UNIL | Université de Lausanne

Unicentre
CH-1015 Lausanne
http://serval.unil.ch

Year : 2022

THREE ESSAYS IN EMPIRICAL ASSET PRICING

Bashchenko Oksana

Bashchenko Oksana, 2022, THREE ESSAYS IN EMPIRICAL ASSET PRICING

Originally published at : Thesis, University of Lausanne

Posted at the University of Lausanne Open Archive http://serval.unil.ch
Document URN : urn:nbn:ch:serval-BIB_E9C82159AF5C4

Droits d'auteur
L'Université de Lausanne attire expressément l'attention des utilisateurs sur le fait que tous les documents publiés dans l'Archive SERVAL sont protégés par le droit d'auteur, conformément à la loi fédérale sur le droit d'auteur et les droits voisins (LDA). A ce titre, il est indispensable d'obtenir le consentement préalable de l'auteur et/ou de l'éditeur avant toute utilisation d'une oeuvre ou d'une partie d'une oeuvre ne relevant pas d'une utilisation à des fins personnelles au sens de la LDA (art. 19, al. 1 lettre a). A défaut, tout contrevenant s'expose aux sanctions prévues par cette loi. Nous déclinons toute responsabilité en la matière.

Copyright
The University of Lausanne expressly draws the attention of users to the fact that all documents published in the SERVAL Archive are protected by copyright in accordance with federal law on copyright and similar rights (LDA). Accordingly it is indispensable to obtain prior consent from the author and/or publisher before any use of a work or part of a work for purposes other than personal use within the meaning of LDA (art. 19, para. 1 letter a). Failure to do so will expose offenders to the sanctions laid down by this law. We accept no liability in this respect.

UNIL | Université de Lausanne

FACULTÉ DES HAUTES ÉTUDES COMMERCIALES

DÉPARTEMENT DE FINANCE

THREE ESSAYS IN EMPIRICAL ASSET PRICING

THÈSE DE DOCTORAT

présentée à la

Faculté des Hautes Études Commerciales
de l'Université de Lausanne

pour l'obtention du grade de
Docteure ès Sciences Économiques, mention « Finance »

par

Oksana BASHCHENKO

Directeur de thèse
Prof. Eric Jondeau

Jury

Prof. Christian Zehnder, Président
Prof. Roxana Mihet, expert interne
Prof. Pierre Collin-Dufresne, expert externe

LAUSANNE
2022

UNIL | Université de Lausanne

FACULTÉ DES HAUTES ÉTUDES COMMERCIALES

DÉPARTEMENT DE FINANCE

THREE ESSAYS IN EMPIRICAL ASSET PRICING

THÈSE DE DOCTORAT

présentée à la

Faculté des Hautes Études Commerciales
de l'Université de Lausanne

pour l’obtention du grade de
Docteure ès Sciences Économiques, mention « Finance »

par

Oksana BASHCHENKO

Directeur de thèse
Prof. Eric Jondeau

Jury

Prof. Christian Zehnder, Président
Prof. Roxana Mihet, expert interne
Prof. Pierre Collin-Dufresne, expert externe

LAUSANNE
2022

UNIL | Université de Lausanne
HEC Lausanne
Le Décanat
Bâtiment Internef
CH-1015 Lausanne

IMPRIMATUR

Sans se prononcer sur les opinions de l'autrice, la Faculté des Hautes Etudes Commerciales de l'Université de Lausanne autorise l'impression de la thèse de Madame Oksana BASHCHENKO, titulaire d’un bachelor en Mathématiques de l’Université Nationale Taras Chevtchenko de Kiev, titulaire d’un master en Mathématiques Actuarielles et Financières de l’Université Nationale Taras Chevtchenko de Kiev, titulaire d’un master en Mathématiques Appliquées à l’Economie et à la Finance de l’Université Paris 1 Panthéon-Sorbonne, en vue de l'obtention du grade de docteure ès Sciences économiques, mention « finance ».

La thèse est intitulée :

THREE ESSAYS IN EMPIRICAL ASSET PRICING

Lausanne, le 17 juin 2022

La Doyenne

Marianne SCHMID MAST

Members of the thesis committee

Prof. Eric JONDEAU
Professor of Finance, Faculty of Business and Economics, University of Lausanne and Swiss Finance Institute
Thesis supervisor

Prof. Roxana MIHET
Assistant Professor of Finance, Faculty of Business and Economics, University of Lausanne and Swiss Finance Institute
Internal member of the doctoral committee

Prof. Pierre COLLIN-DUFRESNE
Professor of Finance, École Polytechnique Fédérale de Lausanne and Swiss Finance Institute
External member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

PhD in Economics,
Subject area Finance

I hereby certify that I have examined the doctoral thesis of

Oksana BASHCHENKO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: ______________________________ Date: May 6, 2022

