac{(-1)^q (2u)^{j-2q} \Gamma(d-q+j)}{q!(j-2q)!\Gamma(d)} X_{t-j/m}^{(m)} + \varepsilon_t^{(m)'}, \end{aligned}$$

where $|u| < 1$, $d \in (0, 1/2)$, $\mu' = (1-\gamma)\mu$, $\phi' = \beta_1 + \gamma$, $\phi'' = -\gamma\beta_p$, $\varphi_i = \beta_i - \gamma\beta_{i-1}$, *and* $(\varepsilon_t^{(m)} - \gamma\varepsilon_{t-1}^{(m)}) = \varepsilon_t^{(m)'} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.

3 Hybrid ARDL-MIDAS-NeuralNet Time-Series Regressions

To complete our framework, we now present a general hybrid time-series regression structure that allows one to incorporate into the ARDL-MIDAS time-series structure additional components obtained from a deep neural network architecture. We will consider a generic example at first using a feed-forward neural network and then we will specialise, for the NLP sentiment time-series context, to Transformer multi-head attention mechanisms.

3.1 Hybrid ARDL-MIDAS-FFNN Time-Series Regressions

At this point, it suffices to capture the idea where we extend the ARDL-MIDAS regression to the hybrid Feed-Forward Neural Network (FFNN) version (ARDL-MIDAS-FFNN), with neural network depth n (number of computation layers) and additional covariate time-series denoted by $\{\boldsymbol{S}_t^{(m)}\}$ that are observed at frequency m times faster than t,

$$\Phi_p(L)Y_t = \sum_{j=1}^{J} \boldsymbol{\beta}_k^{(j)}(L^{1/m_j})\boldsymbol{X}_t^{(m_j)} + \left\langle \boldsymbol{\beta}_q(L^{1/m}), \left(z^{(n)} \circ z^{(n-1)} \circ \cdots \circ z^{(1)}\right)(\boldsymbol{S}_t^{(m)}) \right\rangle + \epsilon_t, \tag{20}$$

where $z^{(l)}$, $1 \leq l \leq n$ denotes the l-th hidden network layer of dimension $q_m + 1 \in \mathbb{N}$ and $\boldsymbol{\beta}_q(L^{1/m})$ is the MIDAS Distributed Lag (DL) operator at time resolution m times faster than t for a single neuron output layer. Note that this can trivially be generalised for each output layer neuron, but for clarity of notation it suffices to consider this simple case for the model specification.

This *readout* transformed covariate time-series is then combined with a MIDAS DL structure to produce an additional higher-order interaction component that enhances the linear non-interaction terms of the ARDL-MIDAS model through the *readout lag operator parameters* in polynomial $\boldsymbol{\beta}_q(L^{1/m})$. This can then be used either as additional trend structure, or, as we will use them, as Instrumental Variables to reduce bias in the infinite lag ARDL-MIDAS setting.

In this architecture, for a given activation function in the FFNN denoted by $\psi : \mathbb{R} \to \mathbb{R}$, the l-th hidden network layer with activation function ψ is a map

$$z^{(l)} : \{1\} \times \mathbb{R}^{q_{n-1}} \to \{1\} \times \mathbb{R}^{q_n}, \quad \boldsymbol{z} \to z^{(l)}(\boldsymbol{z}) = \left(1, z_1^{(l)}(\boldsymbol{z}), \ldots, z_{q_l}^{(l)}(\boldsymbol{z})\right) \tag{21}$$

with *hidden neurons* $z_j^{(l)}$, $1 \leq j \leq q_l$, being described by

$$z_j^{(l)}(\boldsymbol{z}) = \psi\langle \alpha_j^{(l)}, \boldsymbol{z} \rangle, \tag{22}$$

for given network parameters $\boldsymbol{\alpha}_j^{(l)} \in \mathbb{R}^{q_{l-1}+1}$.

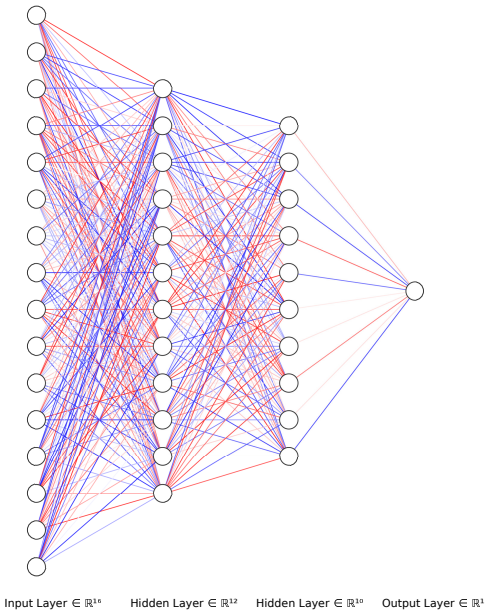
For instance, in the case of the FFNN we could set $n = 3$, $q_1 = 16, q_2 = 12$ and $q_3 = 10$ and we would have the output $\left(z^{(n)} \circ z^{(n-1)} \circ \cdots \circ z^{(1)}\right)(\boldsymbol{S}_t^{(m)})$ that is projected as a time-series into the regression via the MIDAS lag operator and depicted in this case by the architecture given in Figure 1.

3.2 Hybrid ARDL-MIDAS-Transformer Time-Series Regressions

In the context of Natural Language Processing and sentiment analysis it will be meaningful to also consider a class of deep neural network architectures known as "Transformers". This is a specific neural network model which has proven to be especially effective for common natural language processing tasks such as sentiment analysis, see discussion in [15]. In particular, the Transformer model makes use of multiple attention mechanisms [15] which are effectively incorporated as encoders in sequence-to-sequence architectures [16], which aim to capture the sequential nature of text data. A thorough and detailed account of a Transformer is beyond the scope of this manuscript; suffice to say we will think of it as a more complex projection function of the input time-series generically denoted by $\{\boldsymbol{S}_t^{(m)}\}$ to produce a transformed output time-series denoted by $\{T(\boldsymbol{S}_t^{(m)})\}$, in which the transformation is comprised of significantly more components than the FFNN case. The Transformer-based model deployed in the study of the current manuscript and which has been the state-of-the-art in NLP, is BERT [17], which is benefiting from multiple attention modules ("heads"), the basic format of one of which is illustrated diagrammatically in Figure 2. A detailed description of BERT's architecture is available in [18] and [19]. Having this component, this will allow us to develop an ARDL-MIDAS-Transformer model as follows:

$$\Phi_p(L)Y_t = \sum_{j=1}^{J} \boldsymbol{\beta}_k^{(j)}(L^{1/m_j})\boldsymbol{X}_t^{(m_j)} + \langle \boldsymbol{\beta}_q(L^{1/m}), T(\boldsymbol{S}_t^{(m)}) \rangle + \epsilon_t. \tag{23}$$

Figure 1: Example of a simple 3-layer Deep Neural Network architecture for a Feed-Forward Neural Net



To briefly explain the mapping inside the Transformer which we denoted generically by $T(\cdot)$, we may summarise it conceptually as follows. A Sequence-to-Sequence ("Seq2Seq") architecture is a neural net system configuration, which comprises two components: an Encoder and a Decoder. The Encoder takes the input sequence and maps it into a higher dimensional feature space (n-dimensional vector). That abstract vector is then fed into the Decoder which turns it into an output sequence. The output sequence can be in another language, symbols, or a copy of the input. An initial, more intuitive, popular choice for this type of model is Long-Short-Term-Memory (LSTM)-based models. With sequence-dependent data, the LSTM modules can give meaning to the sequence while remembering (or forgetting) the parts they find important (or unimportant). Sentences, for example, are sequence-dependent since the order of the words is crucial for understanding the sentence, hence LSTM models are a natural choice for this type of data. Therefore, a very basic choice for the Encoder and the Decoder of a Sequence-to-Sequence model could be a single LSTM for each of the two components.

The Transformer is then integrating the attention mechanism into the Seq2Seq modelling, effectively looking at an input sequence and deciding at each step which other parts of the sequence are important. In contrast to LSTM-based encoders, for each input that the attention-based Encoder reads, the attention mechanism takes into account at the same time several other inputs that precede and follow the current input and decides which ones are important by attributing different weights to those inputs. The Decoder will then take as input the encoded sentence and the weights provided by the attention mechanism.

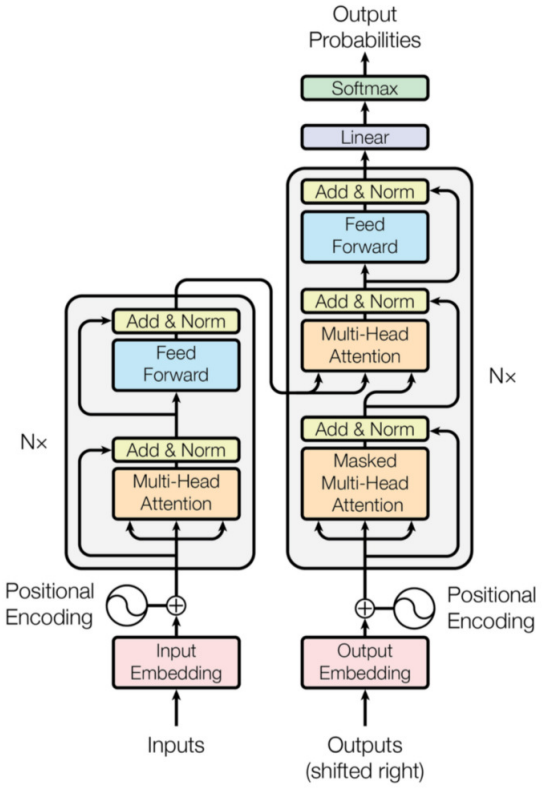
Hence, BERT (BERT: Bidirectional Encoder Representations from Transformers) is an architecture for transforming one sequence into another one via an Encoder and Decoder, but it differs from previous sequence-to-sequence models because it does not imply any Recurrent Networks (GRU, LSTM, etc.), but rather is solely based on Transformer modules. The BERT Transformer architecture provides significant improvements on Natural Language Tasks such as sentiment extraction. We use it in a manner to facilitate effective Instrumental Variable design to reduce bias in the estimation of our infinite-lag ARDL-MIDAS-Koyck-transformed Gegenbauer long memory time-series regression models.

3.3 Deep Neural Networks for Instrumental Variable Design to Reduce Estimation Bias

In this section we will explain why we have chosen to extend our ARDL-MIDAS structure to an ARDL-MIDAS-NN or ARDL-MIDAS-Transformer type time-series regression. Consider the case of the infinite lag ARDL-MIDAS model parametrised under a Koyck Transform as in Proposition 2.1 then we know that the transformed model given by

$$\Phi_p(L)Y_t = \mu + \widetilde{\beta}_\infty(L, L^{1/m})X_t^{(m)} + \varepsilon_t^{(m)} \tag{24}$$

Figure 2: The Transformer model architecture that forms the fundamental building block of BERT [17] for NLP tasks such as sentiment analysis (figure from "Attention Is All You Need" by [15]).



is modified via the proposed MIDAS-Koyck Transform to produce, for $\gamma \in (0,1)$ a transform of the ARDL-MIDAS(p,∞,K,m) model into the simplified ARDL-MIDAS$(p,1,K,m)$ given by

$$Y_t = \beta_0' + \phi' Y_{t-1} + \phi'' Y_{t-1-p} + \sum_{i=2}^{p} \varphi_i Y_{t-i} + \beta_0 \sum_{k=0}^{K} B(k;\psi) X_{t-k/m}^{(m)} + \varepsilon_t^{(m)'}. \tag{25}$$

In this model the classical Ordinary Least Squares OLS estimator will be biased by the fact that Y_{t-1} and $\varepsilon_t^{(m)'} = \epsilon_t^{(m)} - \epsilon_{t-1}^{(m)}$ are no longer independent, as a result of the Koyck Transformation. In this case it is standard practice to replace the regression variable Y_{t-1} with an Instrumental Variable which should capture similar information to Y_{t-1} but remain uncorrelated with white noise $\varepsilon_t^{(m)'}$. This is where we use the Neural Network structure to build such an Instrumental Variable and we obtain the regression model given by

$$Y_t = \beta_0' + \phi' \widetilde{Y}_{t-1} + \phi'' \widetilde{Y}_{t-1-p} + \sum_{i=2}^{p} \varphi_i \widetilde{Y}_{t-i} + \beta_0 \sum_{k=0}^{K} B(k;\psi) X_{t-k/m}^{(m)} + \varepsilon_t^{(m)'}. \tag{26}$$

where we select the instrumental variable as

$$\widetilde{Y}_t = \langle \boldsymbol{\beta}_q(L^{1/m}), \left(z^{(n)} \circ z^{(n-1)} \circ \cdots \circ z^{(1)}\right)(\boldsymbol{S}_{t-1}^{(m)}) \rangle. \tag{27}$$

We will explain this further below in the context of the Natural Language processing sentiment time-series context where we will consider state-of-the-art transformer models.

4 Natural Language Processing for Sentiment Time-series Structures

In this section we discuss the specifics of the application studied in this paper, where the response time-series, denoted by $\{Y_t\}$, for our ARDL-MIDAS-Transformer model is a novel construction of an entropy-based sentiment time-series from a corpus of news articles specific to particular financial assets, in this case from the cryptocurrency market. We

will also discuss how to construct the instrumental variable time-series $\{\widetilde{Y}_t\}$ to replace $\{Y_t\}$ in the infinite-lag ARDL-MIDAS-Transformer model as discussed in Section 3.3. This will be obtained from what is known in machine learning as generative embedding feature learning and we will use specific mechanisms for this in the NLP sentiment context: Transformer models as presented in the previous section (BERT [17]) and semantic- and grammar-based rules for sentiment extraction such as VADER (Valence Aware Dictionary for sEntiment Reasoning [20]).

Pre-trained Transformer models such as BERT have achieved state-of-the-art performance on natural language processing tasks and have been adopted as feature extractors for solving downstream tasks such as question answering, natural language inference, and sentiment analysis. The current state-of-the-art Transformer based pre-trained models consist of dozens of layers and millions of parameters. While deeper and wider models yield better performance, they also need large GPU/TPU memory modules and a significant amount of text corpus for fine-tuning. For example, BERT-large is trained with 335 million parameters, and requires at least 16 GB of GPU memory to fine-tune (24 GB is the minimum recommended). The larger size of these models limits their applicability in time- and memory-constrained environments. Furthermore, there is an entire discipline now emerging to attempt to simplify and understand the layers of such complex architectures, see examples such as [21].

In the context of sentiment extraction, alternative methods have been developed based on tailored Sentiment Lexicons or Context-Free Grammar (CFG) parsing; a popular example explored in this work is the VADER approach and its associated micro-blogging sentiment lexicons. A substantial number of sentiment analysis approaches rely greatly on an underlying sentiment (or opinion) lexicon. A sentiment lexicon is a list of lexical features (e.g. words) which are generally labelled according to their semantic orientation as either positive or negative.

Manually creating and validating such lists of opinion-bearing features, while being among the most robust methods for generating reliable sentiment lexicons, is also one of the most time-consuming. For this reason, much of the applied research leveraging sentiment analysis relies heavily on preexisting manually constructed lexicons ([22], [23], [24]) in which words are categorized into binary classes (i.e., either positive or negative) according to their context free semantic orientation. We have developed a tailored lexicon for cryptocurrency markets taking into account the jargon and colloquialisms that arise in cryptocurrency text that is unique to this domain. We will construct the target regression time-series $\{Y_t\}$ using our novel sentiment extraction framework, described below. Then, we will utilise both BERT and VADER to construct the Instrumental Variable time-series $\{\widetilde{Y}_t\}$.

Therefore, in this section we briefly introduce how to transform processed text tokens into a time-series of distributions, in the process explaining what is known in the natural language processing context as the text embedding representation. Note that we are specifically interested in producing text embeddings with the aim to incorporate them in time-series regression models. Few approaches to constructing sentiment indices readily admit a time-series construction. A recent example is [25], who construct a sentiment scoring rule based on the difference between the number of positive and negative words in Tweets, which is an approach significantly different to ours; for an in-depth description of our framework we refer the interested reader to [26] where we employ the sentiment time-series construction to COVID-19 sentiment analysis, and [27] where we study statistical causality in crypto markets.

4.1 Distributional Sequential Text Data Embedding for Response Time-Series $\{Y_t\}$

The embedding framework we construct is based on the well-known *bag-of-words model* (BoW), which is commonly applied in natural language processing (NLP) and information retrieval [28]. The idea behind BoW in NLP is to represent a segment of text as a collection ("bag") of words without considering the order in which they appear in the text. Instead, we are now setting BoW into a time-series context, and present a novel online formulation that allows us to incorporate the text-based sentiment index into a time-series system. Furthermore, in this way we avoid the computational limitations of BoW, stemming from having to manipulate large sparse matrices whose size depends on the number of distinct document words and corpus size, and may well be in the order of hundreds of thousands.

We begin by introducing some basic notation: t denotes a "token", i.e. a linguistic unit of one or more characters (a word, a number, a punctuation character etc), $\mathcal{V}$ is the *vocabulary*, namely a finite set of tokens that is valid in the language, and $\mathcal{D}$ is a *dictionary* ($\mathcal{D} \subseteq \mathcal{V}$), i.e. a finite set of tokens, which we consider adequate to express the topic under study. We will work with n-grams, where n denotes the number of tokens in the text processing unit we consider, namely a set of n consecutive terms.

The time-series embedding is defined by the 3-ary relation $\mathcal{R} \subseteq \mathcal{V} \times \mathbb{D} \times \tilde{\mathcal{N}}$, where $\mathbb{D} = \{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^p\}$, $\mathcal{D}^j \subseteq \mathcal{V}$ is a set of dictionaries each of size q_j, and $\tilde{\mathcal{N}} = \{\mathbb{N}^{q_1}, \mathbb{N}^{q_2}, \ldots, \mathbb{N}^{q_p}\}$. To compute the members of $\tilde{\mathcal{N}}$ for each element of $\mathcal{R}$ we use the following equation, which defines $\mathcal{R}$:

$$\hat{\gamma}_N^{j,l}\left(\tilde{\nu}_N, \mathcal{D}^{j,l}\right) = \begin{cases} \frac{m_N^{j,l}}{n \times N}, & r_m\left(\tilde{\nu}_N, \mathcal{D}^{j,l}\right) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

where $\tilde{\nu}_N = \{\tilde{\nu}_{wN}\}_{w=1:n}$, $m_N^{j,l} = |\{\nu' : \nu' \in \tilde{\nu}_N\} \cap \{\mathcal{D}^{j,l}\}|$, and $\mathcal{D}^{j,l}$ denotes a dictionary token $l \in \{1, \ldots, q_j\}$, for dictionary $j \in \{1, \ldots, p\}$, and

$$r_m\left(\tilde{\nu}_N, \mathcal{D}^{j,l}\right) = \begin{cases} 1, & m_N^{j,l} \geq m_{min} \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

where N is the index of the current timestep, in n-gram "time" which indexes n-grams in our setting. Therefore, at each N we have a vector of dimension q_j which is the embedding of the n-gram at N. In this construction, the condition in Equation 29 restricts the count of any token of $\mathcal{D}^j$ which is in n-gram $\nu_{1N}, \ldots, \nu_{nN}$ at timestep N to be at least m_{min}.

In order to capture the time-dependent nature of text, we note that the total number of observed tokens increases as we shift the n-gram towards the end of the text. Therefore, we want to recursively extract proportions of the dictionary tokens within the n-gram at time N. To account for this effect we apply the following transformation at each N:

$$\tilde{\hat{\gamma}}_N^{j,l}(\cdot) = \begin{cases} \frac{\sum_{i=1}^{N-1} m_i^{j,l} + m_N^{j,l}}{M_N}, & r_m(\cdot) = 1 \\ 0, & \text{otherwise,} \end{cases} \tag{30}$$

where $m_N^{j,l}$ is the count of token l in dictionary $\mathcal{D}^j$ at timestep N, and M_N is the total count of tokens we have observed up to timestep N which satisfy $r_m(\cdot) = 1$.

It is important to point out at this stage that the support of the distribution of proportions is restricted by the condition in Equation 29. Tokens with count less than m_{min} will be excluded from M_N, and consequently the support of the distribution. To construct the time-series for the current study, we set $n = 20$ and $m_{min} = 1$.

4.2 Converting Sequential Text Embedding to Sentiment Index Time-Series $\{Y_t\}$

The final stage of the construction comprises mapping this time-series of distributions onto a scalar summary to create a sequence of summary statistics that will define the sentiment index time-series.

Using the embedding extracted from token occurrences, we construct additional time-series using properties of the empirical distribution of the embedded text. We acquire the density of the token proportions of Equation 30:

$$g_N^{j,l}(\tilde{\nu}_N, \mathcal{D}^{j,l}) = \frac{\mathbb{I}^{j,l}(\tilde{\nu}_N)\tilde{\hat{\gamma}}_N^{j,l}(\tilde{\nu}_N, \mathcal{D}^{j,l})}{\sum_{l=1}^{q_j} \mathbb{I}^{j,l}(\tilde{\nu}_N)\tilde{\hat{\gamma}}_N^{j,l}(\tilde{\nu}_N, \mathcal{D}^{j,l})} \tag{31}$$

where, as before, $\tilde{\nu}_N$ denotes the n-gram at time-step N, and the indicator function $\mathbb{I}^{j,l}(\tilde{\nu}_N)$ selects the n-gram terms:

$$\mathbb{I}^{j,l}(\tilde{\nu}_N) = \mathbb{I}(\tilde{\nu}_N, \mathcal{D}^{j,l}) = \begin{cases} 1, & \text{if } l \in \{l' \,:\, \mathcal{D}^{j,l'} \in \tilde{\nu}_N \text{ for some } w\} \\ 0, & \text{otherwise} \end{cases}, \tag{32}$$

and then we can effectively study the density itself, that changes per n-gram, or use a suitable summary of it.

We expect that the frequency with which words are used in the course of the text, as well as the richness of the dictionary, will reflect on the value of the entropy of the empirical distribution of proportions, which we use to construct our time-series. The entropy is a vector-valued process of dimension p, $\boldsymbol{H}_N = [H_N^{(1)}, \ldots, H_N^{(p)}]$, whose marginal component that corresponds to the j^{th} dictionary is given , for $j = 1, \ldots, p$, by:

$$H_N^{(j)}\left(\tilde{\nu}_N\right) \Big| \left\{g_N^{j,l}(\cdot)\right\}_{l=1:q_j} = \begin{cases} -\sum_{l=1}^{q_j} \mathbb{I}^{j,l}(\tilde{\nu}_N) g_N^{j,l} \ln(g_N^{j,l}(\cdot)), & \exists\, l \text{ s.t. } g_N^{j,l}(\tilde{\nu}_N, \mathcal{D}^{j,l}) \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

Using this framework, we construct the daily sentiment index per news source for positive and negative polarities. We then provide the robust median of the sentiment per day in each polarity to produce a robust, polarity-based collection of daily sentiment indices in which to study relationships between retail sentiment and price dynamics.

Formally, our daily Entropy Sentiment index is constructed as follows [27]:

$$\text{entropy\_index}(\tau) = \text{median}(H_{i,\tau}, \ldots, H_{i+k,\tau}), \tag{34}$$

where $H_{i,\tau}, \ldots, H_{i+k,\tau}$ are the entropy values of the text segments $i, \ldots, i+k$, coming from articles written on the same calendar day τ.