Prof. Eric JONDEAU
Thesis supervisor

University of Lausanne
Faculty of Business and Economics

PhD in Economics,
Subject area Finance

I hereby certify that I have examined the doctoral thesis of

Oksana BASHCHENKO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: ______________________________ Date: 06-05-2022

Prof. Roxana MIHET
Internal member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

PhD in Economics,
Subject area Finance

I hereby certify that I have examined the doctoral thesis of

Oksana BASHCHENKO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: Date: _May 8, 2022__

Prof. Pierre COLLIN-DUFRESNE
External member of the doctoral committee

Acknowledgements

I would like to thank my advisor Eric Jondeau for his supervision. I had full research freedom, being able to work on the topics I find the most inspiring, and at the same time, my supervisor was always there when I needed his support. I also thank Pierre Collin-Dufresne and Julien Hugonnier for the valuable discussions. I am grateful to Roxana Mihet for our fruitful collaboration on Fincap project. I also appreciate the administrative support of Catherine and Christina. I thank all professors from the Swiss Finance Institute for all the knowledge they shared with us via courses, talks, and seminars. I appreciate being a part of the Swiss Finance Institute research environment and thank SFI for its support.

My special gratitude goes to Alexis for the plenty of deep and thought-provoking discussions. Some of them evolved from a few sentences said over lunch into the chapters of this thesis, and I have really enjoyed working together on our co-authored papers. I appreciate the multiple conversations I had with my officemates Zhimin, Martina, and Ina. I am grateful for Ina's and Irina's support during the first years of PhD, as well as for our yoga mornings, plain-air sessions, and improvisation evenings. I thank my colleagues and PhDnet for all the adventures of the PhD life we shared together.

I am grateful to my friend Olha, who was always there for me, despite being on the other continent. And I have especially appreciated her optimism during these challenging times of lockdowns and war. Finally, I would like to thank my family: my parents and my brother - for all their love and support, and for always believing in me. It gave me the strength and perseverance to pursue my research goals.

Lausanne, June 14, 2022 O. B.

Abstract

This thesis consists of three applications of machine learning techniques to empirical asset pricing.

In the first part, which is co-authored work with Alexis Marchal, we develop a new method that detects jumps nonparametrically in financial time series and significantly outperforms the current benchmark on simulated data. We use a long short-term memory (LSTM) neural network that is trained on labelled data generated by a process that experiences both jumps and volatility bursts. As a result, the network learns how to disentangle the two. Then it is applied to out-of-sample simulated data and delivers results that considerably differ from the benchmark: we obtain fewer spurious detection and identify a larger number of true jumps. When applied to real data, our approach for jump screening allows to extract a more precise signal about future volatility.

In the second part, which is co-authored work with Alexis Marchal, we develop a methodology for detecting asset bubbles using a neural network. We rely on the theory of local martingales in continuous-time and use a deep network to estimate the diffusion coefficient of the price process more accurately than the current estimator, obtaining an improved detection of bubbles. We show the outperformance of our algorithm over the existing statistical method in a laboratory created with simulated data. We then apply the network classification to real data and build a zero net exposure trading strategy that exploits the risky arbitrage emanating from the presence of bubbles in the US equity market from 2006 to 2008. The profitability of the strategy provides an estimation of the economical magnitude of bubbles as well as support for the theoretical assumptions relied on.

In the third part, I propose a new methodology to construct interpretable, fundamental-based pricing factors from news to explain Bitcoin returns. Each news article from a specialized cryptocurrency website is classified in a semi-supervised manner into one of the few predefined topics. Topic sentiments become factors contributing to the price variation. I use a cutting-edge NLP algorithm (SBERT network) to embed linguistic data into a vector space, which allows the application of an intuitive classification rule. This approach permits the exclusion of news pieces that describe the price movements per se from the analysis, thus mitigating endogeneity concerns. I show that non-endogenous news contains fundamental information about Bitcoin. Thus I reject the concept of Bitcoin price being based on pure speculation and show that Bitcoin returns are partially explained by fundamental topics. Among those, the adoption of cryptocurrencies and blockchain technology is the most important aspect. On top of that, I study the media expressed attitude toward Bitcoin from the functions of money perspective. I show that investors consider Bitcoin as the store of value rather than the medium of exchange.

Keywords: Asset pricing, machine learning, volatility modelling, LSTM, natural language processing, BERT.

Contents

List of Figures

List of Tables

Introduction

This thesis consists of three chapters. One global approach unites these papers: in each of them, a deep learning algorithm is used to provide insights about asset prices. While the applications of machine learning algorithms in finance have gained great popularity, these methods are sometimes frowned upon. The main critique point is the fact that AI is a "black box" and thus offers limited research insights. In this thesis, I show that this is not necessarily the case. I use deep neural networks rather as a tool that solves particular clearly formulated data-related tasks (different for each chapter).