4.2.1 Reference dictionary

The previous description of our framework for the construction of a lexicon-based sentiment index makes evident the requirement of an expressive dictionary (lexicon) of English words that will be purpose-built for the crypto space, as well as a collection of crypto-specific words annotated with sentiment information for that space - positive, negative or neutral words. The lexicon will be the basis upon which all text tokens are related.

To construct the dictionary, often people collect the most frequent tokens present in the corpus of documents that is available for training and evaluation of their model. However, we argue that this approach significantly restricts the representational power of the dictionary. In contrast, we treated the construction of the dictionary as a separate task. We collected a general English dictionary, as well as a number of dictionaries covering different topics, including Engineering and Technology, Media, Business, Economics, Finance, Mathematics, and Computing, all of which are pertinent to the crypto space. The dictionaries were constructed by collecting words present in online dictionaries, mainly those of Oxford University. After obtaining the word lists via web scraping, we further curated them by cleaning the tokens from scraping artefacts. Finally, together with experts of the crypto community, we manually compiled a list of words that express positive, negative or neutral sentiment when used in the context of cryptocurrency markets.

4.3 Combining Multiple News Source Sentiment Time-Series: Volume-Based Weighting for Crypto Market Sentiment

If we consider the different crypto assets (Bitcoin - BTC, Ethereum - ETH) that form the focus of this study, then the news articles written about each of these assets can be considered as "topics" in an NLP text processing context. We can then consider different options for combining the sentiment time-series from these different topics across different news sources.

Let $X_\tau^{(s,j,q)}$ denote the sentiment indices where the index s refers to sentiment polarity $s \in$ {positive, negative, absolute magnitude}, the index j refers to asset $j \in$ { BTC, ETH }, $q \in$ {Cryptodaily, Cryptoslate} refers to the news source of the articles, τ is an n-gram "time" index, and $N_{s,j}$ denotes the total number of n-grams (or could also be sentences) of "topic" j, with sentiment s in all news sources. For calendar time units $t = 1, \ldots, T$ we can partition $\left\{X_\tau^{(s,j,q)}\right\}_{\tau=1}^{N_{s,j}}$ by grouping the observations that come from articles published on the same day in each news source: $\left\{X_\tau^{(s,j,q)}\right\}_{\tau=1}^{n_t^{s,j,q}}$, for $t = 1, \ldots, T$, $\sum_{t=1}^{T} \sum_q n_t^{s,j,q} = N_{s,j}$ and $n_t^{s,j,q} \geq 0$.

To capture a market wide sentiment for a given polarity $s \in$ {positive, negative, absolute magnitude} and asset j we use a volume based weighting rule:

$$X_t^{(s,j)} = \sum_q w_t^{s,j,q} \tilde{X}_t^{(s,j,q)}, \quad w_t^{s,j,q} = \frac{n_t^{s,j,q}}{\sum_q n_t^{s,j,q}}, \tag{35}$$

where $\tilde{X}_t^{(s,j,q)} = m\Big(\Big\{X_\tau^{(s,j,q)}\Big\}_{\tau=1}^{n_t^{s,j,q}}\Big)$, where $m(\cdot)$ denotes the mapping from each segment (set of n-grams) of topic j, news source q, and sentiment s that corresponds to time t. The weights are assigned according to the volume of n-grams per day for each topic, which ensures that article lengths have no effect on the weight.

In Figures 3 we plot the smoothed volume-weighted positive and negative sentiment indices as well as the index of absolute sentiment strength, for articles referring to Bitcoin.

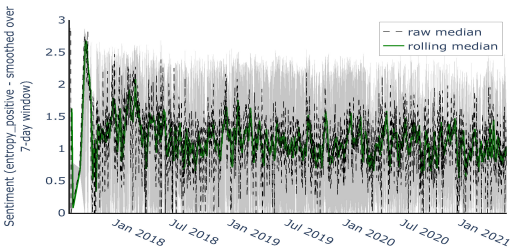
The extracted crypto sentiments for Ethereum are available in the Supplementary Appendix, Section A.

5 Estimation of ARDL-MIDAS-Transformer Long-Memory Regressions

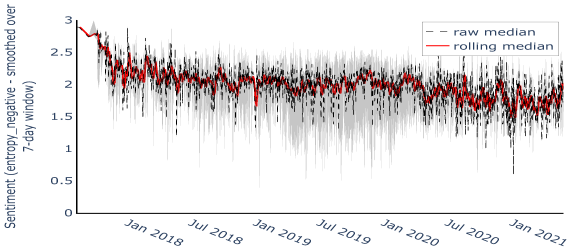
In this section we explain a simple five stage estimation procedure to fit the infinite-lag Koyck ARDL-MIDAS-Transformer Gegenbauer Long Memory model. The procedure can be used to also fit intermediate models such as Multiple-MIDAS, the ARDL-MIDAS or other variants, such as the ARDL-MIDAS-NeuralNet models for different

Figure 3: Sentiment indices with 95% confidence intervals constructed from articles about Bitcoin published on Cryptodaily (http://cryptodaily.co.uk) and Cryptoslate (http://cryptoslate.com).

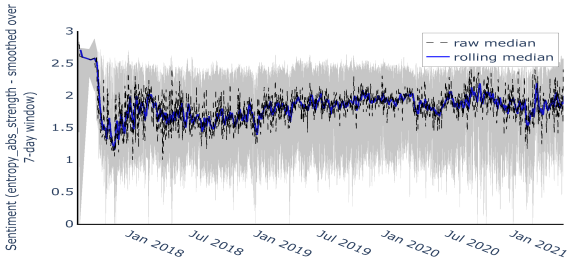
(a) Positive sentiment



(b) Negative sentiment



(c) Absolute sentiment



architectures; as we discuss this explicitly in the sentiment NLP context, we focus on Transformers for our study. The four stages proceed as follows:

- Stage 1: Crawl and scrape cryptocurrency news articles to create a corpus of crypto news for topics of relevance from each news feed identified. Munge the text articles according to a range of chosen pre-processing steps to address aspects of data cleaning and denoising, in terms of punctuation, numbering, letter casing, stemming, stopword removal, word compounds, and removal of low-frequency words. This stage produces an intra-daily time-series of n-grams (n tokens), time-stamped and ordered, which we denote by $\{\boldsymbol{S}_t^{(m)}\}$. In addition, construct a time-series of article sentences, which will be the input to the BERT and VADER models.
- Stage 2: Construct the Entropy Sentiment time-series for each news source and combine them as described in Section 4.3 to make the response time-series $\{Y_t\}$. Then, construct the instrumental variable generative embeddings $\{\widetilde{Y}_t\}$, via the following steps. First, construct the BERT- and VADER-based sentiment time-series per news source and combine them in a single index across all news sources also according to the method of Section 4.3. Second, fit the regression models of Section 6.1.1 to generate a range of alternative possible IV $\{\widetilde{Y}_t\}$.
- Stage 3: Fit the Koyck-transformed ARDL model and assess which of the different instrumental variables of the previous step at time scale t are appropriate. Perform statistical testing on the suitability of the Transformer/BERT and VADER as IV versus the crypto specific entropy sentiment signal.
- Stage 4: Using the selected IVs at time scale t, fit the ARDL-MIDAS model to learn the optimal model structure for (p, K, m), and estimate the MIDAS coefficient basis functions $B(k; \Psi)$ and the geometric decay rate for the infinite ARDL-Koyck transform γ.
- Stage 5: Fit the residuals from Stage 4 with a Range-Scale (R/S) estimation process for the Gegenbauer long memory to determine the Gegenbauer hyperbolic ACF decay parameter d and oscillation index u.

The advantage of this five-stage procedure versus a joint estimation of all components in one stage is that standard `R` and `Python` packages may be utilised to perform each stage of the estimation, which we have found to work adequately as outlined in the experimental results below.

Below we add some further details on Stages 3 - 5.

5.1 Stage 3: Estimation of ARDL(∞) Regression

Consider time-series $\{Y_t\}$ given by the daily sentiment score based on our proposed sentiment time-series construction. We will then regress this sentiment scoring method against alternative sentiment extraction methods based on BERT (`https://huggingface.co/nlptown/`, model: bert-base-multilingual-uncased-sentiment) and VADER that we will transform into time-series covariates and denote them by $\{X_t^B\}$ and $\{X_t^V\}$ also constructed from daily measures of sentiment. Then we seek to fit the regression model:

$$Y_t = \beta_0 + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \beta^B \sum_{j=0}^{+\infty} \phi_B^j X_{t-j}^B + \beta^V \sum_{j=0}^{+\infty} \phi_V^j X_{t-j}^V + \epsilon_t. \tag{36}$$

Since the estimation of the classical ARDL(∞) model via a Koyck geometric parametrisation is standard in the time-series literature, we defer the interested reader to a brief summary provided in the Supplementary Appendix, Section C. We present the analysis of this regression in the results Section 6.1.1.

5.2 Stage 4: Estimation of ARDL(∞)-MIDAS Regression

Given the modified Koyck transform applied to the ARDL(∞)-MIDAS time-series regression structure as outlined in Proposition 2.1, one can perform estimation of this model using standard MIDAS model estimation packages such as the `R` package `midasr` as detailed in [29]. This package works with MIDAS models generically specified in a vectorised form as follows:

$$\Phi_p(L) Y_t = \boldsymbol{\beta}_k(L^{1/m}) \boldsymbol{X}_t^{(m)} + \epsilon_t \tag{37}$$

Clearly, the modified Koyck Transformed ARDL-MIDAS model we proposed to develop can readily be represented in this form. Then, the package `midasr` performs a number of different model fitting structures including the U-MIDAS model which is an unrestricted variation of the MIDAS model formulation, in which a frequency alignment transformation and the estimation of the model is performed using Ordinary Least Squares (OLS), see further details in [30].

5.3 Stage 5: Estimation of Long Memory Components

We focus on the class of non-oscillatory long memory models based on the ARFIMA$(0, d, 0)$ type of long memory, i.e. we set $u = 1$, and only have to estimate d, namely the long memory exponent in the model in Proposition 2.1. We adopted this setting as the empirical ACF was not oscillatory and so we simplified the generator of the long-memory fractional difference to the ARFIMA type.

We will estimate this long memory fractional difference parameter d on the residuals of the model from Stage 3. One can then obtain an estimate for the strength of long memory d based on a Hurst exponent estimator, by first estimating the Hurst exponent H [31] and then using the relationship $d = H - 0.5$.

In this work, we adopt the Rescaled Range R/S Hurst exponent estimator that measures the intensity of long-range dependence in a time-series and was originally developed by [31]. Given a time series $\boldsymbol{Y}_{t \in \{1,2,3,\cdots,T\}}$, the sample mean and the standard deviation process are given by

$$\overline{Y}_T = \frac{1}{T} \sum_{j=1}^{T} Y_j \;\text{ and }\; S_t = \sqrt{\frac{1}{t-1} \sum_{j=1}^{t} (X_j)^2}, \tag{38}$$

where $X_t = Y_t - \overline{Y}_T$ is the mean-adjusted series. Then a cumulative sum series is given by $Z_t = \sum_{j=1}^{t} X_j$ and the cumulative range based on these sums is

$$R_t = \text{Max}\left(0, Z_1, \cdots, Z_t\right) - \text{Min}\left(0, Z_1, \cdots, Z_t\right). \tag{39}$$

The following proposition describes the estimator of H as derived in [32].

Proposition 5.1. *Consider a time-series $Y_t \in \mathbb{R}$ and define S_t and R_t in Equations (38) and (39) respectively, then $\exists\ C \in \mathbb{R}$ such that the following asymptotic property of the Rescaled Range R/S holds*

$$[R/S](T) = \frac{1}{T} \sum_{t=1}^{T} R_t/S_t \sim C T^H, \;\; \textit{as} \;\; T \to \infty.$$

In addition, for small sample size T, the Rescaled Range R/S can be adjusted with the following formula [33]:

$$\mathbb{E}[R/S(T)] = \begin{cases} \frac{T-1/2}{T} \frac{\Gamma((T-1)/2)}{\sqrt{\pi}(T/2)} \sum_{j=1}^{T-1} \sqrt{\frac{T-j}{j}}, & \text{for } T \leq 340 \\ \\ \frac{T-1/2}{T} \frac{1}{\sqrt{T\pi/2}} \sum_{j=1}^{T-1} \sqrt{\frac{T-j}{j}}, & \text{for } T > 340 \end{cases},$$

where the $\frac{T-1/2}{T}$ term was added by [34]. The estimate of H can then be obtained by a simple linear regression

$$\log\Big((R/S(T) - \mathbb{E}[R/S(T)]\Big) = \log C + H \log T.$$

Hence, we have the following definition for $\hat{H}$, the estimator of H, based on the unadjusted Rescaled Range R/S analysis and given by:

$$\widehat{H}_{R/S} = \frac{T(\sum_{t=1}^{T} \log R/S(t) \log t) - (\sum_{t=1}^{T} \log R/S(t))(\sum_{t=1}^{T} \log t)}{T(\sum_{t=1}^{T} (\log t)^2) - (\sum_{t=1}^{T} \log t)^2}. \tag{40}$$

The empirical confidence interval of $\widehat{H}$ given in Equation (40) with sample size $T = 2^N$ [35] is

$$(0.5 - \exp(-7.33 \log(\log N) + 4.21), \exp(-7.20 \log(\log N) + 4.04) + 0.5).$$

Note that no asymptotic distribution theory has been derived for the estimated Hurst parameter H for R/S analysis, however, we can apply bootstrap methods to find related properties to test for statistical significance of the estimates in order to detect the long memory properties.

6 Results and Discussion

In this section we will present two real data case studies. The first is to illustrate that the sentiment time-series index constructed in this work is distinct in its information content compared to those extracted either from deep neural network solutions obtained via pre-trained non fine-tuned applications of BERT, or from rule-based systems like VADER. We demonstrate that both of these systems can be combined together to produce a viable time-series index for the design of an effective Instrumental Variable to act as a proxy for the proposed entropy sentiment signal when fitting ARDL(∞)-MIDAS models via OLS, which would otherwise produce biased estimators.

Note that fine-tuning of Transformer-based models and re-construction of domain-specific rule-based systems for sentiment extraction is particularly difficult in this study context which is a "small data" problem relative to typical sentiment studies. The reason for this is that the number of crypto articles available is relatively small compared to the size of corpora used to train deep neural network architectures or inform decisions about potential rule sets that would generalise well. This is one of the key motivations we see for our proposed method, in that it is directly interpretable and applicable in relatively small data contexts such as the case study in question. Therefore, in the first case study we demonstrate that our proposed crypto sentiment index contains significantly different information, compared to the typical approach of just applying BERT and VADER as a black-box package without tailoring or fine-tuning. We demonstrate the value of our sentiment time-series index through an ARDL regression example to show that the covariates of the competing sentiment methods are not strongly expressive of the variation in our daily sentiment signal.

In the second case study, we treat our daily sentiment index as the target response time-series and we seek to explore changes in daily sentiment for cryptocurrency markets in terms of intra-daily crypto price fluctuations and technology factor variations. This will be meaningful for both interpretation of sentiment and price discovery as well as forecasting sentiment at the end of the day, given observations of current intra-daily price and technology network factors. We fit the sequence of infinite lag Koyck-transformed ARDL-MIDAS and ARDL-MIDAS-Transformer Gegenbauer long-memory models to undertake this second case study, exploring along the way each component of the model.

6.1 Case study I: ARDL structure of Y_t and explanatory power of BERT and VADER sentiment methods for Instrumental Variable Construction of $\widetilde{Y}_t$

Let $\{s_{1,\tau_1}, s_{2,\tau_2}, \ldots, s_{N,\tau_N}\}$ be the collection of sentences from all articles, ordered according to article publication date τ_i and order of appearance in the article.

Valence Aware Dictionary for sEntiment Reasoning (VADER, [20]) is a rule-based sentiment model derived from human annotation of online texts. In VADER, first a gold-standard sentiment dictionary is extracted, then validated

using qualitative methods (based on human annotation) and lexical features are extracted together with five rules that incorporate grammatical or syntactical conventions that people use to express sentiment intensity.

In our setting, to construct the sentiment index from VADER we use a daily median filter:

$$\text{VADER\_index}(\tau) = \text{median}\Big(\text{VADER}(s_{i,\tau}), \ldots, \text{VADER}(s_{i+k,\tau})\Big) \in [-1, 1], \tag{41}$$

where $\{s_{i,\tau}, \ldots, s_{i+k,\tau}\}$ are sentences from articles written on the same calendar day τ and $\text{VADER}(\cdot)$ returns the output of the VADER sentiment model.

Naturally, a challenge with this method arises in the context we consider as there is domain-specific knowledge and terminology in the cryptocurrency context that is not adequately captured by the standard formulation of VADER. Nevertheless, one may find examples of the use of the VADER sentiment model in the crypto market, e.g. [36, 37, 38]. These studies significantly differ from our study in three main aspects: first, the type of sentiment model utilised; second, the type and quality of data used to produce the sentiment model; and third, the way in which sentiment is analysed or utilised. We specifically have not concentrated on social media sentiment, which VADER claims to extract, because of data quality challenges stemming from the short, mainly informal nature of social media text. We have instead focused on public news articles from community accepted reliable websites, which undergo editorial processing before publication, and have developed specific sentiment indices able to capture the particular nature of the vocabulary used in the domain due to our purpose-built crypto dictionary.

Furthermore, alternative methods to rule-based approaches include the word embedding based models. Word embeddings are real-valued high dimensional vectors that correspond to specific words, and are obtained via a complex non-linear optimisation process. This approach has been prevalent in the neural network-based NLP paradigm, and the optimisation process that obtains the embeddings aims to either learn a decomposition of the document-term matrix of a corpus of documents (e.g. GloVe [39]), or minimise an entropy measure ('perplexity') for a model that predicts the word that follows a given word sequence ('language modelling'), e.g. the Transformer-based BERT [40] that we also utilise in this work).

For BERT, we used a pre-trained model based on an implementation from Hugging Face (`https://huggingface.co/nlptown/`, model: bert-base-multilingual-uncased-sentiment), a group well-known in the NLP community for code quality. The model has been pretrained on a corpus of product reviews, yet we did not fine-tune, i.e. further train, the model with data from our domain for two reasons: first, to our knowledge there are no datasets of crypto-related public articles that have been annotated with sentiment labels, and second, we did not want to undertake this task as it would require manually annotating more than $3,000$ articles, which places such a process out of our research scope. On the contrary, we want to illustrate: i) that our approach needs a lot less annotating, i.e. only for the domain dictionary construction, ii) contrary to computationally expensive neural models which may capture an unclear concept of sentiment, our method is efficient and offers interpretable and informative results, and iii) the problem of domain mismatch and lack of generalisation of such models in specialised areas, despite the fact that generalisation is one of the main arguments in their favour.

The selected BERT model returns a categorical sentiment score of five levels (0-4), corresponding to the "star rating" of a review: 0 for very negative and 4 for very positive. We again used a daily median filter to construct the sentiment index based on BERT:

$$\text{BERT\_index}(\tau) = \text{median}(\text{BERT}(s_{i,\tau}), \ldots, \text{BERT}(s_{i+k,\tau})) \in \{0, 1, 2, 3, 4\}, \tag{42}$$

where $\{s_{i,\tau}, \ldots, s_{i+k,\tau}\}$ are sentences from articles written on the same calendar day τ and $\text{BERT}(\cdot)$ is the output of the BERT model on a sentence.

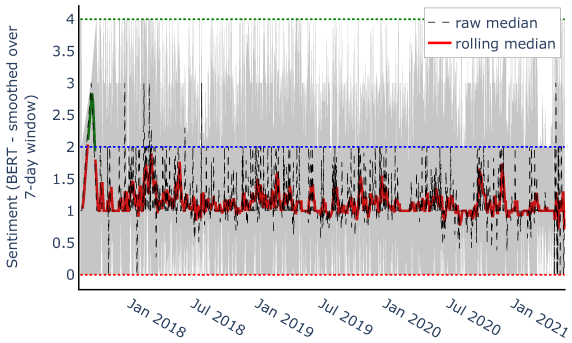
As discussed, in the current case study, in addition to the custom entropy sentiment time-series, we also will be employing sentiment time-series constructed using the BERT and VADER models. The time-series for BERT and VADER will be constructed following the same procedure of Section 4.3, where instead of n-grams, we will be assembling sentences from articles on the same day. We plot the sentiment indices constructed based on BERT and VADER in Figures 4 for BTC, where we indicate polarity with a different choice of color, namely green for positive and red for negative. The corresponding plots for ETH are in the Supplementary Appendix, Section B.

6.1.1 Analysis of Entropy Based Sentiment Time-Series versus Instrumental Variable Construction from BERT and VADER Indices

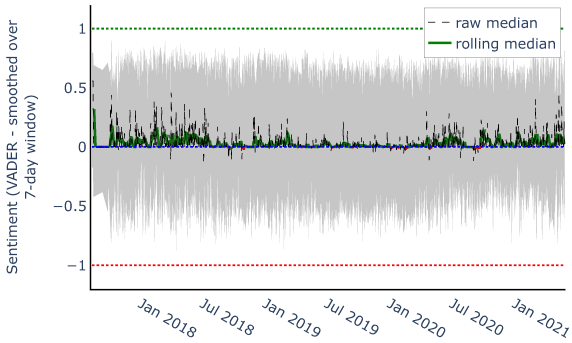
In this section we investigate whether the sentiment indices constructed from the BERT and VADER models can be used as explanatory variables for our entropy sentiment index. For this purpose, we fit distributed lag time-series regression models where the BERT and VADER indices are in the set of regressors, whereas the dependent variable is each of the following Entropy Sentiment indices: absolute sentiment strength, negative sentiment and positive

Figure 4: Sentiment indices based on BERT and VADER with 95% confidence intervals constructed from articles about Bitcoin published on Cryptodaily (http://cryptodaily.co.uk) and Cryptoslate (http://cryptoslate.com)

(a) BERT



(b) VADER



sentiment. Subsequently, the significance of the model parameters relevant to the BERT and VADER covariates is assessed. The model structure we adopt is the following:

$$Y_t = \beta_0 + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \sum_{j=0}^{+\infty} \vec{\beta}_j^T \vec{X}_{t-j} + \epsilon_t, \tag{43}$$

where $\epsilon_t \sim N(0, \sigma^2)$, $\vec{\beta}_j = [\beta_j^B \quad \beta_j^V]^T$, $\vec{X}_t = [X_t^B \quad X_t^V]^T$ and the superscripts B, V stand for BERT and VADER respectively.

We draw attention to the fact that our goal with this model structure is not to develop a predictive model for our sentiment index Y_t but rather to investigate if the covariates from BERT and VADER have any explanatory power for Y_t, given our novel Entropy Sentiment time-series model. We add an autoregressive component in the covariates to account for potential serial dependence in the dependent variable. If we do not account for that, we risk being mislead by the output of the regression: if Y_t has serial dependence then the BERT and VADER covariates can appear to be significant for the dependence structure but that would not mean that they are explanatory for Y_t - it would be an artefact of the chosen model structure.