In the first chapter, we use a long short-term memory (LSTM) network to detect jumps in high-frequency asset prices. As economic data can be observed only at discrete time points, it creates the need for a classification algorithm. The existing statistical benchmark is able to handle the task under the assumption of no abrupt changes in volatility, which is demonstrated to be violated in the real data. Thus we apply the network to detect jumps in asset prices and show its superior performance.

In the second chapter, we use an LSTM network together with a theory from mathematical finance to identify bubbles in the asset price. In this application, we estimate the functional form of the volatility at every time point, and this estimation is further used to conclude whether the asset experiences a bubble or not, relying on the martingale theory of asset bubbles.

These two papers are united by the fact that the network efficiently solves the problem of rolling window estimation in a regime switching framework. Both our applications require an estimation of the volatility of a continuous time stochastic process at every point in time. For this purpose, it is natural to process data in batches and use a rolling window estimator. The statistical tests rely on the fixed size rolling window approach and work relatively well under the assumption that volatility stays in one regime during the whole period of observation. In this case, the approach leads to good estimates, as the process always stays in the same regime, and thus all points used for the estimation come from the one regime. However, the accuracy of this method decreases significantly in a regime shifting environment, especially when the changing points are random. In this case, the choice of rolling window size becomes a challenging task. If the window size is kept fixed, then at some point the rolling window estimator will use the data from two different regimes. This results in an incorrect estimate of the parameter around the regime shift. If the window size is large, then this situation may happen more frequently, and the estimator may not be able to detect some regimes, especially the ones that quickly pass. At the same time, choosing a small window size is not a remedy: such estimator will be able to react to the regime changes faster, but it will be very noisy since only a few observations are used. It is especially problematic in finance, where the data is known to contain a substantial amount of noise. The neural network is able to overcome this problem. Intuitively, it splits data into chunks of the same regime. Thus, it can optimally adjust the window size, and only after this estimate the parameters. In a given identified regime, it includes all the data points for the estimation, maximizing the number of observations being used to increase the estimation quality. At the same time, it avoids using data from another regime.

In the first two chapters, we relied purely on price time series to draw inferences about the underlying data-generating process. In the third chapter, I use a deep learning algorithm to extract signals from text data. I focus on the cryptocurrency market, which poses additional challenges for the pricing, as the concept of fundamentals for this asset is unclear. I use the transformer network (SBERT) to classify media articles into topics and to detect their sentiment. I show that the sentiment of the specific group of articles is consistent with the information theory of media tone, and thus conclude that it carries fundamental information. I am able to explain more than 16% of Bitcoin return variation with the fundamental information, thus rejecting the idea that Bitcoin is a purely speculative asset. On top of that, I examine the media articles to understand which Bitcoin function is the most valued by the investors. I conclude that investors mostly appreciate its store of value property.

1 Deep Learning, Jumps, and Volatility Bursts (co-authored with Alexis Marchal)

1.1 Introduction

A popular stochastic process used to describe the evolution of prices is the so called jump-diffusion model. Under this specification the price is modeled as combination of a drift, a Brownian term and a jump process. We are interested in classifying returns into innovations coming from the continuous Brownian or the discontinuous jump process. However, data being inherently observed at discrete time points the decomposition is not straightforward. At a given frequency, a large Brownian increment will be indistinguishable from a small size jump. This is why jumps are often mistaken for bursts of volatility (i.e. volatility jumps) at "reasonable" frequencies (around 5 minutes). One solution to this problem is to select the highest frequency possible (use so called ultra high-frequency data). Given the continuity of the Brownian component, diffusion-driven innovations will become smaller as the sampling frequency increases, letting the true jumps emerge and making them easier to be detected. The drawback of this approach is that for the highest frequencies the financial data is prone to be contaminated with microstructure noise.

This paper presents a new method that disentangles jumps nonparametrically and is able to separate them from bursts of volatility. Our approach detects the exact time of a jump within a day and is therefore benchmarked to the fundamental test of Lee and Mykland (2008) (LM henceforth). Our neural network largely outperforms the benchmark in presence of jumps inside the volatility of the price process, achieving smaller type I and type II errors, and performs equally good with continuous volatility. In order to achieve these results we use a long short-term memory network. It is first trained on labelled data that were generated from Monte-Carlo simulations. The network is then applied to classify every single return on out-of-sample simulated data.

On real data, since labels are not available we assess the performance of our method via an event study and a volatility forecasting exercise. Both of them confirm the higher accuracy of our approach.

The rest of the paper is organized as follows. Section 1.2 describes the literature. Section 1.3 compares our method with the "classical" statistical tests and presents the main benchmark. Section 1.4 explains the main idea behind the architecture and the training of the network. Section 1.5 assesses the performance of our method on out-of-sample simulated data. Section 1.6 applies the algorithm to classify real data. Finally section 1.7 concludes.