In order to obtain a parsimonious representation of the model and thus reduce the number of parameters we have to estimate, we need to find a suitable functional expression for the coefficients $\vec{\beta}$. As discussed in more detail in Section 2.1, $\vec{\beta}_j$ must form an L_2 sequence so that the sum of the corresponding term is square summable (provided X_t have finite moments) and the process converges, and one option for the functional expression of $\vec{\beta}_j$ is to make them a geometric sequence [41]. This is convenient as we can use the generator form of such sequences which has only two parameters to estimate:

$$\begin{aligned} \beta_j^B &= \beta^B \phi_B^j, \quad 0 < \phi_B < 1 \\ \beta_j^V &= \beta^V \phi_V^j, \quad 0 < \phi_V < 1 \end{aligned} \tag{44}$$

for each covariate correspondingly. The final form of our model then becomes:

$$\begin{aligned} Y_t &= \alpha + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \sum_{j=0}^{+\infty} \beta_j^B X_{t-j}^B + \sum_{j=0}^{+\infty} \beta_j^V X_{t-j}^V + \epsilon_t \\ &= \alpha + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \beta^B \sum_{j=0}^{+\infty} \phi_B^j X_{t-j}^B + \beta^V \sum_{j=0}^{+\infty} \phi_V^j X_{t-j}^V + \epsilon_t \end{aligned} \tag{45}$$

If our analysis using this model shows that Y_t is conditionally independent, or only weakly dependent on the covariates of BERT and VADER, then our sentiment index captures different information to these alternative models. In that case, it may be valuable, for example, to consider all indices together as components in a multimodal sentiment index

model, where modality would correspond to the source, namely underlying model, of each component that captures a different sentiment aspect. Or as we demonstrate in Case Study II, one can use the BERT and VADER sentiment signals combined as an Instrumental Variable for estimation of a model using our Entropy Sentiment signal that has infinite lag structure transformed by a Koyck transform method.

We fit the model in Equation 45 in rolling windows with a length of three months with an one-month overlap, where the regressors are the average-smoothed, z-scaled BERT and VADER indices.

Model I: BERT covariate for Y_t, IV regression with BERT

We construct the IV as follows:

$$\tilde{E^y}_{t-1} = \tilde{\mu}_0 + \tilde{\mu}_1 X^B_{t-1} + \epsilon_{t-1}. \tag{46}$$

We perform a t-test on the regression parameters and compute the IV $\tilde{E^y}_{t-1}$ using only those that are statistically significant. Then we conduct the following OLS regression for E^y_t:

$$E^y_t = \alpha(1-\phi) + \phi\tilde{E^y}_{t-1} + \beta^B X^B_t + \epsilon_t - \phi\epsilon_{t-1}. \tag{47}$$

At this stage, we want first to evaluate the quality of the IV and then the quality of the fit with respect to X^B. To evaluate the instrumental variable we test for autocorrelation in the error terms; if they are not autocorrelated then we have successfully constructed an IV that is not correlated with the errors and the use of OLS was appropriate. We test for error autocorrelation using the Breusch-Godfrey test (LM test for autocorrelation, [42]), for which the null hypothesis is the lack of serial correlation of any order up to p. If we have evidence to accept the null then we proceed to test the parameters for statistical significance, otherwise we consider this fit invalid - we would need to construct a different IV, for example by adding more structure to the corresponding model. If they are significant, we next assess the model by means of the AIC.

The procedure just described for Model I is followed for all subsequent models, therefore we will next present only the different model structures we used.

Model II: VADER covariate for Y_t, IV regression with VADER

IV regression:

$$\tilde{E^y}_{t-1} = \tilde{\mu}_0 + \tilde{\mu}_1 X^V_{t-1} + \epsilon_{t-1}. \tag{48}$$

E^y_t regression:

$$E^y_t = \alpha(1-\phi) + \phi\tilde{E^y}_{t-1} + \beta^V X^V_t + \epsilon_t - \phi\epsilon_{t-1}. \tag{49}$$

Model III: BERT and VADER covariates for Y_t, IV regression with BERT

IV regression:

$$\tilde{E^y}_{t-1} = \tilde{\mu}_0 + \tilde{\mu}_1 X^B_{t-1} + \epsilon_{t-1}. \tag{50}$$

E^y_t regression:

$$E^y_t = \alpha(1-\phi) + \phi\tilde{E^y}_{t-1} + \beta^B X^B_t + \beta^V X^V_t + \epsilon_t - \phi\epsilon_{t-1}. \tag{51}$$

Model IV: BERT and VADER covariates for Y_t, IV regression with VADER

IV regression:

$$\tilde{E^y}_{t-1} = \tilde{\mu}_0 + \tilde{\mu}_1 X^V_{t-1} + \epsilon_{t-1}. \tag{52}$$

E^y_t regression:

$$E^y_t = \alpha(1-\phi) + \phi\tilde{E^y}_{t-1} + \beta^B X^B_t + \beta^V X^V_t + \epsilon_t - \phi\epsilon_{t-1}. \tag{53}$$

Model V: BERT and VADER covariates for Y_t, IV regression with BERT and VADER

IV regression:

$$\tilde{E^y}_{t-1} = \tilde{\mu}_0 + \tilde{\mu}_1 X^B_{t-1} + \tilde{\mu}_2 X^V_{t-1} + \epsilon_{t-1}. \tag{54}$$

E^y_t regression:

$$E^y_t = \alpha(1-\phi) + \phi \tilde{E^y}_{t-1} + \beta^B X^B_t + \beta^V X^V_t + \epsilon_t - \phi\epsilon_{t-1}. \tag{55}$$

Finally, we compare the AIC results for the appropriate models and choose the best among those.

6.1.2 Stage 2 - Model Selection and Calibration

We begin with an analysis of the fitting process for Stage 2 of Section 5. For fitting the Stage 2 model we performed a sequence of fits in rolling windows of seven months, overlapping by one month. Before each sequence of fits, we considered a range of different settings for the response sentiment signal and the model structure:

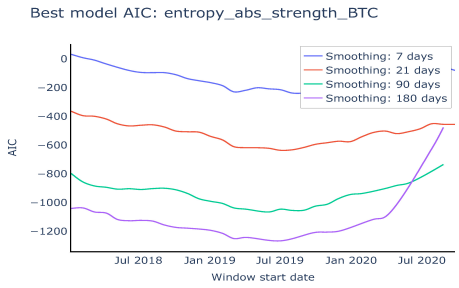
1. we considered different smoothing options for the sentiment response; we applied a rolling median filter with window sizes of 1 week, 3 weeks, 3 months, and 6 months;
2. we considered a range of lag structures for the autoregressive covariates, with lag order p $= 1, \ldots, 5$.

We demonstrate here the results for the total sentiment - the remaining equivalent results for polarity of positive and negative sentiment are available in Supplementary Appendix, Section C.1.

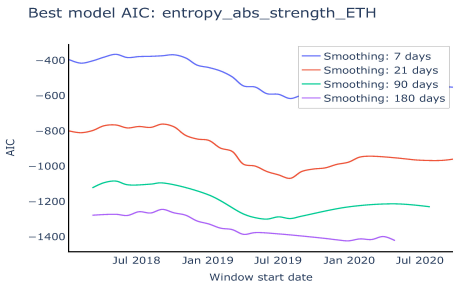
For each configuration, we stored the successfully fitted models and ranked them according to the AIC score. In Figure 5, we see the AIC of the best fitting model per window for all smoothing parametrisations. Informed by these traces, we focus on the best performing model to continue our analysis. The next question we addressed was whether we would need to have a different lag structure per window to capture the sentiment signal's variability. For this purpose, we plot the lag order per fitting window for the best fitting model (Figure 6, left panels), and in addition, we plot the lag order per fitting window for the best fitting model after dropping the coefficients that were not statistically significant to any level up to 90% (Figure 6 right panels). We also plot for comparison (in red trace) the worst performing model according to the AIC. We observe that the differences in model selection are significant when we ignore the coefficient significance level, and therefore we were correct in utilising a different lag structure per fitting window. When we also account for the significance level, we observe that for the absolute sentiment magnitude we still benefit from adapting the lag structure per window.

Figure 5: Stage I rolling window fits: best AR(p) model AIC score.

(a) BTC - abs. sentiment magnitude



(b) ETH - abs. sentiment magnitude

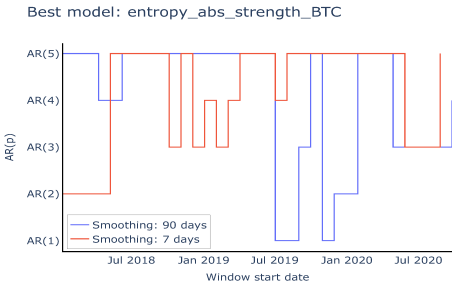


6.1.3 Stage 3 - Evaluation of Model Fit and Quality of Instrumental Variable Design

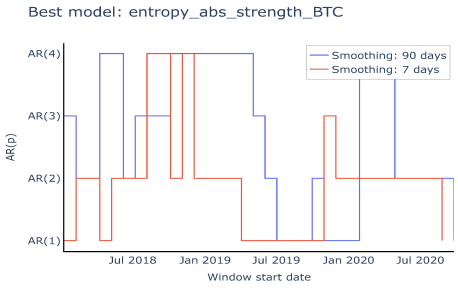
Having selected the best Stage 2 model per signal window, we then evaluate the quality of the fitting by performing a series of statistical tests on the residuals. Furthermore, with this analysis we aim to verify our hypothesis of normality of the error distribution. We employ the widely used Kwiatkowski–Phillips–Schmidt–Shin (KPSS, [43]), the Kolmogorov-Smirnov (KS, [44]), and the Vasicek-Song (VS, [45, 46, 47]) tests to investigate trend stationarity and normality of the error distribution. Table 1 shows the percentage of fitted windows that did not produce evidence to reject the null hypothesis of the tests. The null hypothesis of the KPSS test is that the signal is trend stationary, whereas the null hypothesis of the KS, Vasicek and Song tests is that the tested signal follows a normal distribution. In our

Figure 6: Stage I rolling window fits: best AR lag structure.

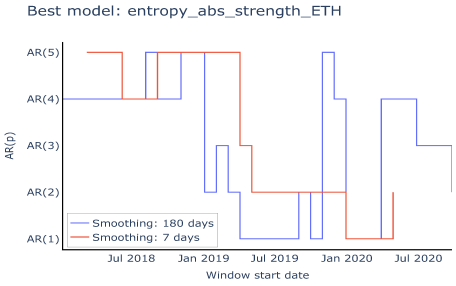
(a) BTC - abs. sentiment magnitude



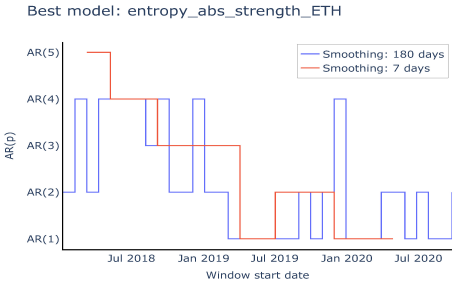
(b) BTC - abs. sentiment magnitude - $\geq 90\%$ coef. significance



(c) ETH - abs. sentiment magnitude



(d) ETH - abs. sentiment magnitude - $\geq 90\%$ coef. significance



experiments, we approximated the distribution of the null hypothesis of the Vasicek test via Monte Carlo sampling, which is the basic formulation of this test.

From Table 1 we first note that the windowed signal is trend stationary. Furthermore, with regards to the error distribution, we observe that the KS test always rejects the null hypothesis of normality, but the Vasicek entropy test shows that for the majority of the fits the error normality assumption holds. This ambiguity is resolved if we consider that the KS test places equal weight on the median tendency of the error distribution and its tails. Therefore, if there is even a little skewness or kurtosis in the distribution, it may reject the null of normal distribution whilst this may not be true. This is the effect we see here, where the Vasicek test does not reject the null hypothesis as it places higher importance to the median tendency of the distribution.

To illustrate this point further and justify our choice for normal errors, we first construct the Q-Q plots of the residuals in a window of the absolute sentiment magnitude for ETH, which we see in Table 1 that exhibits the lowest compliance with the normal error assumption. Figure 7 shows the Q-Q plots obtained for normal and t-Student distributions. We observe that there is evidence of heavier tails in the residuals than a normal distribution, and therefore we fit t-Student distributions of differing degrees of freedom to reflect different strengths of kurtosis that may be present; we illustrate this for three settings $3, 10$, and 15 degrees of freedom. In so doing, we note that the best fit is achieved for t distributions with many degrees of freedom ($t = 15$), which is close to a normal, and therefore the assumption of normal errors is adequate for our purposes. Next, in Figure 8 we plot the density estimate of the residuals in the same window, and in Figure 9 we plot the residuals against the fitted response to verify the absence of any uncaptured trend structure. Both the shape of the density estimates and the residual against fitted values scatterplot verify our assumption.

6.1.4 Stage 3 - Quantitative Evaluation of the Explanatory Power of the BERT and VADER Indices for Instrumental Variable Design

Our goal is to quantify the explanatory power of the BERT and VADER covariates for the sentiment response, after we have obtained an appropriate instrumental variable. Table 2 shows the results of the rolling window regressions for a selection of windows in which we have obtained an instrumental variable with one of the explored model structures. The quality of the instrumental variable was measured with the Breusch-Godfrey test [42], and only windows that showed decorrelated errors (95% significance levels of the coefficients in the IV regression) were kept for the regression against sentiment.

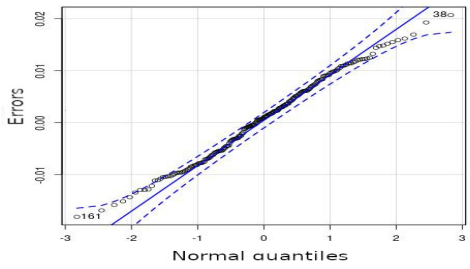
Table 1: Percentage of windows that fail to reject the null hypothesis of the performed tests.

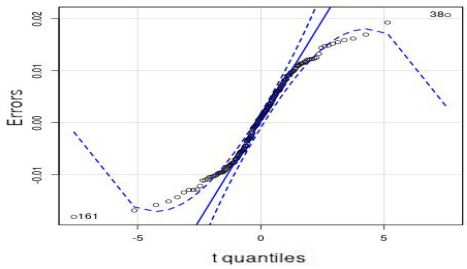
Test	Entropy Abs. strength - BTC	Entropy Negative - BTC	Entropy Positive - BTC	Entropy Abs. strength - ETH	Entropy Negative - ETH	Entropy Positive - ETH
1-week smoothed signals						
KS	0.00	0.00	0.00	0.00	0.00	0.00
VS	100.00	96.67	100.00	93.33	97.06	94.12
KPSS	100.00	100.00	100.00	100.00	100.00	100.00
3-week smoothed signals						
KS	0	0	0	0	0	0
VS	91.18	97.06	97.06	93.75	90.91	97.06
KPSS	100	100	100	100	100	100
3-month smoothed signals						
KS	0	0	0	0	0	0
VS	96.55	92.86	93.1	100	91.3	100
KPSS	100	100	100	100	100	100
6-month smoothed signals						
KS	0	0	0	0	0	0
VS	95.83	92.31	100	77.27	100	86.21
KPSS	100	100	100	100	100	100

Figure 7: Stage I residual distribution example: 6-month window starting on 2020-04-27 for ETH absolute sentiment strength.

(a) Normal errors

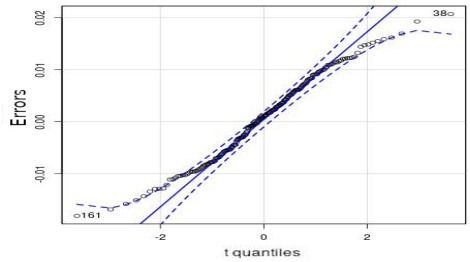
(b) t-Student errors, $t = 3$





(c) t-Student errors, $t = 10$

(d) t-Student errors, $t = 15$



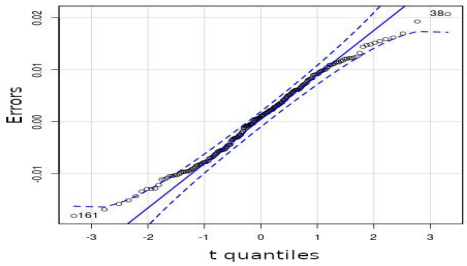
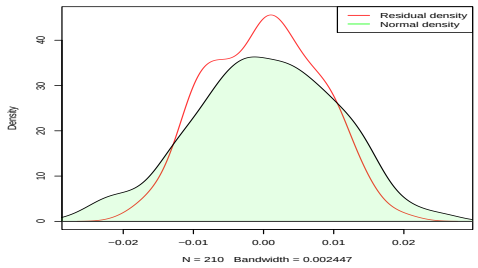


Figure 8: Stage I residual distribution density example: 6-month window starting on 2020-04-27 for ETH absolute sentiment strength. A normal density is illustrated in green and the residual density is plotted in red.

(a) Gaussian kernel density estimate



(b) Residuals violin plot

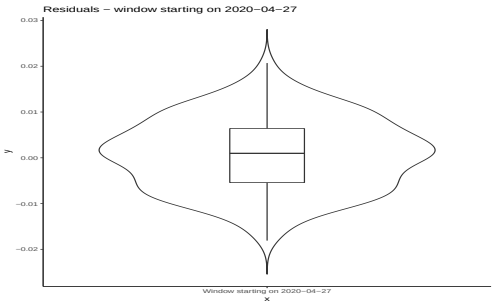
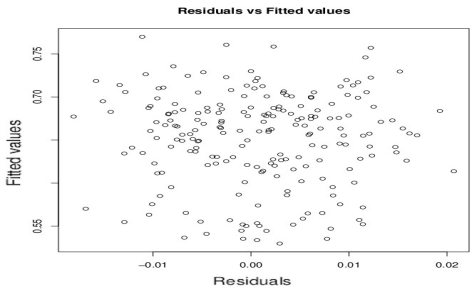


Figure 9: Stage I residuals vs fitted response values: 6-month window starting on 2020-04-27 for ETH absolute sentiment strength.



First, we observe that there were very few periods over a course of almost four years that the BERT and VADER models could explain part of our entropy sentiment index, for all polarities universally. Secondly, we note that the model structures most suitable for the IV regression were M3 and M5, where we used both covariates for the sentiment response, and either BERT only (M3) or BERT and VADER (M5) for the IV regression. Thirdly, it is interesting to see that BERT and VADER are more explanatory for BTC news sentiment rather than ETH, which may be attributed to Bitcoin's popularity leading to simpler, less technical and therefore easier to understand language in the relevant articles. Fourthly, we observe that there are specific periods where the covariate significance was high ($\geq$ 99%): those starting March 3, 2018, April 8 2018, and 27 February 2020. March-October 2018 was a period of relatively low volatility in the Bitcoin price which culminated in Winter 2018 and was sparking a lot of concern among retail investors at the time for fear of a price plunge. In Figure 4, we see that BERT is negative all of the time during that period, and VADER shows some of its few negative sentiment indications at that time. Therefore, we would expect that both indices would help explain more of the negative sentiment content of our index at that time, as we see that is the case. This also applies to the 7-month window starting April 8 2018 and extending to November of the same year, when the lack of volatility was more pronounced. The same, but less pronounced and at a higher price level, phenomenon is also observed for the window starting 27 February 2020 and extending to September 2020.

Both of these observations explain why the BERT and VADER covariates appear to have some explanatory power for our negative sentiment index for BTC. However, it is still evident that our index captures different sentiment content than the state-of-the-art alternative approaches, which we will also demonstrate qualitatively next, before proceeding to show how to leverage our sentiment index in the crypto space.

6.1.5 Stage 3 - Qualitative evidence for the difference between the entropy index and the BERT and VADER indices

In the previous section we statistically proved and quantified the difference in the content of the entropy-based sentiment index versus the indices based on BERT and VADER. In this section we provide visual evidence that illustrates the difference between the indices. In Figure 10 we show the cumulative sentiment content of the different indices for BTC and ETH, each relativised with respect to the beginning of the time period we study.

Table 2: Statistical significance levels of coefficients of BERT/VADER sentiment covariates. Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

1-week smoothed signals						
Sentiment type	**Window Start Date**	**Best IV model**	**BERT coefficient with standard errors**	**BERT significance level**	**VADER coefficient with standard errors**	**VADER significance level**
Entropy negative - BTC	2018-03-09	M5	-1.13E-01 ± 7.41E-03	**	3.1E-01 ± 2.1E-02	*
Entropy positive - BTC	2019-02-02	M3	4.91E-02 ± 2.34E-04	**	6.42E-02 ± 5.80E-04	.
Entropy positive - ETH	2018-04-08	M1	6.90E-02 ± 1.83E-03	*	-	-
Entropy positive - ETH	2018-05-08	M1	6.35E-02 ± 1.24E-03	*	-	-
3-week smoothed signals						
Sentiment type	**Window Start Date**	**Best IV model**	**BERT coefficient with standard errors**	**BERT significance level**	**VADER coefficient with standard errors**	**VADER significance level**
Entropy negative - BTC	2018-03-09	M5	-1.39E-01 ± 1.0E-04	***	4.03E-01 ± 1.91E-04	***
Entropy negative - BTC	2018-04-08	M5	-1.2E-01 ± 2.0E-04	**	3.13E-01 ± 4.38E-04	**
Entropy positive - BTC	2019-02-02	M1	3.96E-02 ± 2.07E-05	*	-	.
Entropy positive - ETH	2019-09-30	M5	6.04E-02 ± 3.34E-04	*	-2.28E-01 ± 1.28E-03	*
3-month smoothed signals						
Sentiment type	**Window Start Date**	**Best IV model**	**BERT coefficient with standard errors**	**BERT significance level**	**VADER coefficient with standard errors**	**VADER significance level**
Entropy Abs. Strength - BTC	2018-04-08	M2	-	.	2.0E-01 ± 4.71E-08	*
Entropy Abs. Strength - BTC	2018-05-08	M4	1.0E-02 ± 5.34E-06	*	3.0E-02 ± 1.11E-04	.
Entropy Abs. Strength - BTC	2020-05-27	M1	7.80E-02 ± 2.86E-07	*	-	.
Entropy Abs. Strength - BTC	2020-08-25	M3	5.98E-02 ± 1.31E-06	.	3.50E-02 ± 5.65E-07	*
Entropy negative - BTC	2020-02-27	M3	9.08E-02 ± 1.35E-07	**	3.24E-02 ± 4.68E-08	**
Entropy positive - BTC	2018-04-08	M4	1.23E-02 ± 8.1E-09	.	2.11E-01 ± 1.03E-07	*
Entropy positive - BTC	2018-05-08	M4	1.18E-02 ± 6.18E-07	*	7.75E-02 ± 1.28E-05	.
6-month smoothed signals						
Sentiment type	**Window Start Date**	**Best IV model**	**BERT coefficient with standard errors**	**BERT significance level**	**VADER coefficient with standard errors**	**VADER significance level**
Entropy Abs. Strength - BTC	2019-10-30	M2	-	.	1.13E-01 ± 7.27E-07	*
Entropy Abs. Strength - BTC	2019-11-29	M2	-	.	1.43E-01 ± 2.28E-07	**

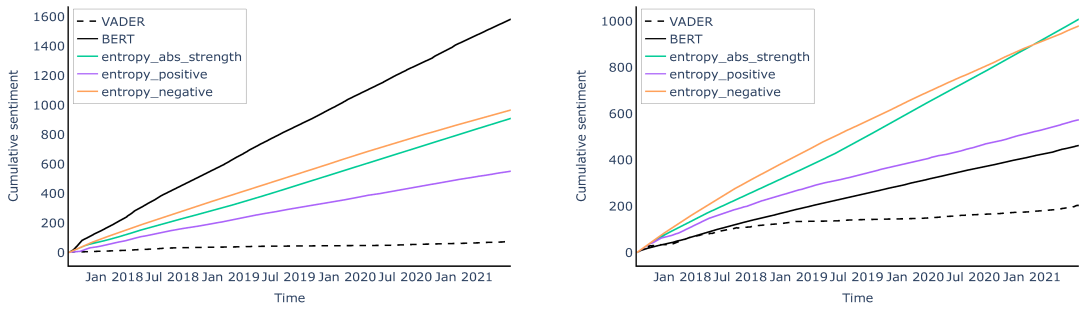
The rates of change of the traces reveal whether the aggregation of sentiment is continuous or intermittent over time, and whether it is affected by sentiment polarity or not. We observe the linear growth of cumulative sentiment, and almost constant rate of change for the most period of time, which means that there is a smooth variation in the way information is captured: a growing amount of different information is accumulated over time, as opposed to certain abrupt rare events being the main drivers of the index. The fact that the growth is linear for both assets is in part due to the fact that the information is rich, i.e. there are articles on Cryptodaily and Cryptoslate regarding BTC and ETH almost on a daily basis during the time period we focus on. Note that the linear pattern is not something dictated by any of the sentiment index models and may change if we study different news sources or assets.