1.2 Literature review

The importance of differentiating between the two sources of risk, the jump component and the continuous Brownian part, is outlined in Aït-Sahalia (2004). The existence of jumps impacts option prices, risk management and asset allocation. Early papers dedicated to the testing for jumps presence proposed a parametric approach using jump-diffusion models with constant volatility. More recent papers have added a stochastic volatility component and state-dependent jump parameters. The estimation for those models, however, is complex and subject to model errors.

A separate branch of literature has focused on nonparametric methods working with intraday data instead of the previously used low-frequency observations. Barndorff-Nielsen and Shephard (2004) introduced the notion of bipower variation (BPV) which is a nonparametric estimator of integrated variance that is consistent in presence of jumps. It established the foundations for multiple nonparametric tests that were proposed afterwards.[1] However most of the tests only allow to assess the continuity of the sample path during a given time period, they do not aim at detecting the exact jump times.

Lee and Mykland (2008) develop a model-free test that identifies the precise moment a jump occurred within a day. For now, it (with some adjustments like in Boudt et al. (2011) that takes into account seasonality) remains the workhorse for this type of task (see the comprehensive reviews of Theodosiou and Zikes (2012) and Mukherjee et al. (2019)).[2] Such test can be useful for insider trading detection, studying the time-series of jumps (arrival in the morning/afternoon, self-exciting, presence of systematic/co-jumps risk) to validate assumptions of asset pricing models, or reduce the number of parameters (regarding stochastic intensity, correlation, and jump size distribution) for estimations . Dumitru and Urga (2012) offer a Monte-Carlo comparison of nine jump tests, concluding that LM demonstrates the best performance. Consequently, we consider LM as the most widely used benchmark to which we compare our results.

Lee and Mykland (2008) suggest to use data sampled not more frequently than at 2 minutes since above that the data would likely be contaminated with market microstructure noise.

[1] The first test using bipower variation was developed by Barndorff-Nielsen and Shephard (2006). A generalized concept (multipower variation) is studied in Barndorff-Nielsen et al. (2006) Building on the previous findings, Corsi et al. (2010) and Podolskij and Ziggel (2010) introduced new jumps tests. Other well-known jump tests include Andersen et al. (2009) (using median realized volatility) and Jiang and Oomen (2007) (using swap variance).

[2] Andersen et al. (2007b) also develop a jump test for each individual return similar to Lee and Mykland (2008) but assume constant intraday volatility.

Christensen et al. (2014) is the first paper to test for the presence of jumps exploiting tick-by-tick data. They conclude that jumps are in fact less common than previously thought and that the tests at lower frequencies spuriously identify bursts of volatility as jumps. We are able to confirm this finding, though working with 2 minutes frequency data and avoiding microstructure noise filtering concerns.

Machine learning algorithms in finance are gaining an overwhelming popularity. Nonetheless, the exploration of jumps with those algorithms did not receive much attention. To the best of our knowledge, the only two papers in this area are the following. Mäkinen et al. (2018) use a recurrent neural network to predict future jumps in prices. However, the training of the network takes as input the jumps already classified by the LM test on real data. They therefore do not develop a new classification tool. The work of Au Yeung et al. (2019) aims to test for jumps (defined by them as regime switching points) using machine learning. We differ from them both in terms of research question and methodology. First, we are able to separate jumps from volatility bursts, which is impossible in their framework. Second, unlike them we train our network on simulated data, instead of using (necessarily misclassified) real data. This allows for clear pattern recognition and thus better performance.

1.3 Machine learning vs. statistical test

We start by motivating why the existing statistical methods might fail to classify jumps as such.

The most popular nonparametric individual jump test developed in Lee and Mykland (2008) is based on the estimator of spot volatility, that is consistent in the presence of jumps. Unlike the realized variance (RV), that measures the total variance of the process and is unable to separate the continuous and jump variations, the bipower variation (BPV) is capable of estimating solely the diffusion variance, ignoring variation of the jump part.[3] Thus, it can be used to construct a statistical test to detect jumps.

Loosely speaking, the LM test classifies a data point as a jump if the return size standardized by estimation of the instantaneous volatility [4] is too high in absolute value. Though model-free, this test requires the data-generating process to satisfy some regularity conditions. The crucial assumption is that the price is represented as a jump-diffusion process

$$d \log S_t = \mu_t dt + \sigma_t dB_t + Y_t dq_t \tag{1.1}$$

with the drift μ_t and the diffusion coefficient σ_t not changing dramatically over a short period

[3] The interested reader can find more details about volatility estimators in appendix B.

[4] Formally, the estimate of the spot volatility is a scaled version of the bipower variation and is computed as $\hat{\sigma}^2(t_i) = \frac{1}{K-2} \sum_{j=i-K+2}^{i-1} |\log S(t_j)/S(t_{j-1})| |\log S(t_{j-1})/S(t_{j-2})|$ where K is the chosen window size and S_t is the spot price of the stock at time t.

of time.[5] From now on we will use interchangeably the notions of diffusion coefficient and (spot) volatility.

This assumption is the main weak point of the test. Its violation (for example, if σ_t itself contains jumps) leads to a significant increase in the amount of spurious detection. Indeed, the test classifies a return as a jump if the absolute value of the test statistic

$$\mathscr{L}(i) = \frac{\log \frac{S(t_i)}{S(t_{i-1})}}{\hat{\sigma}(t_i)} \tag{1.2}$$

is above some threshold. If volatility is not allowed to change dramatically, a high test statistic would indeed reflect a jump. However, as soon as we allow for jumps in the volatility itself, it is not the case anymore. $\hat{\sigma}$ being computed over a moving window, if the diffusion coefficient changes dramatically, the bipower variation will take time to incorporate this change. Keeping this in mind, there are two different scenarios under which the LM test can fail. The first is a positive jump in σ_t, that may generate a high purely diffusion-driven return. Together with the bipower variation that did not have time to adjust for the new high level of volatility and thus stays relatively low, this results in a high test statistic value. LM is unable to distinguish between this scenario and a true jump in the price process, spuriously classifying both as jumps. The second scenario is a sudden volatility decrease. The LM test might not be able to find a real jump following this drop. For the same reason as above, the BPV is not able to incorporate the change immediately. So it may stay higher than the real volatility, lowering the test statistics and covering the true jump. This violation causes a significant misclassification rate. When σ_t contains jumps, the spurious detection rate of the LM test in simulated data is above 150%.

Moreover, the assumption of no dramatic change in volatility is rejected by the real data. As shown in Tauchen and Todorov (2008), volatility of the asset price necessarily contains jumps and they happen much more frequently than just few times per year. So the core assumption of the LM test contradicts the inherent feature of the market data.

Another concern comes from consistency of the bipower variation. As stated in Barndorff-Nielsen and Shephard (2006), the BPV is a consistent estimator of the spot volatility only in absence of leverage effect. The authors acknowledge that this is an unfortunate but important restriction of their results, that confronts the stylized facts of equity data.

To overcome these significant limitations, we are the first to propose using an LSTM network for jump identification and distinguishing them from discontinuous changes in volatility. The

[5] Strictly speaking, the assumption from Lee and Mykland (2008) is $\forall \epsilon > 0$

$$\sup_i \sup_{t_i \le u \le t_{i+1}} |\mu(u) - \mu(t_i)| = O_p(\Delta t^{\frac{1}{2}-\epsilon}),$$

$$\sup_i \sup_{t_i \le u \le t_{i+1}} |\sigma(u) - \sigma(t_i)| = O_p(\Delta t^{\frac{1}{2}-\epsilon}).$$

main advantage of our approach is that we are not limited by any regularity conditions in contrast to the classical statistical tests. Instead, we train the algorithm on data generated by multiple processes and specifications of any desirable complexity, that incorporate stylized facts about price characteristics, such as leverage effect, volatility clustering, volatility jumps etc. As a result, we consistently outperform Lee and Mykland (2008) in the out-of-sample analysis when simulated data is more realistic, and perform comparably to them when their assumptions are met.

1.4 Network architecture and training

The long short-term memory network we choose for jump classification is a special type of recurrent neural network that was introduced in Hochreiter and Schmidhuber (1997). By construction, LSTM networks are well suited for remembering long-term dependencies and thus for handling time-series data. We use a standard LSTM, motivated by the main finding of Greff et al. (2016). They present the largest comparison study for different architectures of LSTM networks and show that improvements of different modifications over the plain vanilla LSTM are marginal. Our network consists of five layers: an input layer, a bidirectional LSTM layer with 200 hidden neurons, a fully connected layer, a softmax layer, and a classification output layer. An interested reader may refer to appendix A to find a brief intuition about neural networks in general and LSTM in particular, as well as a detailed layers description. We simply select a plain vanilla structure which is detailed in appendix A. We have a total of 82 002 parameters that are trained on 175 years of high-frequency data at 2min which implies that we use roughly 100 data points per parameter to avoid overfitting. Given the very large imbalance of the classes in this dataset, the accuracy has to be extremely high for the network to be considered properly trained.

Now that the structure of the network is chosen, it has to be trained to detect jumps. Our methodology is conceptually simple. We first generate training data and since it is simulated, we know exactly when a jump occurred. Then we label each return as “jump” or “no jump” and train the network on them. Finally the network is ready to classify new time-series.

To create the training data, we simulate paths of the price process by discretizing the stochastic differential equation (SDE) governing it. An advantage of our method is that since we do not have to derive the distribution of a test statistic, there is no restriction on the data-generating process. Therefore the training data set as a whole can be composed by multiple time-series possibly generated from different processes of any complexity. Using a variety of them allows the network to concentrate only on the specific feature (jump or not) of the data point, preventing over-fitting. Another benefit of this method is that we can generate virtually an unlimited quantity of labelled data to train on, enhancing the network performance at no cost[6].

[6] Apart from the necessary computational time.