From Figure 10 we understand that BERT and VADER accumulate sentiment at different rates. VADER is slower at gathering information, and in addition gathers sentiment information that is different to any of the information captured by our entropy indices. This is because its rate of change is significantly lower than the rate of the entropy indices and almost constant for the case of BTC (Figure 10, left panel). For ETH (Figure 10, right panel), we note that at the beginning of the period, VADER is able to capture information at a higher rate, which however saturates around the middle of 2018, meaning it was unable to capture any significant information, and seems to slightly pick up again at the beginning of 2021. These findings, respectively, are in agreement with the almost flat line we observe in the smoothed index of Figure 4 (right, VADER) after mid 2018, and the also almost flat line for Ethereum (see Supplementary Appendix, Section B, Figure 3b in early 2019. We can attribute this to the fact that VADER is trained on general online media content and therefore lacks the specificity required for the crypto domain. For the change of rate we observe in the case of Ethereum around early 2019, we remark that this may be indicative of a change in the language of the articles, namely before that period authors wrote in a less technical and more similar way to the average social media user, which we know is the type of language VADER was trained on. On the contrary, we see that BERT's growth rate is higher than the rest of the indices in the case of BTC and almost parallel to our positive polarity entropy sentiment index for ETH. For the latter, given that BERT has identified predominantly negative sentiment as we saw in Supplementary Appendix, Section B, Figure 3a we interpret this observation as a difference in the understanding of sentiment polarity between our positive entropy index and BERT, given that both are based on the same texts: what the entropy index perceives as positive sentiment, BERT sees as mostly negative.

In addition, we see that for BTC (Figure 10, left), the negative index is almost identical to the absolute sentiment strength, which means that most of the time the negative index captures the same global sentiment tendency as the absolute strength index. However, for ETH (Figure 10, right), even if the two indices appear to grow almost in parallel in terms of information, they intersect and deviate in mid 2021. This is consistent with the price boom that ETH experienced at the beginning of the second quarter of 2021, hence the negative sentiment would no longer dominate in the absolute sentiment strength.

Finally, in both plots of Figure 10, and more evidently in the case of ETH (right), we can see that the rate of change is not constant all of the time - it is higher at the start of the period. This is more distinctly shown in the positive entropy index. This period follows the period of the ETH price peak of January 2018 and we can see that our positive sentiment index reacts more strongly to the resulting sentiment signal.

Figure 10: Cumulative sentiment content of the BTC and ETH sentiment indices constructed from articles from Cryptodaily (http://cryptodaily.co.uk) and Cryptoslate (http://cryptoslate.com).



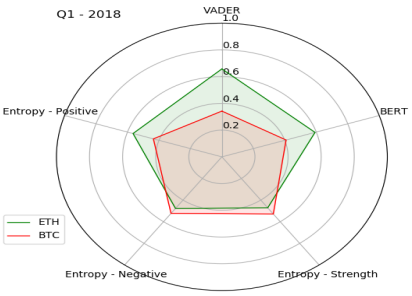
To further illustrate the difference between the signals captured by the examined sentiment constructions we construct the plots of Figures 11 - 14 where we have summarised the sentiment content of the indices in yearly quarters. These figures are obtained by using the interquartile range (IQR) to summarise the sentiment content over each quarter, which would capture the volatility of the signal at each period.

In addition to demonstrating that the sentiment content of the indices is very different, these plots may also reveal significant information about the behaviour of the sentiment indices. We also provide comparative quarterly results for sentiment analysis of the median sentiment and the sentiment volatility under each polarity of sentiment signal. The median sentiment analysis is provided in Supplementary Appendix, Section C.2. In Figure 11, we can see that the IQR of the VADER and BERT indices did not significantly change in Q3 and Q4 of 2018. Given that during those particular quarters Bitcoin was going through a very low volatility period followed by a price crash, hence the sentiment signal, due to mixed reporting about price speculation, was highly varying, it is evident that the BERT and VADER models were unable to capture this variation in sentiment. On the contrary, we observe that the proposed sentiment signals we developed based on Entropy indices are very reactive to the sentiment signal variation. In Q1-Q2 2019, when Bitcoin price started rising again, news reporting started becoming more positive capturing the investors regaining optimism.

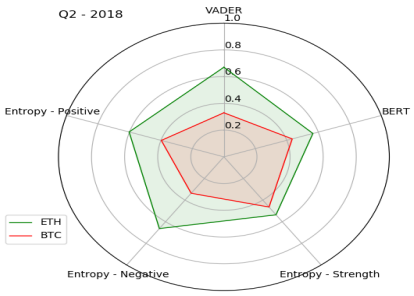
Similarly, observing Figures 13 and 14, we see that the positive Entropy index clearly shows an increase in volatility during Q3 - Q4 2020 and Q1 2021, when Bitcoin price was trending upwards, while in the second quarter of 2021, when there was a price reversal, we remark that the variability of the negative sentiment index starts increasing to reflect the high volatility of the price at that time. Similarly, we observe that BERT reacts significantly to the sentiment signal change between end of 2020 and beginning of 2021, even though we know from Figure 4 (left - BERT) that at that time BERT mostly identified negative sentiment. VADER on the other hand seemed unable to adapt to any change in sentiment after Q4 2020, showing only little change in IQR.

Figure 11: Sentiment index IQR per quarter for each of the sentiment indices.

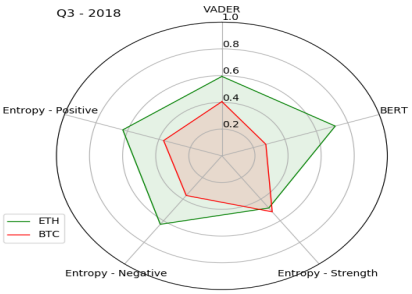
(a) Q1 - 2018



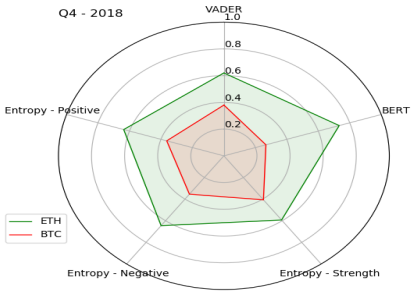
(b) Q2 - 2018



(c) Q3 - 2018



(d) Q4 - 2018



6.2 Case Study II: MIDAS-Koyck Transform Calibration Results

In the second case study, we are interested in exploring a regression relation between daily sentiment, as captured by our daily Entropy index, versus a technology based time-series covariate given by daily hash rate, and the price signal given by intra-daily asset price on an hourly time frame.

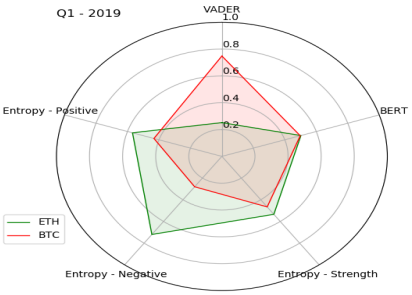
We focus on the intra-daily close price of Bitcoin, and extract the price and hash rate data from CoinMarketCap (`https://coinmarketcap.com/`) for the period of September 2017 - May 2021.

6.2.1 Model Selection for Finite-Lag ARDL-MIDAS-Transformer Time-Series Regressions

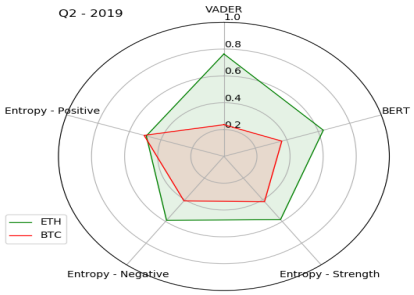
Before proceeding with the analysis, we investigate the fitting of a wide range of model parametrisations with respect to the autoregressive lags, the covariates and the parameters of the Almon weight functions, which, for this study, was the Exponential Almon function.

Figure 12: Sentiment index IQR per quarter for each of the sentiment indices.

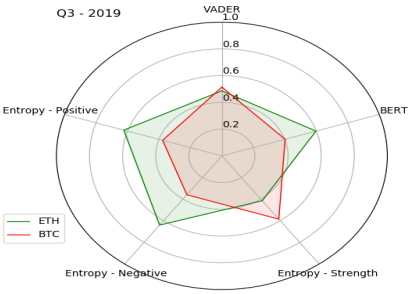
(a) Q1 - 2019



(b) Q2 - 2019



(c) Q3 - 2019



(d) Q4 - 2019

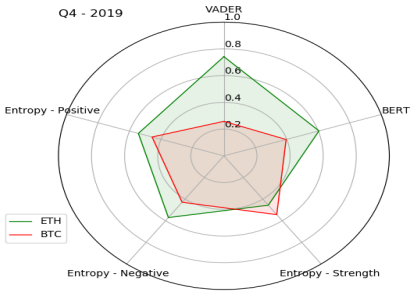
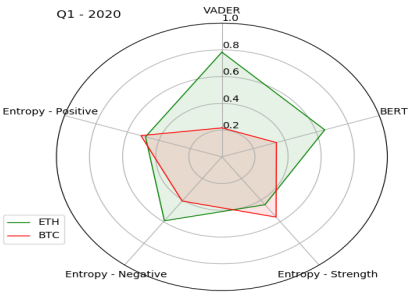
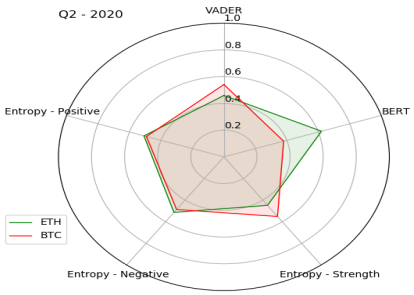


Figure 13: Sentiment index IQR per quarter for each of the sentiment indices.

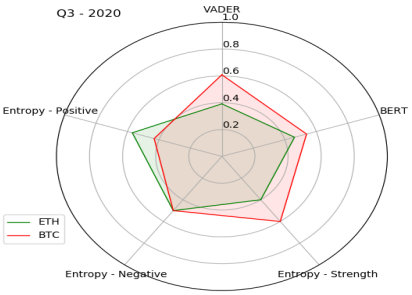
(a) Q1 - 2020



(b) Q2 - 2020



(c) Q3 - 2020



(d) Q4 - 2020

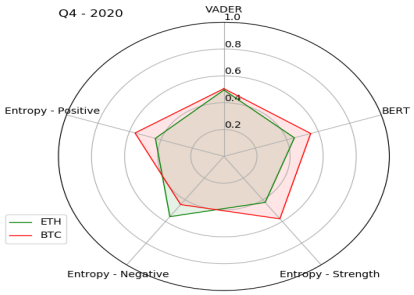
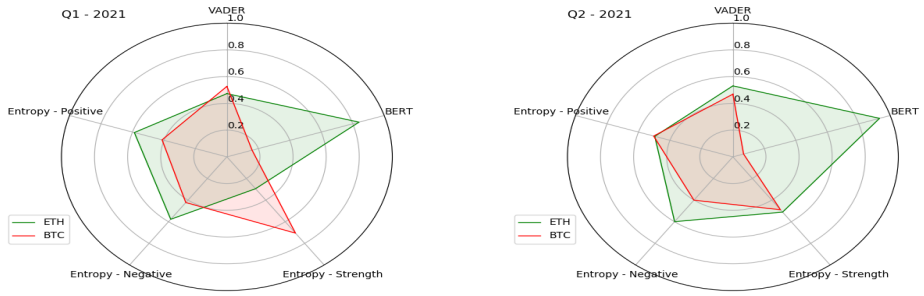


Figure 14: Sentiment index IQR per quarter for each of the sentiment indices.

(a) Q1 - 2021 (b) Q2 - 2021



We adopt in this section the framework of infinite-lags in the ARDL structure, where we apply the Koyck-MIDAS transform to obtain a reparametrised model family as described in Equation 16 using the instrumental variable of Equation 27 constructed from BERT and VADER sentiment signals.

Informed by previous research [27], we explored the options for the lag structure that are included in Table 3. In addition, we used all three different types of sentiment polarity (absolute sentiment strength, positive, negative), and explored a number of optimisation routines from the `R` package `optim`. We assessed the models that fit successfully for the whole range of data using the AIC and Mean Squared Error (MSE) criterion and provide the top-4 fitting models in Table 4 according to these two model selection criteria.

Table 3: The options that we explored for the lag structure in the regression covariates.

Regression component	Lag structure
Response autoregressive component	1-5 days, i.e. 1-5 daily lags
BTC high-frequency component	1-5 days, i.e. 24 - 120 hourly lags
Hash Rate low-frequency component	1-6 months, i.e. 30 - 180 daily lags

Table 4: Best fitting models for different configurations of lag structures and training set-ups.

Model	Sentiment Type	Autoregressive lags (days)	Hash Rate lags (months)	BTC hourly close price lag (days)	Smoothed covariates (1-week median filter)	Hash Rate Almon lag number	BTC close price Almon lag number	AIC	MSE
M1	Abs. sent. strength	5	6	5	True	1	1	-3.9969E+02	0.04112
M2	Abs. sent. strength	5	6	5	True	1	2	–3.9805E+02	0.04110
M3	Abs. sent. strength	5	6	5	False	2	1	-3.9007E+02	0.04146
M4	Abs. sent. strength	5	6	5	False	1	1	-3.8732E+02	0.04163

6.2.2 Assessment of Time-Series ARDL-MIDAS-Transformer Regression Model Over Time

Based on the model search of the previous section, we proceed to study the calibration of the best fitting model in a rolling window fashion, to study how the regression relationship between the response sentiment and the covariates evolves over time. We explore four window sizes, i.e. $720-, 900-, 1080-,$ and $1260-$day rolling windows with a step size of 30 days, and conduct our analysis on the best fitting model according to the AIC (M1) and the MSE (M2).

In addition to the parametrisation of the models as presented in Table 4, we also perform the fitting after applying the Box-Cox transform on the sentiment response, as the initial fittings revealed heteroscedasticity in the residuals. The Box-Cox transform [48] aims to reduce that by transforming the response to resemble a normal variable. The transform, in the case of a positive variable, which is our sentiment response, is given as follows:

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log(y), & \text{otherwise,} \end{cases} \tag{56}$$

where $-5 \leq \lambda \leq 5$. In practice, we found a change in the regression performance for a small range of negative λ values close to zero, and we therefore investigated only the values of $\lambda = -1,\ -0.5, 0$.

The results of the rolling window fits are presented in Table 5 in terms of the percentage of rolling widows where the covariates of the regression were statistically significant. The list is ordered with first key the total number of windows with statistically significant covariates, and second and third keys the total number of windows where the hourly close price (BTC/h) and Hash Rate (HR) covariates respectively were significant. Based on the results of this analysis, we select the best window to employ in the following studies, which is a 1080-day window for the best fitting model according to MSE (M2 in Table 4), and a 1260-day window for the best-fitting model according to AIC (M1 in Table 4).

Table 5: Percentage of rolling windows with statistically significant covariates in the regression. The model configuration is in terms of: (autoregressive daily lags, BTC/h daily lags, HR monthly lags, HR Almon param., BTC Almon param., Box-Cox applied, Box-Cox Lambda if applied, window step size, window size).

Model configuration	% significant - all covariates	% significant BTC/h Almon parameters	% significant HR Almon parameters	% significant at 99% level	% significant at 99.9 % level
5-5-6-1-1-TRUE_-0.5-30-1260	**61.364**	**37.5**	**50**	**61.364**	**61.364**
5-5-6-1-1-TRUE_-0.5-30-1080	60	30	50	60	60
5-5-6-1-1-TRUE_0-30-1260	59.091	37.5	50	59.091	59.091
5-5-6-1-1-FALSE_0-30-1260	59.091	37.5	50	59.091	59.091
5-5-6-1-1-TRUE_-1-30-1260	59.091	37.5	50	59.091	59.091
5-5-6-1-1-TRUE_0-30-1080	58.182	25	50	58.182	58.182
5-5-6-1-1-TRUE_-1-30-900	56.97	43.333	36.667	56.97	56.97
5-5-6-1-1-TRUE_0-30-900	56.818	40.625	40.625	56.818	56.818
5-5-6-1-1-TRUE_-0.5-30-900	56.818	37.5	40.625	56.818	56.818
5-5-6-1-1-FALSE_0-30-1080	56.364	20	50	56.364	56.364
5-5-6-1-1-FALSE_0-30-900	54.545	33.333	43.333	54.545	54.545
5-5-6-1-2-FALSE_0-30-1260	51.515	16.667	16.667	51.515	51.515
5-5-6-1-2-TRUE_-1-30-1080	**50.505**	**27.778**	**50**	**50.505**	**50.505**
5-5-6-1-2-TRUE_-0.5-30-1080	50.505	27.778	44.444	50.505	50.505
5-5-6-1-2-TRUE_-0.5-30-1260	50	12.5	37.5	50	50
5-5-6-1-1-TRUE_-1-30-720	47.934	27.273	29.545	47.934	47.934
5-5-6-1-2-TRUE_-1-30-1260	47.727	12.5	25	47.727	47.727
5-5-6-1-1-TRUE_0-30-720	47.521	27.273	31.818	47.521	47.521
5-5-6-1-2-TRUE_0-30-1080	47.273	15	40	47.273	47.273
5-5-6-1-1-TRUE_-0.5-30-720	47.107	25	29.545	47.107	47.107
5-5-6-1-2-TRUE_-1-30-900	46.853	38.462	30.769	46.853	46.853
5-5-6-1-2-TRUE_-0.5-30-900	46.753	32.143	35.714	46.753	46.753
5-5-6-1-1-FALSE_0-30-720	45.868	31.818	27.273	45.868	45.868
5-5-6-1-2-TRUE_0-30-900	45.455	41.667	29.167	45.455	45.455
5-5-6-1-2-TRUE_0-30-1260	45.455	12.5	25	45.455	45.455
5-5-6-1-2-FALSE_0-30-900	44.056	26.923	34.615	44.056	44.056
5-5-6-1-2-FALSE_0-30-1080	43.939	8.333	41.667	43.939	43.939
5-5-6-1-2-TRUE_-1-30-720	36.364	12.5	34.375	36.364	36.364
5-5-6-1-2-TRUE_0-30-720	35.758	26.667	30	35.758	35.758
5-5-6-1-2-FALSE_0-30-720	35.227	18.75	28.125	35.227	35.227
5-5-6-1-2-TRUE_-0.5-30-720	34.091	15.625	25	34.091	34.091

Next, we provide details about the study of the Almon polynomial coefficients over time for the low- and high-frequency covariates. The majority of the details for the model studies showing dynamics of the Exponential-Almon lag structures for each covariate are provided in the accompanying Supplementary Appendix in Section D, and here we present the results for Model M1 (best according to AIC) with the coefficients for the BTC hourly close price and the daily hash rate in Figures 15 and 16, when fitting the model without applying the Box-Cox transform and with Box-Cox applied with $\lambda = 0$.

6.2.3 Statistical Significance of Interactions between data at Mixed Frequencies

In this section, we continue to explore the results of model M1 by looking at the statistical significance of the coefficients of the autoregressive response covariate, as well as the low- and high-frequency covariates. In Figure 17 we plot the p-values for the fitted M1 models in which the Distributed Lag MIDAS covariates were found to be statistically significant. For the corresponding plots for model M2, please refer to the Supplementary Appendix, Section D. In terms of performance, we observe in Figure 23 that the models fit with a Box-Cox transform exhibit an improved performance compared to the models without.

Figure 15: M1: Exponential Almon coefficients structure of BTC hourly close price over time for the model fit on the complete dataset. Top Row Plots: No Box-Cox Transform applied. Bottom Plots: Box-Cox Transform with $\lambda = 0$ applied.

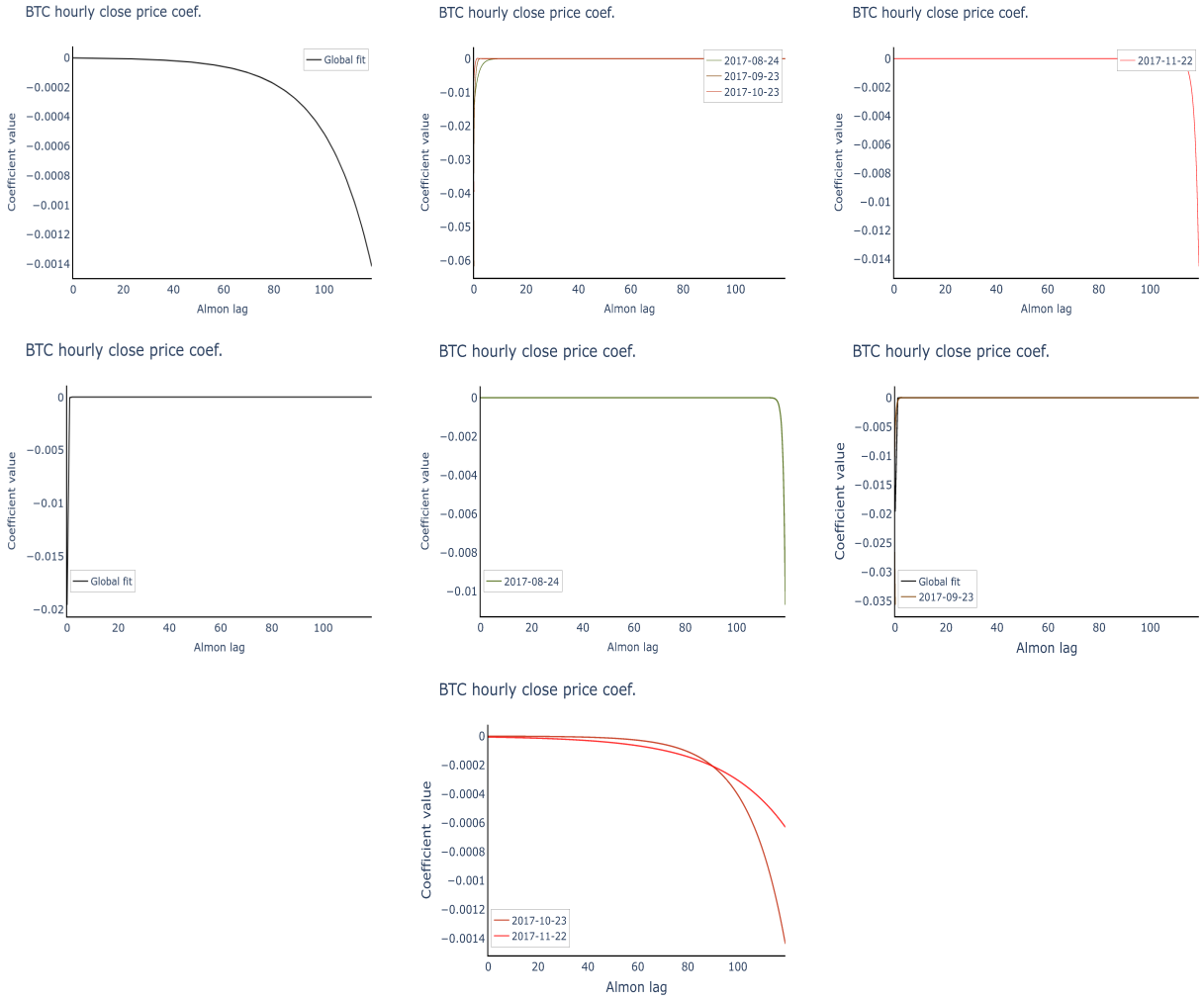


Figure 16: M1: Exponential Almon coefficients structure of Hash Rate over time for the model fit on the complete dataset. Left Plots: No Box-Cox Transform applied. Right Plots: Box-Cox Transform with $\lambda = 0$ applied.

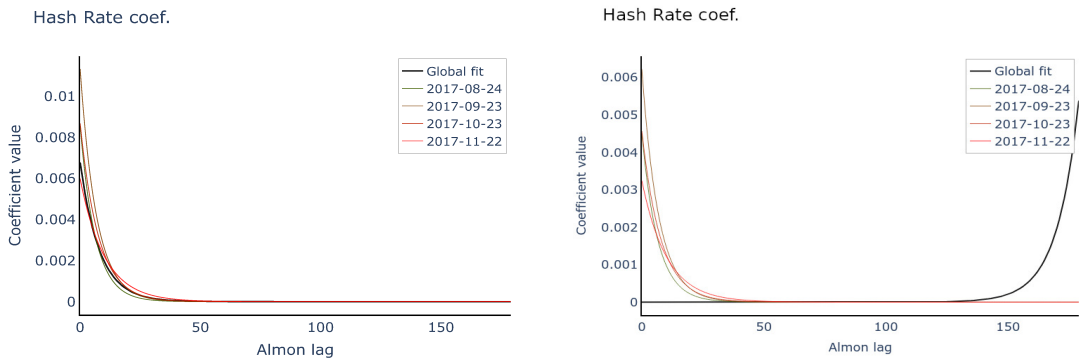
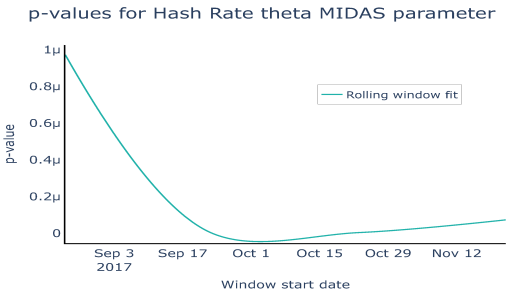
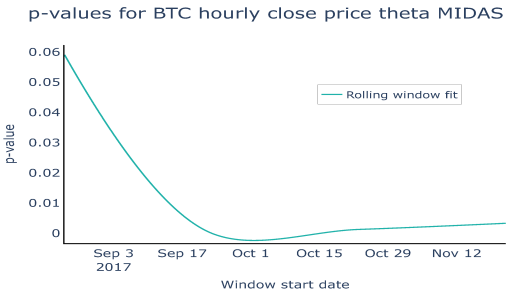


Figure 17: M1: Left Subplots: Hash Rate low-freq. covariate. Right Subplots: BTC hourly close price high-freq. covariate. P-values for the statistically significant Exponential Almon parameters (β_1, ψ).

(a) No Box-Cox Transform Applied

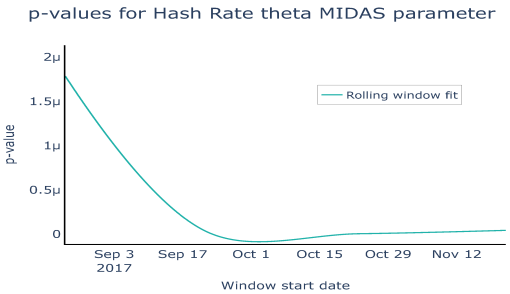


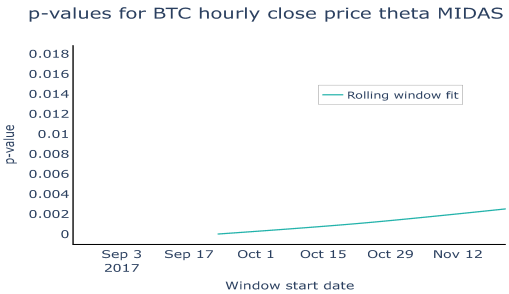
(b) No Box-Cox Transform Applied



(c) Box-Cox Transform Applied ($\lambda = 0$).

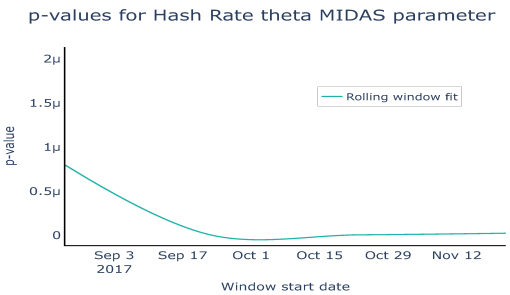
(d) Box-Cox Transform Applied ($\lambda = 0$).

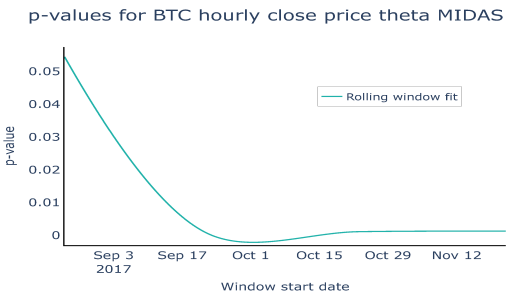




(e) Box-Cox Transform Applied ($\lambda = -0.5$).

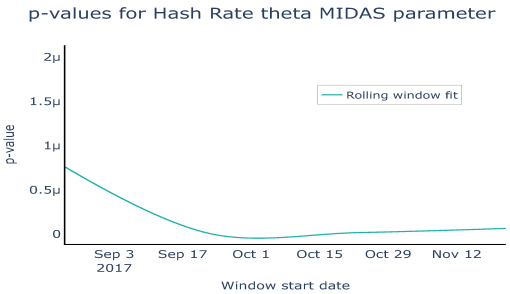
(f) Box-Cox Transform Applied ($\lambda = -0.5$).

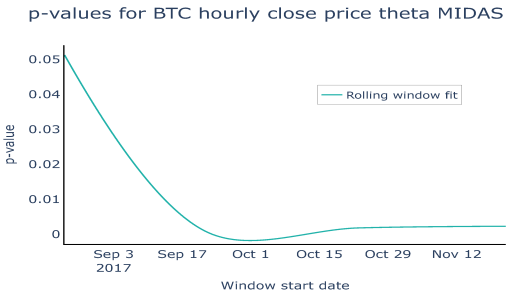




(g) Box-Cox Transform Applied ($\lambda = -1$).

(h) Box-Cox Transform Applied ($\lambda = -1$).





Furthermore, in the Supplementary Appendix, Section D.1, we demonstrate the p-values on the significance of the AR lags of the Koyck-Midas transformed coefficients in Equation 16 for the cases of no Box-Cox transformation and a few cases of Box-Cox transformed fitted models, for models M1 and M2.

6.2.4 Mixed-data Long Memory Structures

We now seek to explore how the weight function that we employ for the MIDAS coefficients affects the persistence properties of the high- and low-frequency covariates. To investigate, we estimate the Hurst exponent from the covariates, the response, and the residuals per fitting window and explore their relationship. We plot the estimated Hurst exponents of the response and the residuals in Figures 18 - 22. Note that the Hurst exponent is a positive number upper-bounded by 1 - the small bias that is observed in some estimates is consistent with the behaviour of the estimator in large-scale synthetic studies that we performed when testing the behaviour of this estimator on controlled synthetic studies, before applying it in this real data case study.

We observe that the long memory strength in the response, as captured via the residuals, is attenuated compared to the long memory in the covariates as a result of the MIDAS structure in the model. Therefore, the long memory structure of the covariates is not trivially transferred to the response in such settings, which has to be considered if including the long memory is an important desired feature for the model.

Figure 18: M1: BTC/h (left) and Hash rate (right) vs Residuals and Sentiment response Hurst exponents. No Box-Cox transform was applied. Red, diamond-shaped markers correspond to the response-derived Hurst, whilst the blue, circle-shaped markers to the residual-derived Hurst.

(a) (b)

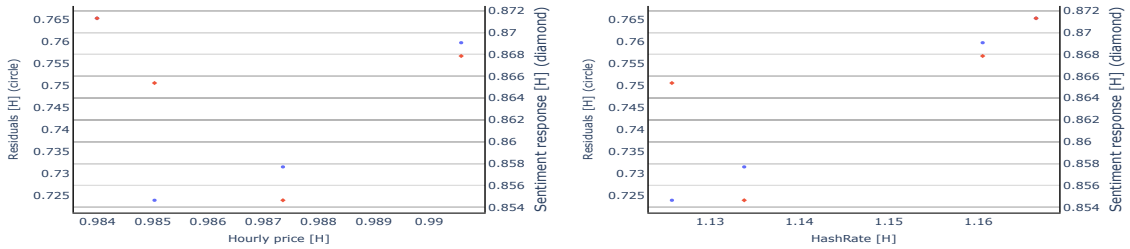
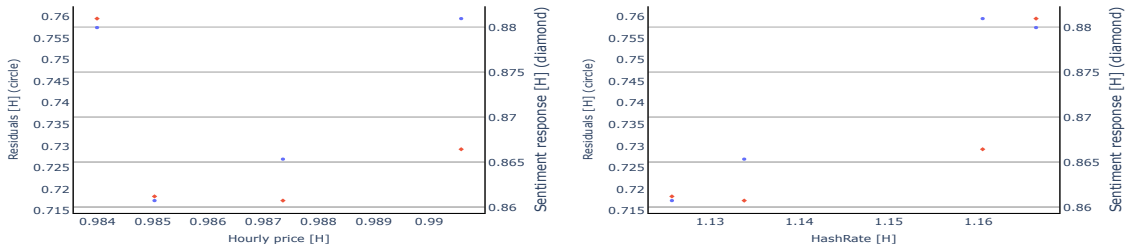


Figure 19: M1: BTC/h (left) and Hash rate (right) vs Residuals and Sentiment response Hurst exponents. The Box-Cox transform with $\lambda = 0$ was applied. Red, diamond-shaped markers correspond to the response-derived Hurst, whilst the blue, circle-shaped markers to the residual-derived Hurst.

(a) (b)



Note that should we wish to use Gegenbauer polynomial MIDAS weights, we could estimate the d and u parameters from the residuals. The long memory d is defined as $d = H - 0.5$, whilst, in this instance, we can estimate the cyclic frequency u by observing the autocorrelation plots of the residuals per window, as we observed that the ACF exhibits almost no oscillation around the x axis, which means that $u = 0$, i.e. the long memory is coming from an ARFIMA-type process. We have provided a detailed analysis of the autocorrelation profiles of the regression for model M1 and model M2 with and without the Box-Cox transform in the Supplementary Appendix, Section D.2

Figure 20: M1: BTC/h (left) and Hash rate (right) vs Residuals and Sentiment response Hurst exponents. The Box-Cox transform with $\lambda = -0.5$ was applied. Red, diamond-shaped markers correspond to the response-derived Hurst, whilst the blue, circle-shaped markers to the residual-derived Hurst.

(a) (b)

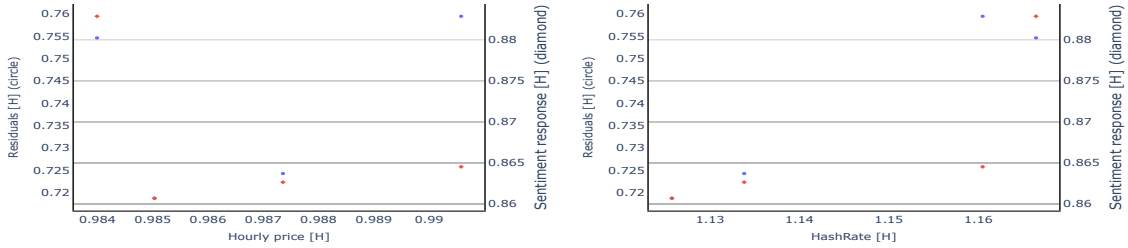


Figure 21: M1: BTC/h (left) and Hash rate (right) vs Residuals and Sentiment response Hurst exponents. The Box-Cox transform with $\lambda = -1$ was applied. Red, diamond-shaped markers correspond to the response-derived Hurst, whilst the blue, circle-shaped markers to the residual-derived Hurst.

(a) (b)

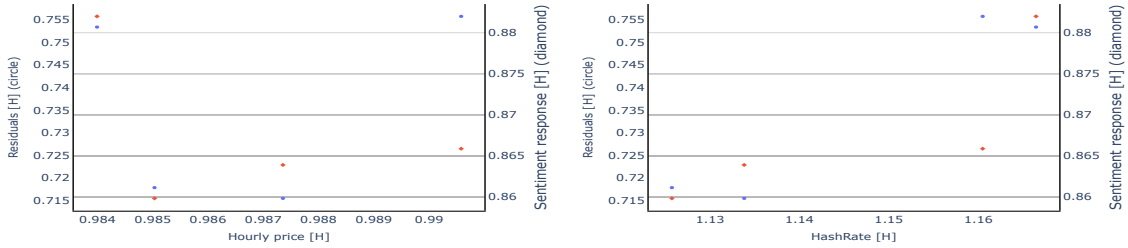


Figure 22: M2: BTC/h (left) and Hash rate (right) vs Residuals and Sentiment response Hurst exponents. No Box-Cox transform was applied. Red, diamond-shaped markers correspond to the response-derived Hurst, whilst the blue, circle-shaped markers to the residual-derived Hurst.

(a) (b)

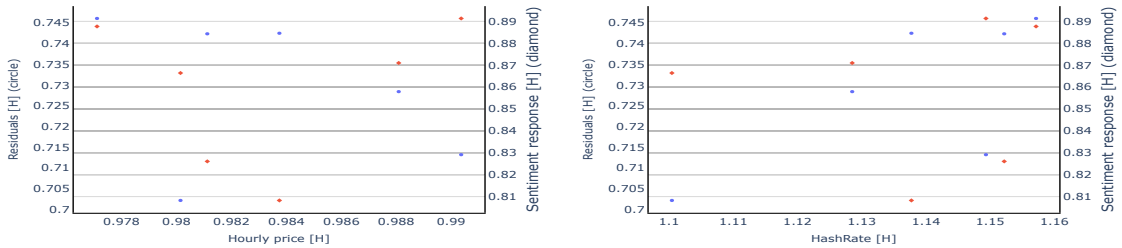
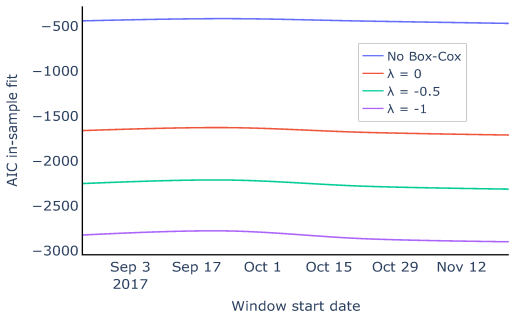


Figure 23: M1: AIC for each rolling window fit.



6.2.5 Mixed-data Infinite-Lag Koyck Regressions: Estimation of γ

In this section, we study the in-sample fitting performance of the ARDL-MIDAS model with the Koyck transform to obtain tractability with an infinite number of lags. We estimate the model with and without (see Sections 6.2.1-6.2.4) the adjustments dictated by the Koyck transform, and then, having obtained the regression coefficients for the autoregressive lags of Equation 16 before and after the transformation, we can form a system of equations to solve for decay parameter γ of the transform under the constraint $0 < \gamma < 1$.

We inform our selection for the autoregressive covariate in Equation 16 from our analysis in Case Study I, and use as instrumental variable a linear combination of the Entropy Sentiment and the BERT and VADER sentiment covariates, which decorrelate the sentiment response from the errors in this formulation as studied in Case Study I. Specifically, the IV in the current study is the difference between the Entropy Sentiment index, median-smoothed weekly in a rolling window fashion, and the average of the also weekly median-smoothed BERT and VADER sentiment signals.

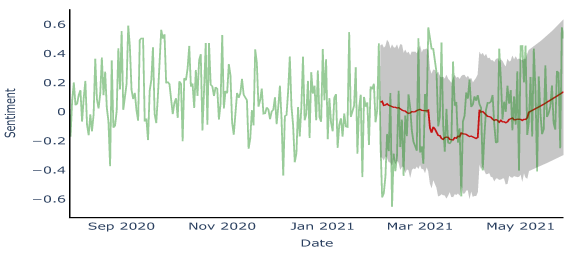
6.2.6 Forecasting and the effect of the infinite-lag Koyck transform

We finalise the study with an analysis of how well the proposed infinite lag ARDL-MIDAS-Transformer long memory time-series regression models perform on out-of-sample forecasting of daily crypto sentiment. To assess the forecasting performance of the fitted models, we use each model fitted on a rolling window to forecast one month ahead of the window. First, for the models without the infinite lag Koyck adjustment, we perform the fitting without applying the Box-Cox transform, and present the results in Figure 24 and Tables 6a and 6b.

Second, we extend the estimated models with the Koyck adjustment of Equation 16 for a range of values of γ on a grid: $0.01, 0.05, 0.1, \ldots, 0.95$, where the step size after 0.1 is 0.05. We evaluate the forecast performance in terms of MSE and illustrate the results in Figure 25. We observe that the performance after the Koyck adjustment (left y-axis) is significantly diminished, which means that the geometric decay we adopted for the coefficients in the Koyck transform is not optimal in this setting.

Figure 24: Forecasting with rolling-windows fitted models. The true sentiment signal has been detrended in the following plots as we did for the fitting. The red trace denotes the cubic spline-smoothed signal, and the grey band denotes the 95% confidence interval.

(a) M1 - best model according to AIC



(b) M2 - best model according to MSE

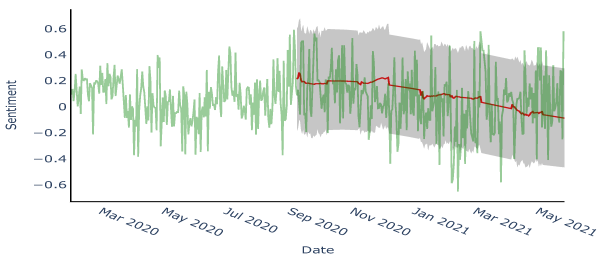


Table 6: Forecasting with rolling-window fitted models.

(a) Performance of model M1 (best according to AIC).

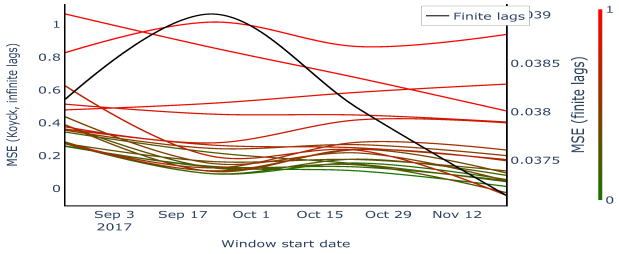
30-day forecast start	M1
2021-02-05	1.15E-01
2021-03-07	7.47E-02
2021-04-06	6.36E-02

(b) Performance of model M2 (best according to MSE).

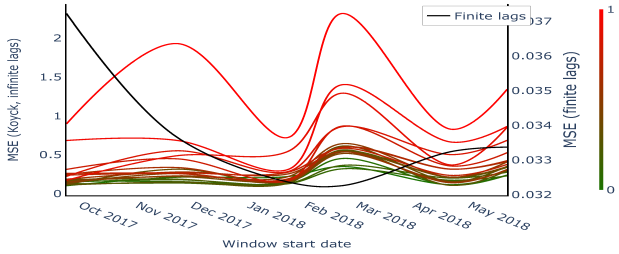
30-day forecast start	M2
2020-09-08	9.58E-02
2020-11-07	3.45E-02
2021-02-05	1.21E-02
2021-04-06	5.78E-02

Figure 25: MSE forecasting performance with the rolling-window fitted models of Section 6.2.1, after having added the infinite-lag Koyck transform for a range of values for γ. The latter are sampled from a grid on $(0, 1)$, which is shown with the color graduation: the redder the trace, the higher the value of γ. The reference model without the Koyck transform is plotted in black.

(a) M1 - best model according to AIC



(b) M2 - best model according to MSE



7 Conclusion

In this work, we have proposed a novel class of time-series regression models that incorporate several relevant interpretable features: infinite-lag ARDL regressions, Mixed Data Sampling MIDAS multiple time resolution regression structures, Deep Neural Network architectures for Instrumental Variable design for reduction of the estimation bias in the ARDL-MIDAS-Transformer class of models, and, finally, fractional integration in the form of Gegenbauer long memory polynomials for the MIDAS configuration. Each of these model components is carefully explained and detailed and then a thorough real data statistical analysis is undertaken on cryptocurrency market sentiment constructed daily from news articles. The daily sentiment is then regressed against intra-daily hourly closing price dynamics and money supply, as captured by mining Hash Rate. Overall, we see the advantage in this application in fitting performance, incorporation of sophisticated long memory signal characteristics, and interpretation capabilities achieved with this new class of models we introduce, namely the class of infinite lag ARDL-MIDAS-Transformer time-series regression models.

8 Software and Technical Appendix

Code and data for reproducibility purposes are available at `https://github.com/ichalkiad/ardlmidasdnn`.

The reader is also referred to the supplementary Technical Appendix for additional results and analyses.

9 Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] B. Liu, Sentiment analysis and opinion mining, Synthesis Lectures on Human Language Technologies 5 (1) (2012) 1–167. arXiv:https://doi.org/10.2200/S00416ED1V01Y201204HLT016, doi:10.2200/S00416ED1V01Y201204HLT016.

[2] B. Pang, L. Lee, et al., Foundations and trends® in information retrieval, Foundations and Trends® in Information Retrieval 2 (1-2) (2008) 1–135.

[3] P. J. Dhrymes, L. R. Klein, K. Steiglitz, Estimation of distributed lags, International Economic Review 11 (2) (1970) 235–250.