In order to incorporate many stylized facts of equity data, we have chosen the following specification for the stock price S_t and its diffusion coefficient σ_t:

$$dS_t = S_t(\mu_t dt + \sigma_t dB_{1t} + dJ_{1t}), \tag{1.3}$$

$$d\sigma_t = \alpha(\overline{\sigma} - \sigma_t)dt + \upsilon\sqrt{\sigma_t}dB_{2t} + dJ_{2t}. \tag{1.4}$$

The price follows a jump-diffusion model and we introduce a rich structure for the stochastic volatility which is governed by a mean-reverting process with jumps. The jump component J_{2t} introduces bursts of volatility, defined as a sudden change in the diffusion coefficient of the price. This specification for the stochastic volatility allows us to incorporate the conventional models with finite activity jumps [7]. The sources of volatility risk B_{2t} and J_{2t} can be (negatively) correlated with B_{1t} and J_{1t} to incorporate the leverage effect. In order to realistically reproduce real life patterns of securities, it is important to take all of that into account. Using this framework the network will learn how to identify jumps and disentangle them from bursts of volatility.

In order to simulate sample paths we need to select values for the parameters used in the data-generating process(es). One idea could be to take real financial data, estimate those parameters (for instance using maximum likelihood) and simulate our labelled data using them. This procedure has two drawbacks. First, there is an estimation risk. Second, financial markets are ever changing. This means that those parameters could have multiple regimes and change over time. To overcome these difficulties we train the network on a whole set of parameters. By constructing a realistic set, the network will learn what jumps look like under various environment. This gives us hope that whatever regime the market is in, the network will perform decently in the real data. In this way we also avoid re-training the network often.

Clearly, this method is not limited to using SDEs of the form of (1.3) and (1.4). The procedure would work using any data-generating process to create labelled data and then following the same steps.

1.5 Performance on simulated data

In this section we assess the performance of our method using out-of-sample simulated data. We start by generating new time-series, using parameters that the network was never trained on. For every number reported in the tables below, the network was tested using 2000 Monte-Carlo trials, each individual trial consisting of data generated at a 2 minutes frequency for 0.5 year.

First of all, we compare our network to the benchmark of Lee and Mykland (2008) when the

[7] For further details see Cont and Tankov (2004).

data satisfies their assumption about “local” changes in spot volatility. Table 1.1 presents the percentage of correct (% detection)[8] and spurious (% spurious)[9] classification of jumps by our network and the benchmark. We see that we perform roughly as good as the benchmark in this case. The good performance of LM is explained by the fact that all of their assumptions are met. But as pointed out by Tauchen and Todorov (2008) this is not a realistic framework to describe real world high-frequency equity data. Indeed, spot volatility should contain jumps.

Our network starts to significantly outperform in all dimensions (type I & type II errors) as soon as we introduce jumps inside the diffusion coefficient. This violates the main assumption of LM and it explains the drastically different results described in tables 1.2 and 1.3.

	Network	Lee & Mykland (benchmark)
% detection	88.32	94.15
% spurious	2.1	0.38

Table 1.1 – This table compares the performance of our method versus the benchmark. “%detection” stands for the percentage of correctly identified jumps. “%spurious” stands for the percentage of spurious jumps identified. The simulated data does not contain jumps inside the volatility (intensity of volatility jump arrival is 0).

	Network	Lee & Mykland (benchmark)
% detection	92.87	88.08
% spurious	17.52	200.75

Table 1.2 – This table compares the performance of our method versus the benchmark. “%detection” stands for the percentage of correctly identified jumps. “%spurious” stands for the percentage of spurious jumps identified. The simulated data contains bursts of volatility (intensity of volatility jump arrival is 65 per year).

	Network	Lee & Mykland (benchmark)
% detection	88.60	82.40
% spurious	25.96	147.61

Table 1.3 – This table compares the performance of our method versus the benchmark. “%detection” stands for the percentage of correctly identified jumps. “%spurious” stands for the percentage of spurious jumps identified. The simulated data contains bursts of volatility (intensity of volatility jump arrival is 6000 per year).

Furthermore, these results can be visualized in figure 1.1 which grasps the essence of this paper. We plot one time-series of simulated returns (standardized by the bipower variation) that are classified both by our network and the LM test. For readability, circles correspond to returns coming purely from the diffusion component and stars are returns truly containing a jump. We describe the data points by discussing four categories: (i) both methods agree and

[8] $\% \text{ detection} = \frac{\text{\# of correct jump detection}}{\text{total \# of true jumps}}$.

[9] $\% \text{ spurious} = \frac{\text{\# of no jumps, classified as jumps}}{\text{total \# of true jumps}}$.

are correct, (ii) both methods agree and are mistaken, (iii) LM outperforms the network, and (iv) the network outperforms the LM test.

(i) The major part of the returns are small (in absolute value) and come from the diffusion component. Those are correctly classified as “no jump” by both methods (blue circles). Similarly, the isolated large (in absolute value) returns are correctly classified as a jump by both tests (green stars).