[4] L. R. Klein, The estimation of distributed lags, Econometrica: Journal of the Econometric Society (1958) 553–565.

[5] E. J. Hannan, The estimation of relationships involving distributed lags, Econometrica: Journal of the Econometric Society (1965) 206–224.

[6] K. L. Marinus, Distributed lags and investment analysis / by L. M. Koyck, Contributions to economic analysis, North-Holland Pub. Co., Amsterdam, 1954.

[7] J. H. Stock, F. Trebbi, Retrospectives: who invented instrumental variable regression?, Journal of Economic Perspectives 17 (3) (2003) 177–194.

[8] E. Ghysels, P. Santa-Clara, R. Valkanov, There is a risk-return trade-off after all, Journal of Financial Economics 76 (3) (2005) 509–548.

[9] E. Ghysels, P. Santa-Clara, R. Valkanov, Predicting volatility: getting the most out of return data sampled at different frequencies, Journal of Econometrics 131 (1-2) (2006) 59–95.

[10] E. Ghysels, A. Sinko, R. Valkanov, Midas regressions: Further results and new directions, Econometric reviews 26 (1) (2007) 53–90.

[11] E. Andreou, E. Ghysels, A. Kourtellos, Should macroeconomic forecasters use daily financial data and how?, Journal of Business & Economic Statistics 31 (2) (2013) 240–251.

[12] E. Andreou, E. Ghysels, A. Kourtellos, Forecasting with mixed-frequency data, in: The Oxford handbook of economic forecasting, 2011, p. 225–246.

[13] J. Beran, Statistics for long-memory processes, Vol. 61, CRC press, 1994.
URL https://doi.org/10.1201/9780203738481

[14] E. Ghysels, P. Santa-Clara, R. Valkanov, The midas touch: Mixed data sampling regressions, manuscript, University of North Carolina and UCLA (2004).

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6000–6010.

[16] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, p. 3104–3112.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[18] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das, et al., What do you learn from context? probing for sentence structure in contextualized word representations, arXiv preprint arXiv:1905.06316 (2019).

[19] I. Tenney, D. Das, E. Pavlick, Bert rediscovers the classical nlp pipeline, arXiv preprint arXiv:1905.05950 (2019).

[20] C. J. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM), June 1–4 2014, The AAAI Press, Ann Arbor, Michigan, USA, 2014, pp. 216–225.
URL https://www.scinapse.io/papers/2099813784

[21] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, Q. Liu, Tinybert: Distilling bert for natural language understanding, arXiv preprint arXiv:1909.10351 (2019).

[22] J. Pennebaker, C. Chung, M. Ireland, A. Gonzales, R. Booth, The development and psychometric properties of liwc2007 (01 2007).

[23] L. Zhang, B. Liu, Sentiment Analysis and Opinion Mining, Springer US, Boston, MA, 2017, pp. 1152–1161. doi:10.1007/978-1-4899-7687-1_907.

[24] T. Loughran, B. McDonald, When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks, Journal of Finance 66 (1) (2011) 35–65.
URL https://doi.org/10.1111/j.1540-6261.2010.01625.x

[25] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, M. R. Yeganegi, Text Mining in Big Data Analytics, Big Data and Cognitive Computing 4 (1) (January 2020). doi:10.3390/bdcc4010001.
URL https://doi.org/10.3390/bdcc4010001

[26] I. Chalkiadakis, H. Yan, G. W. Peters, P. V. Shevchenko, Infection rate models for covid-19: Model risk and public health news sentiment exposure adjustments, PLOS ONE 16 (6) (2021) 1–39. doi:10.1371/journal.pone.0253381.
URL https://doi.org/10.1371/journal.pone.0253381

[27] I. Chalkiadakis, A. Zaremba, G. W. Peters, M. J. Chantler, On-chain analytics for sentiment-driven statistical causality in cryptocurrencies, Available at SSRN 3742063 (2020).
URL https://ssrn.com/abstract=3742063

[28] Z. Harris, Distributional Structure, Word 10 (23) (1954) 146–162.
URL https://doi.org/10.1080/00437956.1954.11659520

[29] E. Ghysels, V. Kvedaras, V. Zemlys, Mixed frequency data sampling regression models: the r package midasr, Journal of statistical software 72 (1) (2016) 1–35.

[30] C. Foroni, M. Marcellino, C. Schumacher, Unrestricted mixed data sampling (midas): Midas regressions with unrestricted lag polynomials, Journal of the Royal Statistical Society. Series A (Statistics in Society) (2015) 57–82.

[31] H. E. Hurst, Long-term storage capacity of reservoirs, Transactions of the American Society of Civil Engineers 116 (1) (1951) 770–799. arXiv:https://ascelibrary.org/doi/pdf/10.1061/TACEAT.0006518, doi:10.1061/TACEAT.0006518.

[32] B. B. Mandelbrot, Limit theorems on the self-normalized range for weakly and strongly dependent processes, Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 31 (4) (1975) 271–285.
URL https://doi.org/10.1007/BF00534968

[33] A. Annis, E. Lloyd, The expected value of the adjusted rescaled hurst range of independent normal summands, Biometrika 63 (1) (1976) 111–116.
URL https://doi.org/10.1093/biomet/63.1.111

[34] E. E. Peters, Fractal market analysis : applying chaos theory to investment and economics, Wiley finance editions, J. Wiley & Sons, New York, 1994.

[35] R. Weron, Estimating long-range dependence: finite sample properties and confidence intervals, Physica A: Statistical Mechanics and its Applications 312 (1) (2002) 285–299. doi:https://doi.org/10.1016/S0378-4371(02)00961-5.
URL https://www.sciencedirect.com/science/article/pii/S0378437102009615

[36] J. Abraham, D. Higdon, J. Nelson, J. Ibarra, Cryptocurrency price prediction using tweet volumes and sentiment analysis, SMU Data Science Review 1 (3) (2018) 1.
URL https://scholar.smu.edu/datasciencereview/vol1/iss3/1

[37] O. Kraaijeveld, J. De Smedt, et al., The predictive power of public twitter sentiment for forecasting cryptocurrency prices, Journal of International Financial Markets, Institutions and Money 65 (C) (2020).
URL https://doi.org/10.1016/j.intfin.2020.101188

[38] Y. B. Kim, J. G. Kim, W. Kim, J. H. Im, T. H. Kim, S. J. Kang, C. H. Kim, Predicting fluctuations in cryptocurrency transactions based on user comments and replies, PloS One 11 (8) (2016) e0161197.
URL https://doi.org/10.1371/journal.pone.0161197

[39] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. doi:10.3115/v1/D14-1162.
URL https://www.aclweb.org/anthology/D14-1162

[40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association

for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
URL https://www.aclweb.org/anthology/N19-1423

[41] R. C. Hill, W. E. Griffiths, G. G. Judge, Undergraduate econometrics / R. Carter Hill, William E. Griffiths, George G. Judge., 2nd Edition, John Wiley, New York, 2001.

[42] T. S. Breusch, Testing for autocorrelation in dynamic linear models, Australian Economic Papers 17 (31) (1978) 334–355. doi:https://doi.org/10.1111/j.1467-8454.1978.tb00635.x.
URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8454.1978.tb00635.x

[43] D. Kwiatkowski, P. C. Phillips, P. Schmidt, Y. Shin, Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?, Journal of Econometrics 54 (1) (1992) 159–178. doi:https://doi.org/10.1016/0304-4076(92)90104-Y.
URL https://www.sciencedirect.com/science/article/pii/030440769290104Y

[44] D. S. Dimitrova, V. K. Kaishev, S. Tan, Computing the kolmogorov-smirnov distribution when the underlying cdf is purely discrete, mixed, or continuous, Journal of Statistical Software, Articles 95 (10) (2020) 1–42. doi:10.18637/jss.v095.i10.
URL https://www.jstatsoft.org/v095/i10

[45] O. Vasicek, A test for normality based on sample entropy, Journal of the Royal Statistical Society: Series B (Methodological) 38 (1) (1976) 54–59. doi:https://doi.org/10.1111/j.2517-6161.1976.tb01566.x.
URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1976.tb01566.x

[46] K.-S. Song, Goodness-of-fit tests based on kullback-leibler discrimination information, IEEE Transactions on Information Theory 48 (5) (2002) 1103–1117. doi:10.1109/18.995548.

[47] J. Lequesne, P. Regnault, vsgoftest: An r package for goodness-of-fit testing based on kullback-leibler divergence, Journal of Statistical Software, Code Snippets 96 (1) (2020) 1–26. doi:10.18637/jss.v096.c01.
URL https://www.jstatsoft.org/v096/c01

[48] G. E. P. Box, D. R. Cox, An analysis of transformations, Journal of the Royal Statistical Society. Series B (Methodological) 26 (2) (1964) 211–252.
URL http://www.jstor.org/stable/2984418

HYBRID ARDL-MIDAS-TRANSFORMER TIME-SERIES REGRESSIONS FOR MULTI-TOPIC CRYPTO MARKET SENTIMENT DRIVEN BY PRICE AND TECHNOLOGY FACTORS: SUPPLEMENTARY APPENDIX

Ioannis Chalkiadakis
School of Mathematical and Computer Sciences,
Heriot-Watt University
ic14@hw.ac.uk

Gareth W. Peters
Department of Statistics & Applied Probability
University of California Santa Barbara
garethpeters@ucsb.edu

Matthew Ames
ResilientML
Victoria, Australia
matt.ames87@gmail.com

August 19, 2021

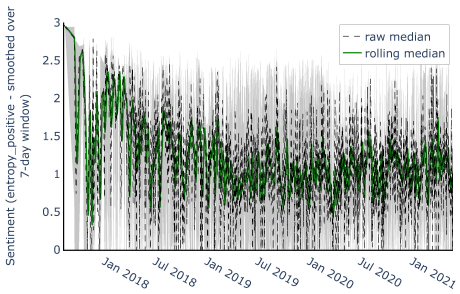
Contents

A Additional Entropy Sentiment time-series: Ethereum

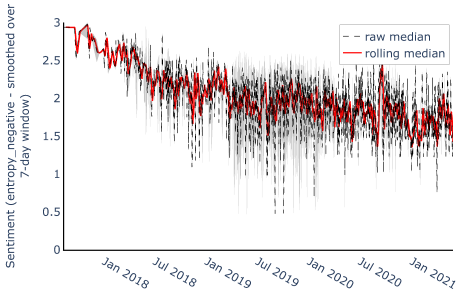
In Figure 1 we plot the smoothed volume-weighted positive and negative sentiment indices as well as the index of absolute sentiment strength, for articles referring to Ethereum.

Figure 1: Sentiment indices with 95% confidence intervals constructed from articles about Ethereum published on Cryptodaily (http://cryptodaily.co.uk) and Cryptoslate (http://cryptoslate.com)

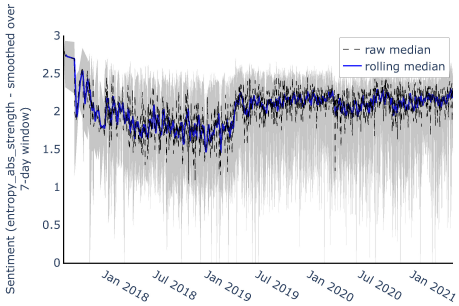
(a) Positive sentiment



(b) Negative sentiment



(c) Absolute sentiment strength

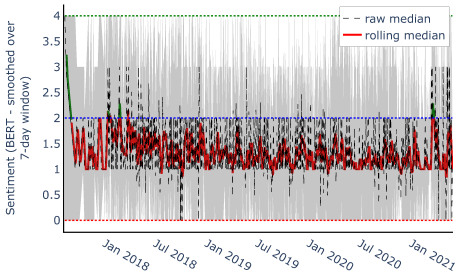


B Additional BERT and VADER sentiment time-series: Ethereum

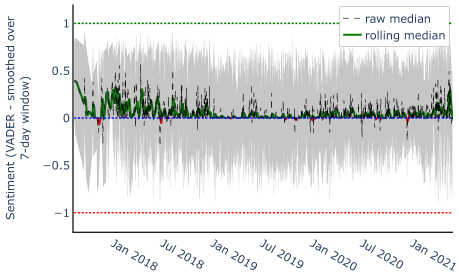
In Figure 2 we plot the sentiment time-series for ETH obtained via the BERT and VADER models.

Figure 2: Sentiment indices based on BERT and VADER with 95% confidence intervals constructed from articles about Ethereum published on Cryptodaily (http://cryptodaily.co.uk) and Cryptoslate (http://cryptoslate.com)

(a) BERT



(b) VADER



C Two-stage Estimation of ARDL(∞) Regression via the classical Koyck transform

In this section we will present the estimation procedure for the classical infinite-lag distributed model, where we consider utilising two covariates. We will consider time-series $\{Y_t\}$ given by the daily sentiment score based on our proposed Entropy Sentiment time-series construction. We will then regress this sentiment scoring method against alternative sentiment extraction methods based on BERT and VADER that we will transform into time-series covariates and denote them by $\{X_t^B\}$ and $\{X_t^V\}$, also constructed from daily measures of sentiment. Then we seek to fit the regression model, which includes the geometric decay parametric form for the coefficients as described in Section 2.1 of the main manuscript:

$$\begin{aligned} Y_t &= \alpha + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \sum_{j=0}^{+\infty} \beta_j^B X_{t-j}^B + \sum_{j=0}^{+\infty} \beta_j^V X_{t-j}^V + \epsilon_t \\ &= \alpha + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \beta^B \sum_{j=0}^{+\infty} \phi_B^j X_{t-j}^B + \beta^V \sum_{j=0}^{+\infty} \phi_V^j X_{t-j}^V + \epsilon_t. \end{aligned} \tag{1}$$

First we will include only one of the additional covariates, say BERT. At times t and $t-1$ we have:

$$\begin{aligned} Y_t &= \alpha + \sum_{i=1}^{p} \gamma_i Y_{t-i} + \sum_{j=0}^{+\infty} \beta_j^B X_{t-j}^B + \epsilon_t, \\ Y_{t-1} &= \alpha + \sum_{i=1}^{p} \gamma_i Y_{t-1-i} + \beta^B \sum_{j=0}^{\infty} \phi_B^j X_{t-1-j}^B + \epsilon_{t-1}. \end{aligned} \tag{2}$$

Working as before, we begin by multiplying the second row with the geometric decay rate ϕ_B:

$$\phi_B Y_{t-1} = \phi_B \alpha + \phi_B \sum_{i=1}^{p} \gamma_i Y_{t-1-i} + \beta^B \sum_{j=0}^{+\infty} \phi_B^{j+1} X_{t-1-j}^B + \phi_B \epsilon_{t-1} \tag{3}$$

Subtracting the expressions in equations 2 (for Y_t) and 3 we obtain:

$$Y_t - \phi_B Y_{t-1} = \alpha(1-\phi_B) + \epsilon_t - \phi_B \epsilon_{t-1} + \sum_{i=1}^{p} \gamma_i (Y_{t-i} - \phi_B Y_{t-1-i}) + \beta^B X_t^B,$$

and equivalently we have:

$$Y_t = \alpha(1-\phi_B) + \phi_B Y_{t-1} + \sum_{i=1}^{p} \gamma_i (Y_{t-i} - \phi_B Y_{t-1-i}) + \beta^B X_t^B + \epsilon_t - \phi_B \epsilon_{t-1}. \tag{4}$$

If we then work in the same way for the VADER covariate we obtain:

$$Y_t = \alpha(1-\phi_V) + \phi_V Y_{t-1} + \sum_{i=1}^{p} \gamma_i (Y_{t-i} - \phi_V Y_{t-1-i}) + \beta^V X_t^V + \epsilon_t - \phi_V \epsilon_{t-1},$$

and after adding the last two models, one obtains:

$$Y_t = \alpha(1-\phi) + \phi Y_{t-1} + \sum_{i=1}^{p} \gamma_i (Y_{t-i} - \phi Y_{t-1-i}) + \beta^B X_t^B + \beta^V X_t^V + \epsilon_t - \phi \epsilon_{t-1},$$

where we have substituted $\phi = \frac{\phi_B + \phi_V}{2}$.

The final regression model can be estimated with a least squares procedure. Note that we cannot use ordinary least squares (OLS) as it would produce inconsistent estimators since the error term is autocorrelated and therefore correlated with the lagged dependent variable Y_{t-1}. We will instead estimate the model in a two-step procedure using instrumental variables (IV) regression. Because we do not aim to obtain a good model for Y_t for prediction purposes, to simplify our estimation we keep only one lag of the dependent variable in the regression and set $\gamma_i = 0$ for $i = 1, \ldots, p$:

$$Y_t = \alpha(1-\phi) + \phi Y_{t-1} + \beta^B X_t^B + \beta^V X_t^V + \epsilon_t - \phi \epsilon_{t-1}. \tag{5}$$

For completeness, we repeat here the two steps of the regression, which have been presented in detail in the main manuscript, Section 6.1:

Stage I : Regress the entropy sentiment index against its lag $1, \ldots, p$ values, by fitting an AR(p) model on Y_t.

Stage II : Determine the most suitable model for the instrumental variables of Y_t covariate.

C.1 Stage I

In this first step we remove the autocorrelation from the independent variable by regressing the following:

$$Y_t = \mu_0 + \sum_{j=1}^{p} \mu_j Y_{t-j} + \epsilon_t. \tag{6}$$

We can estimate the parameters of the previous AR(p) model by ordinary least squares, and given those we obtain the raw residuals E_t^y. This first regression will be our reference model for the regressions in Stage II; if we improve the fit with statistical significance in terms of AIC [1] by adding covariates, then the added covariates will be successful in explaining some of the content of Y_t.

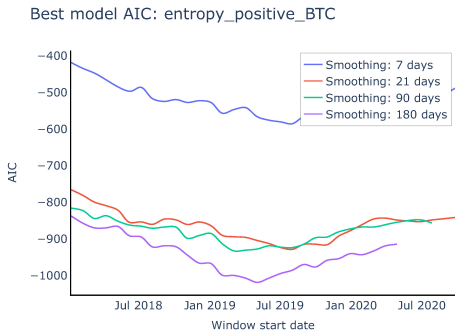
C.1.1 Koyck transform Stage I: model selection and calibration plots for the Positive and Negative Entropy Sentiment indices

As described in the main manuscript, for each configuration we stored the successfully fitted models and ranked them according to the AIC. In Figure 3, we plot the AIC of the best fitting model per window for all smoothing parametrisations. We next plot the lag order per fitting window for the best fitting model (Figure 4 - BTC, 5 - ETH, left panels), and in addition, in the same plots we demonstrate the lag order per fitting window for the best fitting model after dropping the coefficients that were not statistically significant to any level up to 90%.

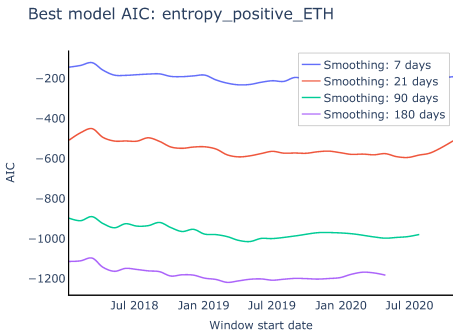
In Figures 4 and 5, we use red colour to plot the worst performing model according to the AIC. Note that for Bitcoin, the importance of selecting a different lag per window exists in the case of statistically significant coefficients, although less prevalent, in the fits for the signals of different polarities (positive vs negative); there the lag structure mainly fluctuates around either one or two lags.

Figure 3: Stage I rolling window fits: best AR(p) model AIC score.

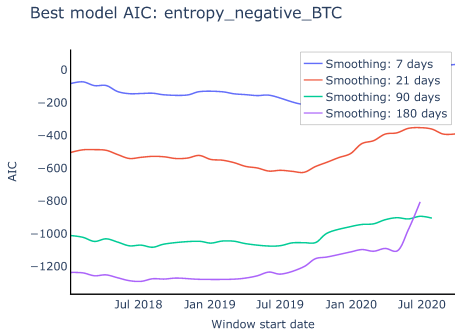
(a) BTC - positive sentiment



(b) ETH - positive sentiment



(c) BTC - negative sentiment



(d) ETH - negative sentiment

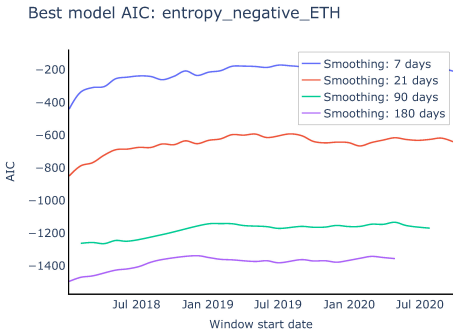
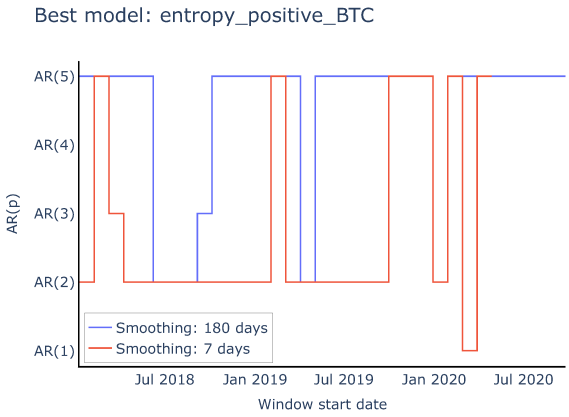
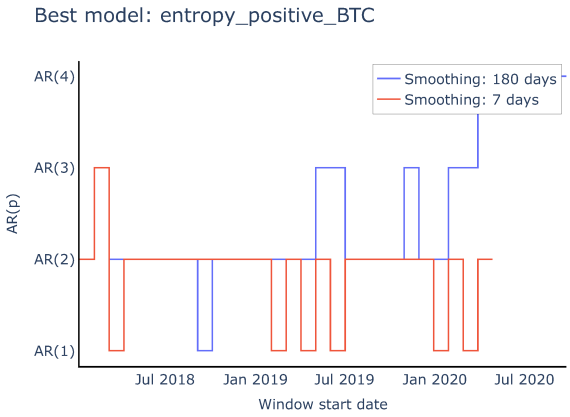


Figure 4: Stage I rolling window fits: best AR lag structure.

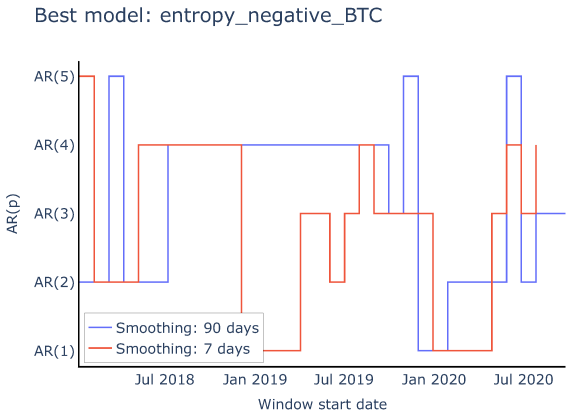
(a) BTC - positive sentiment



(b) BTC - positive sentiment - $\geq 90\%$ coef. significance



(c) BTC - negative sentiment



(d) BTC - negative sentiment - $\geq 90\%$ coef. significance

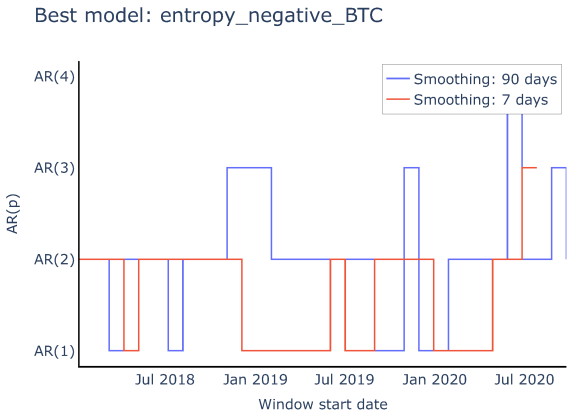
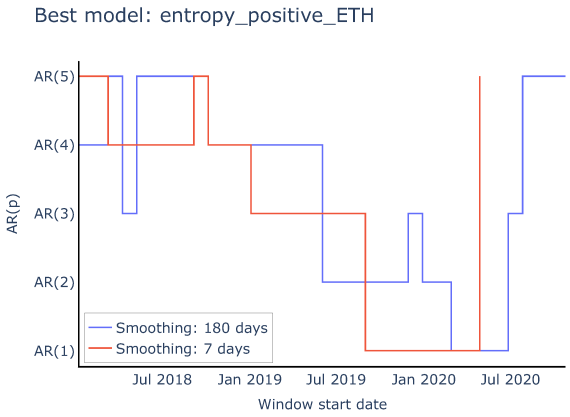
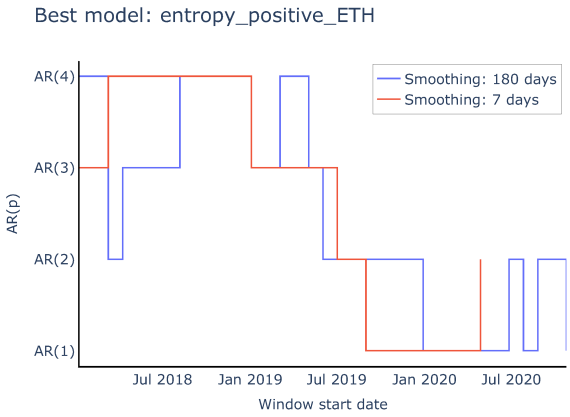


Figure 5: Stage I rolling window fits: best AR lag structure.

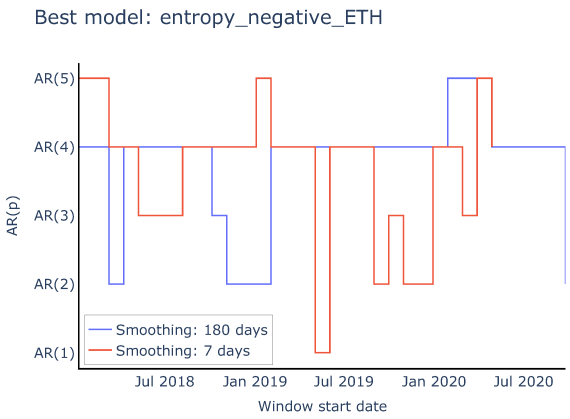
(a) ETH - positive sentiment



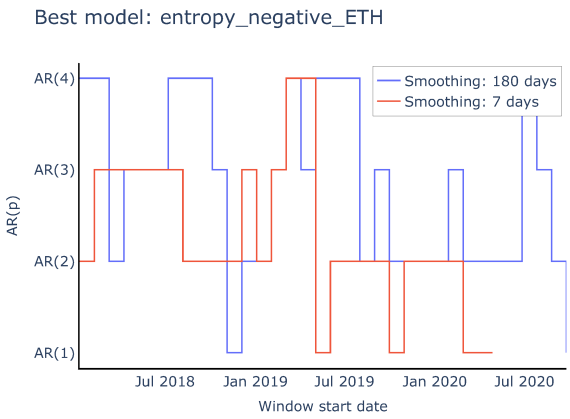
(b) ETH - positive sentiment - $\geq 90\%$ coef. significance



(c) ETH - negative sentiment



(d) ETH - negative sentiment - $\geq 90\%$ coef. significance



C.2 Stage II

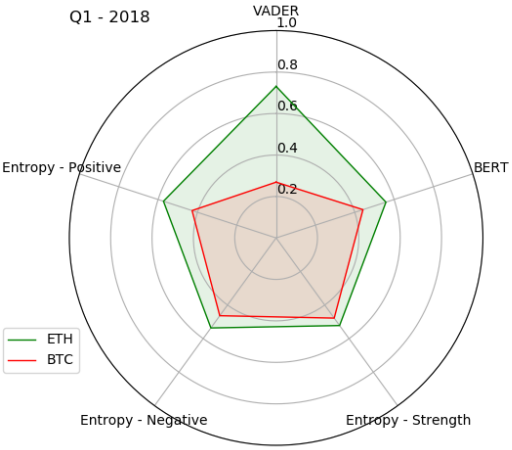
The second stage comprises the use of instrumental variables regression to remove the correlation between the error term and Y_{t-1}. To construct the instrumental variable we have a number of possible regression models which we discuss in detail in Section 6.1.1 of the main manuscript. After trying a range of regression models for the instrumental variable and the residuals of Eq. 6, E_t^y, we will choose the combination that provides both an instrumental variable that is independent from the error term, and a statistically significant fit to E_t^y.

C.2.1 Koyck transform Stage II: Qualitative evidence for the difference between the entropy index and the BERT and VADER indices using the robust median summary

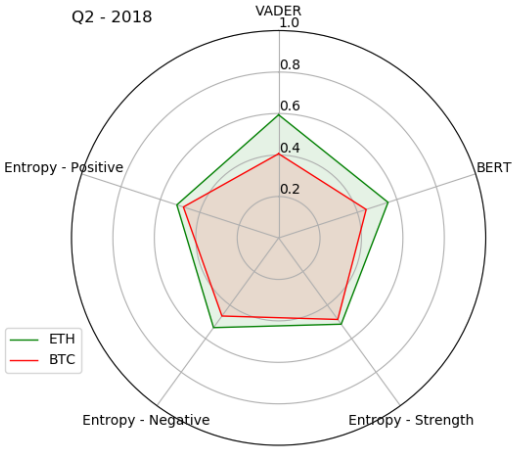
In addition to the IQR volatility-based summary radar plots of the main manuscript, to further illustrate the difference between the signals captured by the examined sentiment constructions we construct the plots of Figures 6 - 9 where we have summarised the sentiment content of the indices in yearly quarters. Figures 6 - 9 we have used the median as a robust summary of the sentiment signal for the corresponding quarters. Looking at Figure 7 (b) we can see that the positive entropy sentiment index exhibits a clear peak in its tendency, showing that it reacted to the sentiment change. However, from Figure 4a of the main manuscript, we observe that BERT captured negative sentiment at that time, whereas VADER captured only a low positive sentiment and mainly fluctuated around neutral.

Figure 6: Sentiment index median per quarter for each of the sentiment indices.

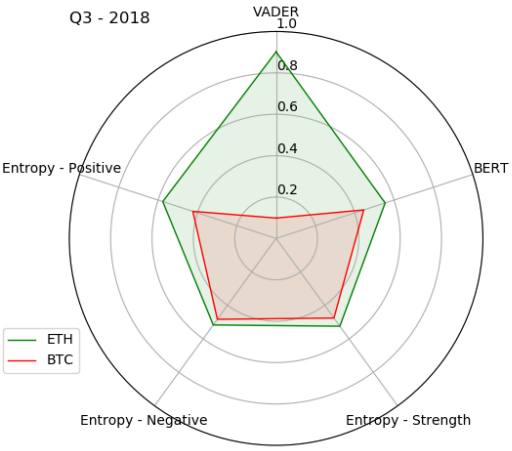
(a) Q1 - 2018



(b) Q2 - 2018



(c) Q3 - 2018



(d) Q4 - 2018

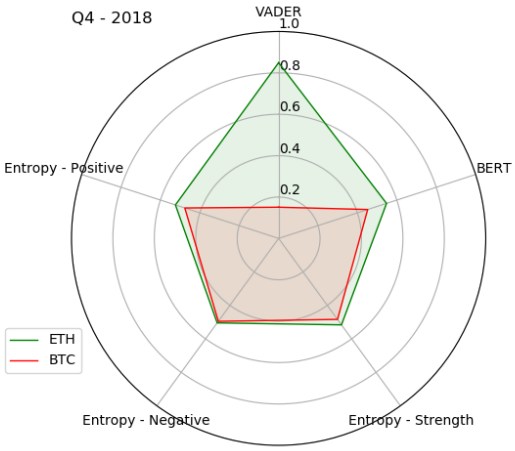
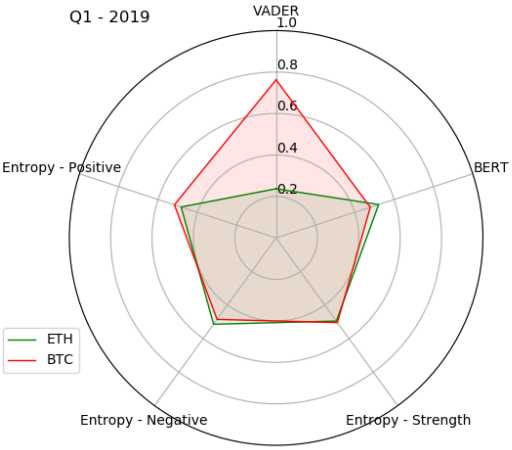
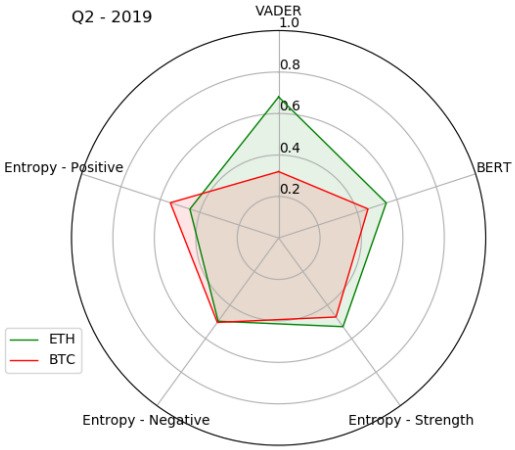


Figure 7: Sentiment index median per quarter for each of the sentiment indices.

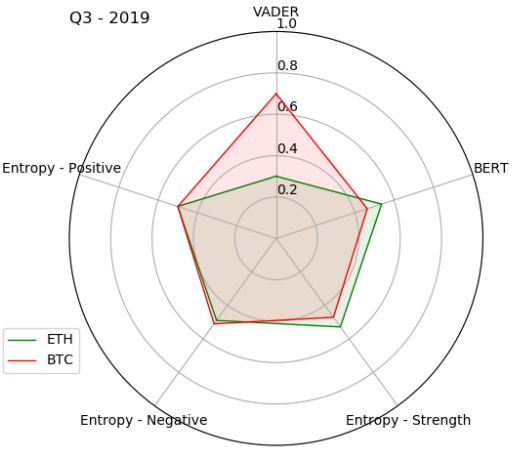
(a) Q1 - 2019



(b) Q2 - 2019



(c) Q3 - 2019



(d) Q4 - 2019

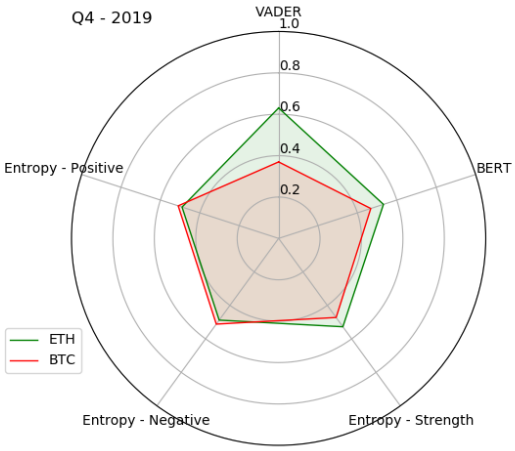
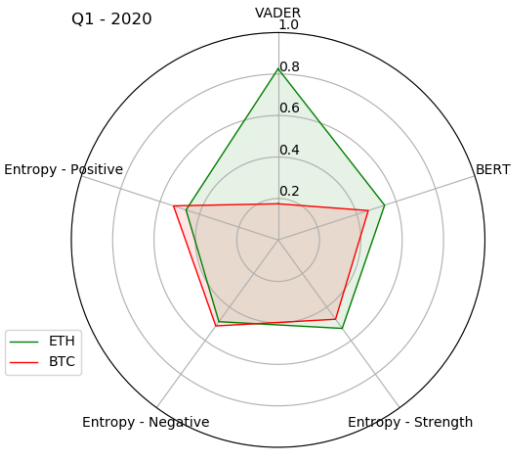
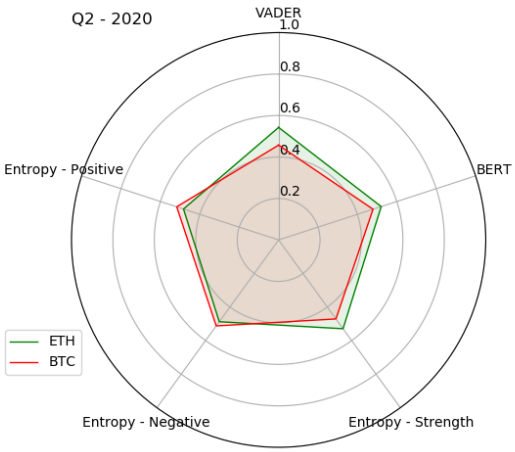


Figure 8: Sentiment index median per quarter for each of the sentiment indices.

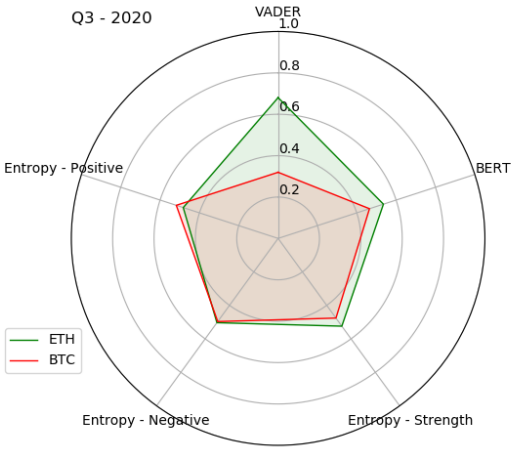
(a) Q1 - 2020



(b) Q2 - 2020



(c) Q3 - 2020



(d) Q4 - 2020

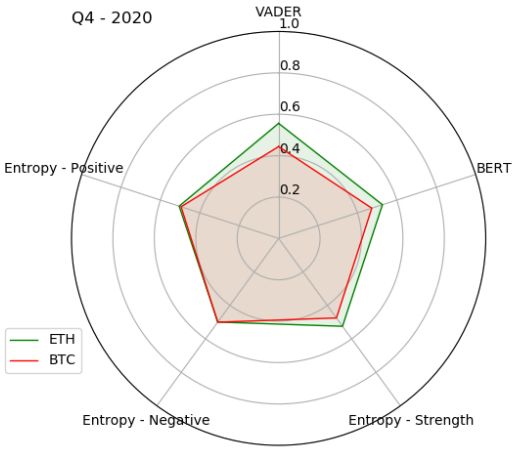
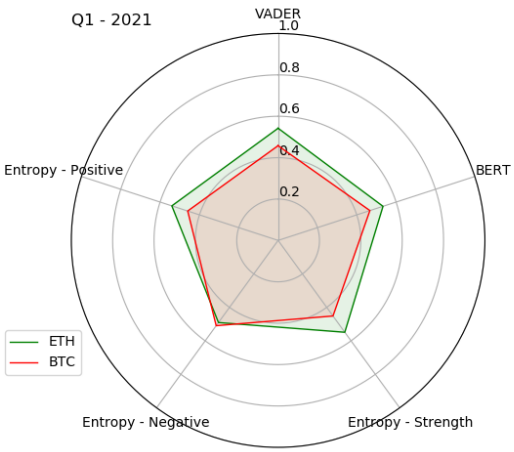
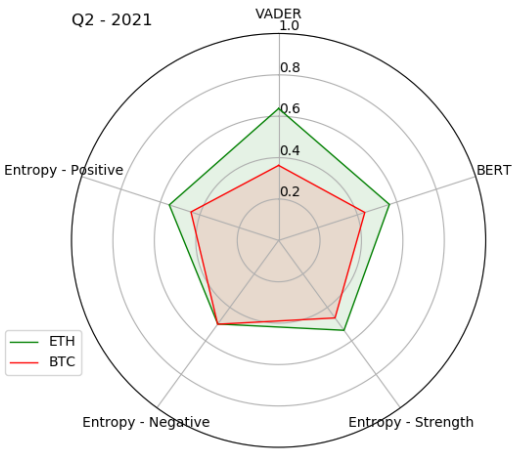


Figure 9: Sentiment index median per quarter for each of the sentiment indices.

(a) Q1 - 2021



(b) Q2 - 2021



D Assessment of regression model over time

In this section, we present additional plots (Figures 10 - 11) for models M1 and M2 that illustrate the model evolution over time, via the coefficients of the Almon lags for the Bitcoin hourly price and the Hash Rate.

Figure 10: M2: BTC hourly close price - Almon coefficients structure over time. The black trace illustrates the coefficients for the model fit on the complete dataset. The red and green lines demonstrate the Almon coefficients from a rolling window fit. For clarity, we show the fits in separate panels. The highlighted entries in the legend show the start of the 1080-day window fit for the traces that are included in the plot. As the windows go into the past, the trace graduates from red to green colour. No Box-Cox transform was applied.

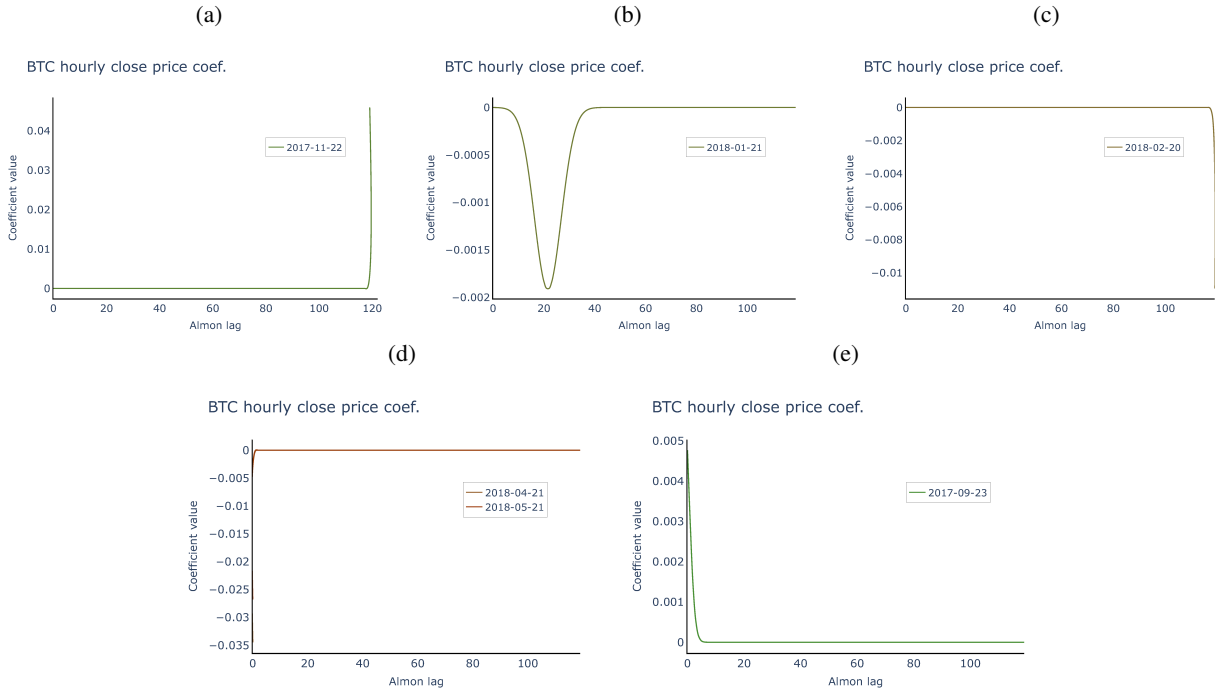
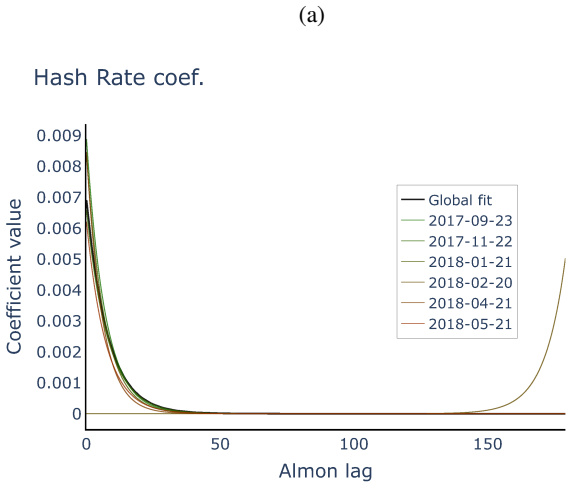


Figure 11: M2: Hash Rate - Almon coefficients structure over time. The black trace illustrates the coefficients for the model fit on the complete dataset. The red and green lines demonstrate the Almon coefficients from a rolling window fit. For clarity, we show the fits in separate panels. The highlighted entries in the legend show the start of the 1080-day window fit for the traces that are included in the plot. As the windows go into the past, the trace graduates from red to green colour. No Box-Cox transform was applied.

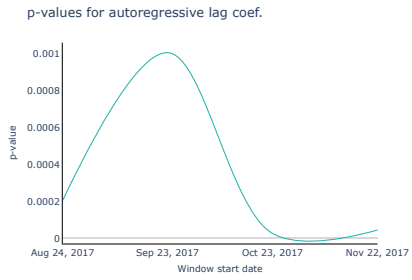


D.1 Statistical significance of interactions between data at mixed frequencies

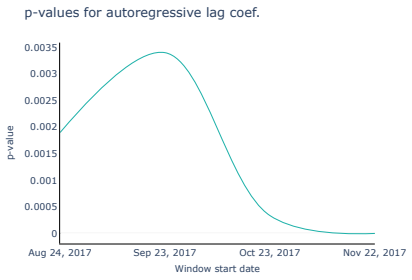
In this section we present further details to explore the statistical significance of the coefficients of the autoregressive covariate, as well as the low- and high-frequency covariates. In Figures 12 - 17 we plot the p-values only when they were statistically significant.

Figure 12: M1: Autoregressive covariate - statistically significant lags. No Box-Cox transform was applied.