(ii) There are a few true jumps that both methods fail to recognize as such (black stars). The reason is that those jumps are small in magnitude comparatively to the Brownian term. In the same spirit, both tests wrongly identify the data points as being a jump (green circles) when the diffusion component realization is unusually high.

The most interesting part of the analysis is where the methods disagree.

(iii) There is no true jump correctly classified only by the LM test. This is in line with our results from tables 1.2 and 1.3 that display a higher detection rate for our neural network. However, our method is not perfect: rarely, it is the only one to mistakenly categorize a point as a jump (yellow circles).

(iv) Since this plot depicts the returns standardized by the BPV, the LM test can be imagined as two horizontal lines symmetrically placed around zero on this plane. Every return between them is classified as no jump, while any outside of this region is identified as a jump. As we see, such approach results in significant amount of misidentification. This happens because the assumption of local changes in volatility (fundamental for the LM test) is violated now. There are two cases where our method outperforms the benchmark. First when a positive jump inside the spot volatility σ_t occurs. In this case, larger diffusion increments coupled with the bipower variation that does not adjust fast enough result in a test statistic that is too high and spurious jump identification by LM (red circles). This category is overwhelmingly big, in line with results presented in tables 1.2 and 1.3. The LM test spuriously identifies as a jump more points than the entire amount of true jumps in the sample. The second scenario is when σ_t experiences a fast drop. The BPV might stay too high, lowering the test statistic and masking the true jump (yellow stars).

To ensure the robustness of our method, we test the algorithm on a different data-generating model. Before, we conducted the out-of-sample analysis by solely changing the parameters. From now on we also modify the structure of the SDEs. The price S_t contains two Brownian terms and a jump component. On top of that, all the randomness in the diffusion coefficients is created by pure jump processes (as suggested by Tauchen and Todorov (2008)). The model is described by the following system of SDEs

$$dS_t = S_t(\mu_t dt + \sigma_{1t} dB_{1t} + \sigma_{2t} dB_{2t} + dJ_{1t}), \tag{1.5}$$

$$d\sigma_{1t} = \alpha_1(\overline{\sigma}_1 - \sigma_{1t})dt + dJ_{2t}, \tag{1.6}$$

$$d\sigma_{2t} = \alpha_2(\overline{\sigma}_2 - \sigma_{2t})dt + dJ_{3t}. \tag{1.7}$$

The results stemming from this specification are displayed in table 1.4. As before our method outperforms the benchmark.

	Network	Lee & Mykland (benchmark)
% detection	86.05	80.04
% spurious	14.96	59.33

Table 1.4 – This table compares the performance of our method versus the benchmark. "%detection" stands for the percentage of correctly identified jumps. "%spurious" stands for the percentage of spurious jumps identified. The simulated data is governed by the model (1.5)-(1.7).

As a final robustness check, we simulate the price by using (1.5) but the paths for σ_{1t} and σ_{2t} are obtained from market data. This is a way to make our simulations closer to reality while being able to assess the performance of the test. For volatility estimation we decide not to use the bipower variation. The reason is that eliminating the impact of jumps requires the time window K to be large enough, over-smoothing volatility. Instead, we first take a time-series of real returns from which we remove the jumps detected by our network. Then the spot volatility is estimated using the realized volatility computed over a shorter window. Applying this procedure to two different stocks gives us two paths that are used as $\left(\sigma_{1t}\right)_{t=0}^{T}$ and $\left(\sigma_{2t}\right)_{t=0}^{T}$. This implies that when performing the 2000 trials, the paths of the diffusion coefficients stay fixed. Also, this technique is subject to errors both in jumps detection and in volatility estimation. However, we believe that this approach provides a meaningful complementary robustness check. The results are presented in table 1.5 and are in line with all previous findings.

	Network	Lee & Mykland (benchmark)
% detection	88.63	81.15
% spurious	3.92	64.43

Table 1.5 – This table compares the performance of our method versus the benchmark. "%detection" stands for the percentage of correctly identified jumps. "%spurious" stands for the percentage of spurious jumps identified. The stock price comes from the process (1.5) but the paths for σ_{1t} and σ_{2t} are estimated from real data.

Confidence intervals can be build empirically for the LSTM network by constructing a histogram of the accuracy on each independent simulation. For each simulated path, we compute the % detection as defined previously for the network and for the Lee & Mykland estimator and we subtract them. The result is displayed on figure 1.2. Most of the mass is over the positive region of the real line meaning that most of the time, our network has a higher percentage of correct detections than Lee & Mykland for the *same* simulated path.