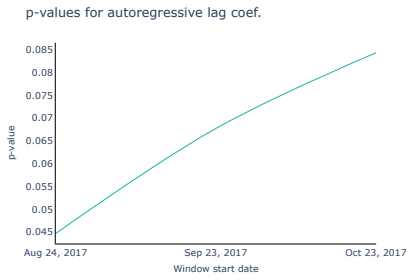
(a) lag 1



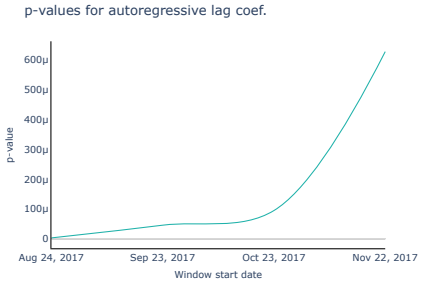
(b) lag 2



(c) lag 3



(d) lag 4



(e) lag 5

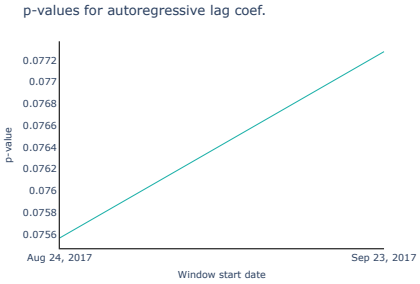
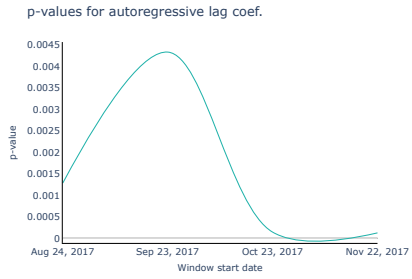
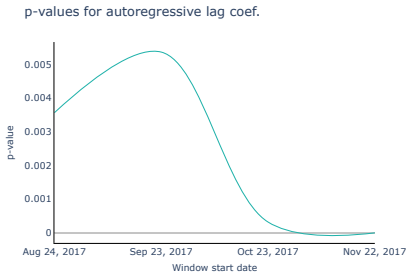


Figure 13: M1: Autoregressive covariate - statistically significant lags. The Box-Cox transform with $\lambda = 0$ was applied.

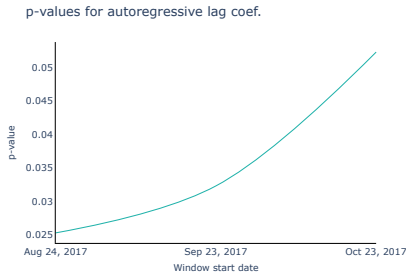
(a) lag 1



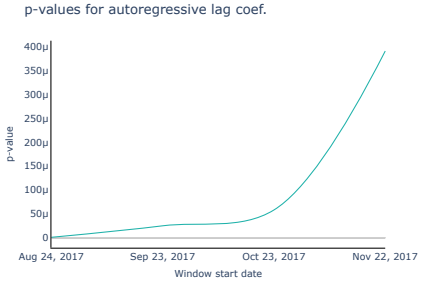
(b) lag 2



(c) lag 3



(d) lag 4



(e) lag 5

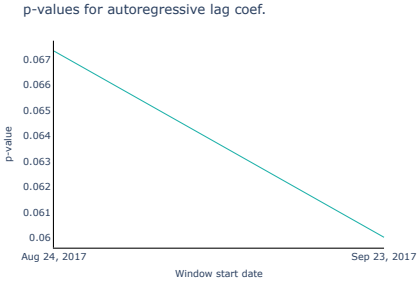
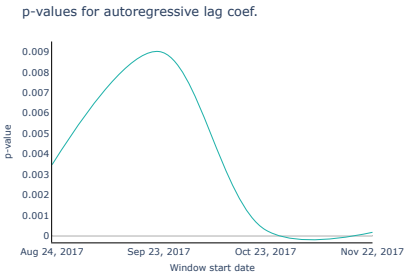
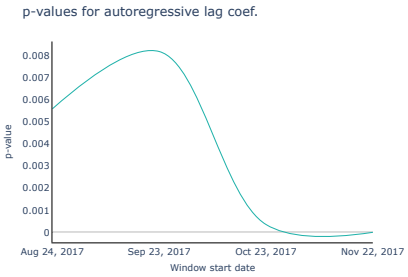


Figure 14: M1: Autoregressive covariate - statistically significant lags. The Box-Cox transform with $\lambda = -0.5$ was applied.

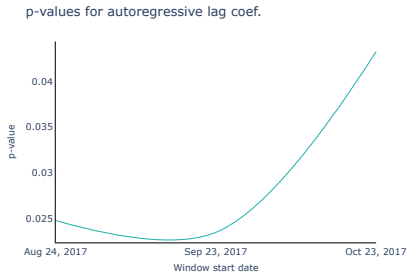
(a) lag 1



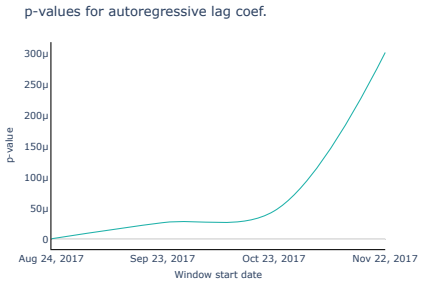
(b) lag 2



(c) lag 3



(d) lag 4



(e) lag 5

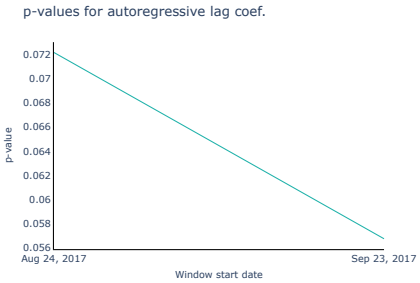
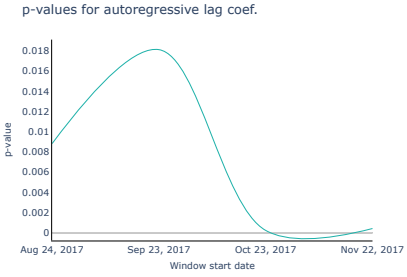
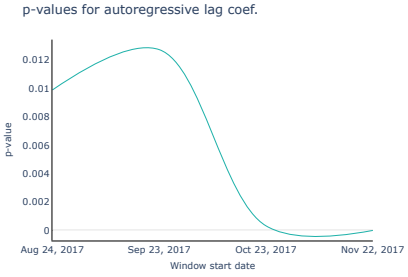


Figure 15: M1: Autoregressive covariate - statistically significant lags. The Box-Cox transform with $\lambda = -1$ was applied.

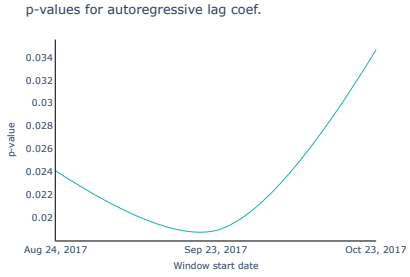
(a) lag 1



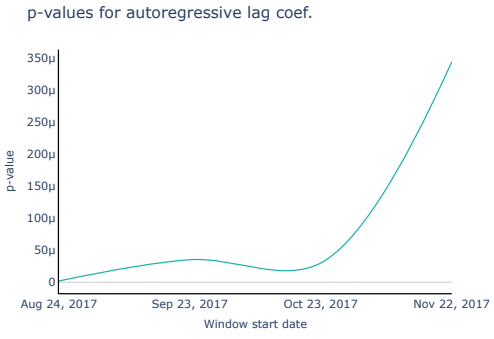
(b) lag 2



(c) lag 3



(d) lag 4



(e) lag 5

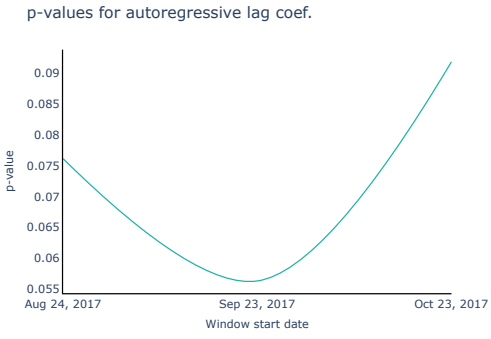


Figure 16: M2: Hash Rate low-freq. covariate - statistically significant Exponential Almon parameters (θ_1, ψ). No Box-Cox transform was applied.

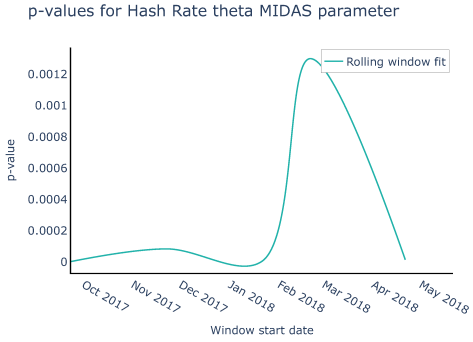
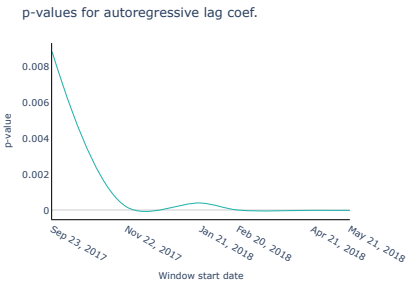
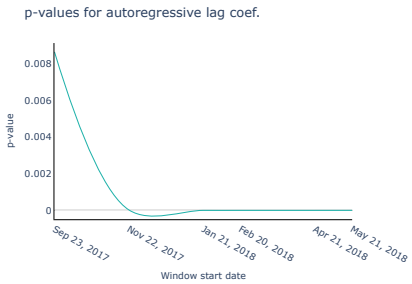


Figure 17: M2: Autoregressive covariate - statistically significant lags. No Box-Cox transform was applied.

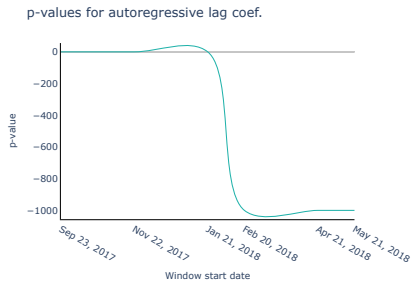
(a) lag 1



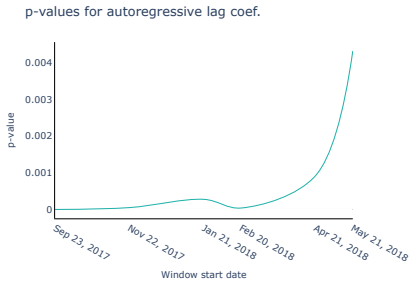
(b) lag 2



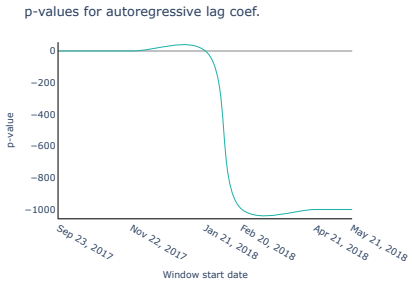
(c) lag 3



(d) lag 4



(e) lag 5

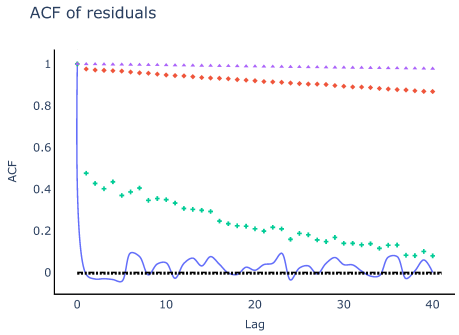


D.2 Mixed-data long memory structures - Autocorrelation functions

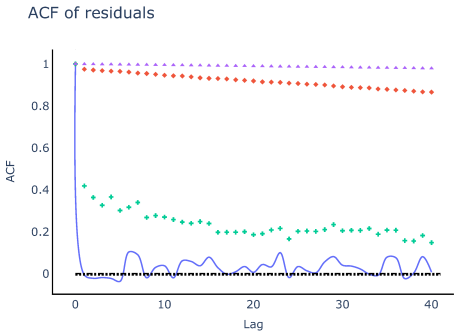
In the case of the long memory Gegenbauer polynomial MIDAS weights, we could estimate the d and u parameters from the residuals using the estimator of Section 5.3 in the main manuscript. The cyclic frequency u can be estimated by observing the autocorrelation plots of the residuals per window: note that the ACF exhibits almost no oscillation around the x axis, which means that $u = 0$, i.e. the long memory is coming from an ARFIMA-type process. We present the ACF plots for models M1 and M2 in Figures 18 - 22.

Figure 18: M1: ACF of residuals per fitting window. The Box-Cox transform with $\lambda = 0$ was applied.

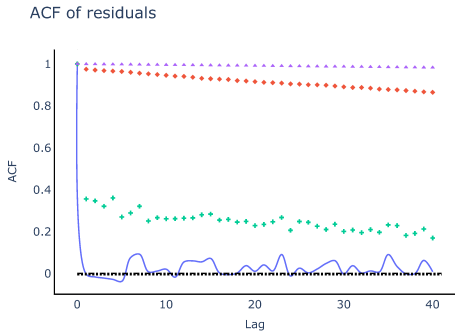
(a)



(b)



(c)



(d)

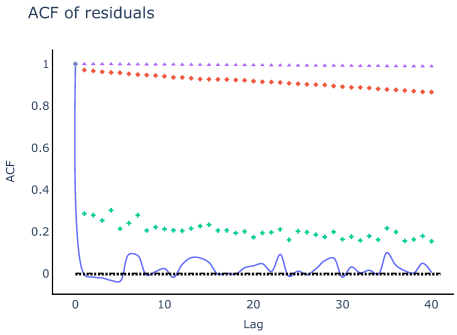
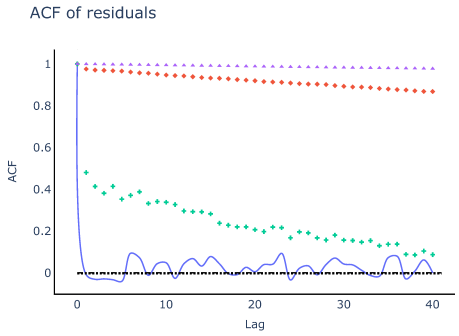
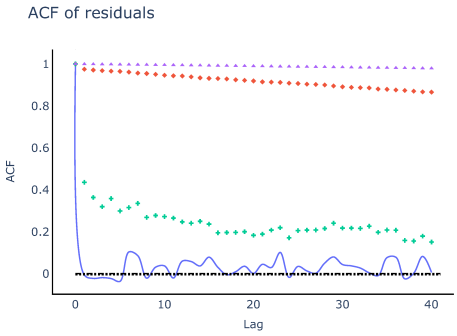


Figure 19: M1: ACF of residuals per fitting window. The Box-Cox transform with $\lambda = -0.5$ was applied.

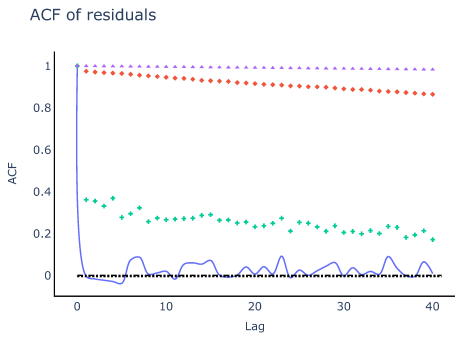
(a)



(b)



(c)



(d)

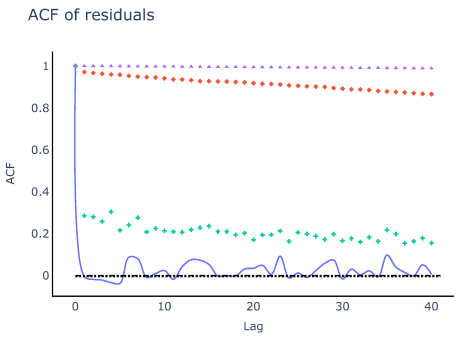
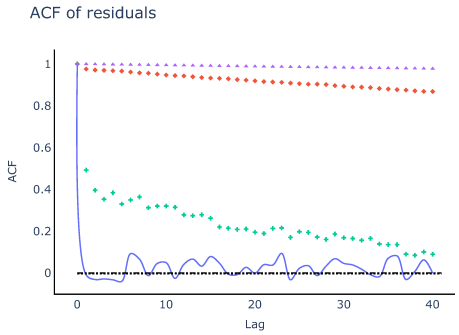
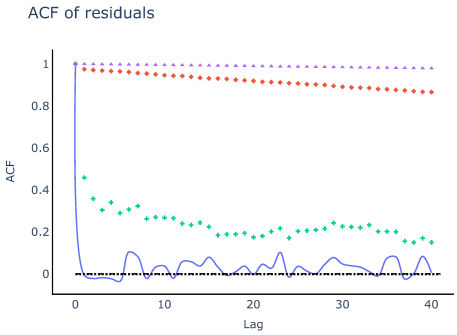


Figure 20: M1: ACF of residuals per fitting window. The Box-Cox transform with $\lambda = -1$ was applied.

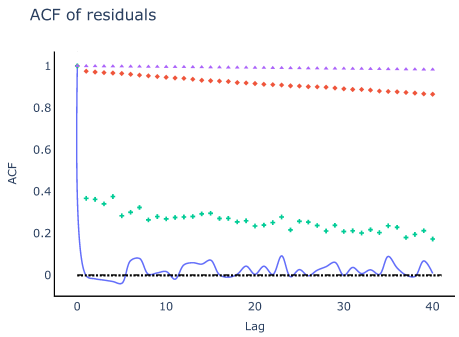
(a)



(b)



(c)



(d)

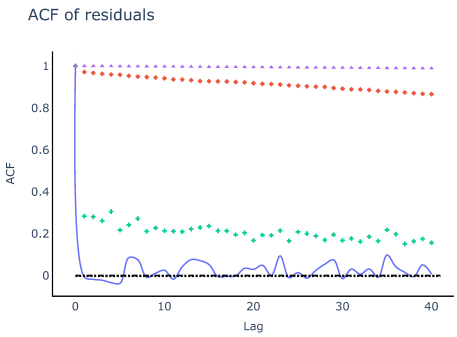
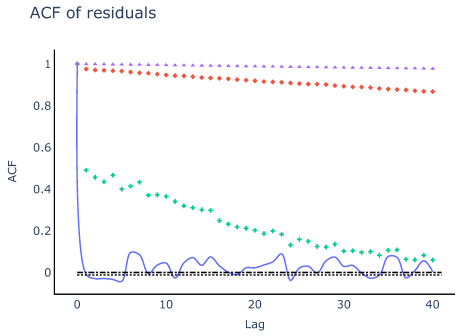
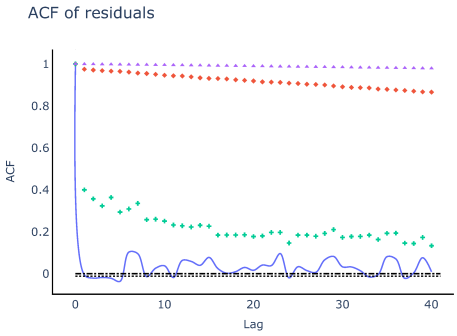


Figure 21: M1: ACF of residuals per fitting window. No Box-Cox transform was applied.

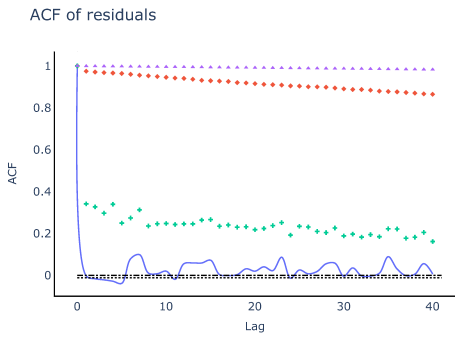
(a)



(b)



(c)



(d)

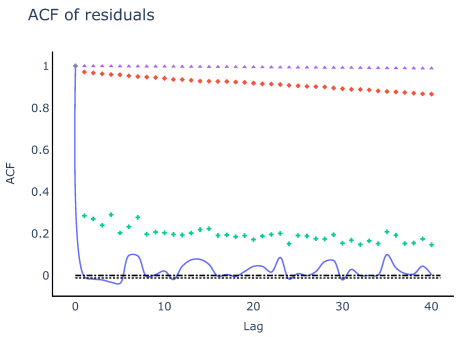
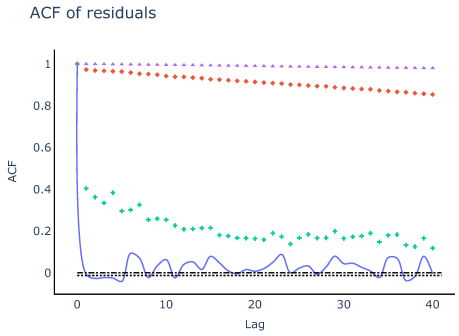
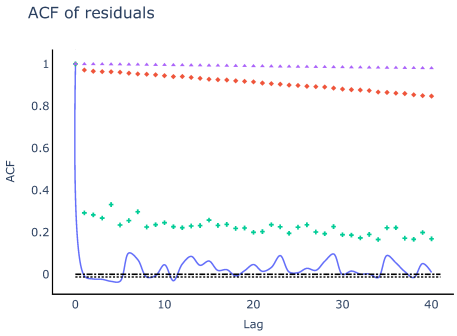


Figure 22: M2: ACF of residuals per fitting window. No Box-Cox transform was applied.

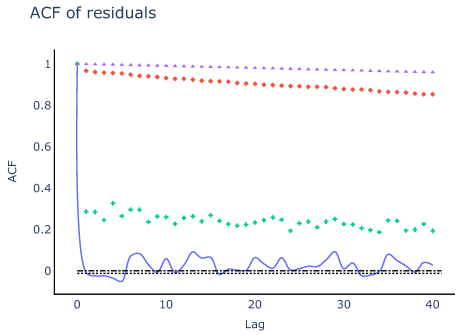
(a)



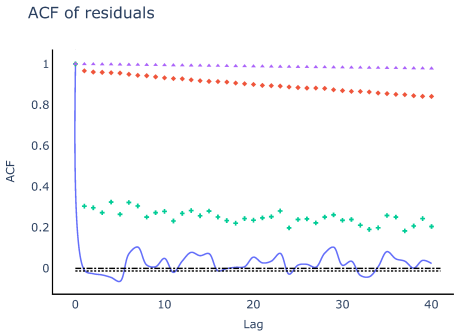
(b)



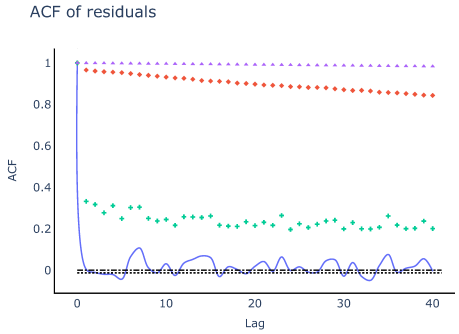
(c)



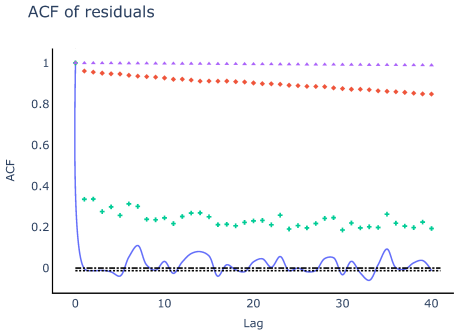
(d)



(e)



(f)



References

[1] H. Akaike, A new look at the statistical model identification, IEEE transactions on automatic control 19 (6) (1974) 716–723.
URL https://doi.org/10.1109/TAC.1974.1100705