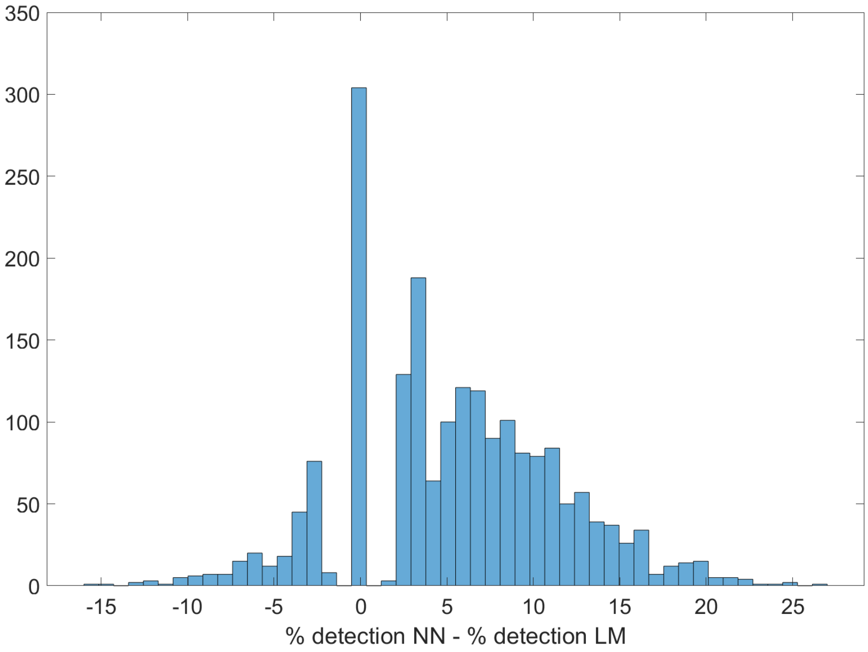


Figure 1.2 – Histogram of the percentage of correct detection by the LSTM network minus the percentage of correct detection by the estimator of Lee & Mykland over the same simulated path, repeated multiple times on independent draws.

We also build a similar histogram but for % spurious. The result is on figure 1.3 and this time most of the mass lays over the negative region. This means that on the *same* path, our network is much less often mistaken.

The estimator of Lee and Mykland (2008) only relies on past and current information when classifying a datapoint. However if the application is simply to analyze historical data there is no reason to impose such restriction. This is why whenever we use an LSTM network we construct it with a bidirectional layer which also accesses future data in order to improve the performance. But in the interest of a fairer comparison, in this paragraph we will discuss the relative performance of the statistical estimator and a unidirectional LSTM. Both of them will therefore rely on the same information set (the past and the present). The results are presented in table 1.6. We observe that even thought the performance of the network decreases if we use a unidirectional layer, it remains superior to the statistical test. Histograms are provided in figures 1.4 and 1.5.

	Bi-Network	Lee & Mykland
% detection	90.94	85.50
% spurious	10.09	114.96
	Uni-Network	Lee & Mykland
% detection	89.94	85.47
% spurious	17.96	113.73

Table 1.6 – Comparison of bidirectional and unidirectional LSTM networks with the statistical test of Lee and Mykland (2008).

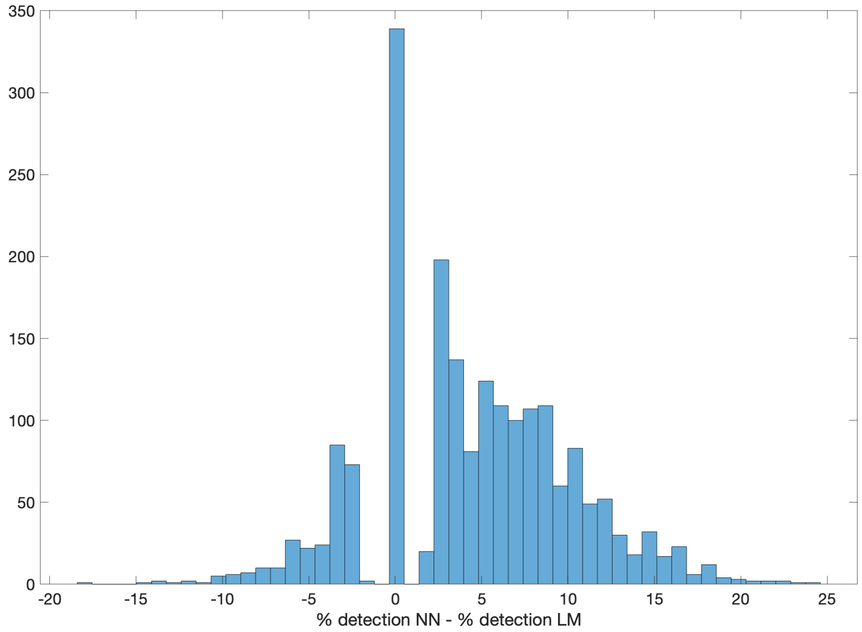


Figure 1.4 – Histogram of the percentage of correct detection by the unidirectional LSTM network minus the percentage of correct detection by the estimator of Lee & Mykland over the same simulated path, repeated multiple times on independent draws.

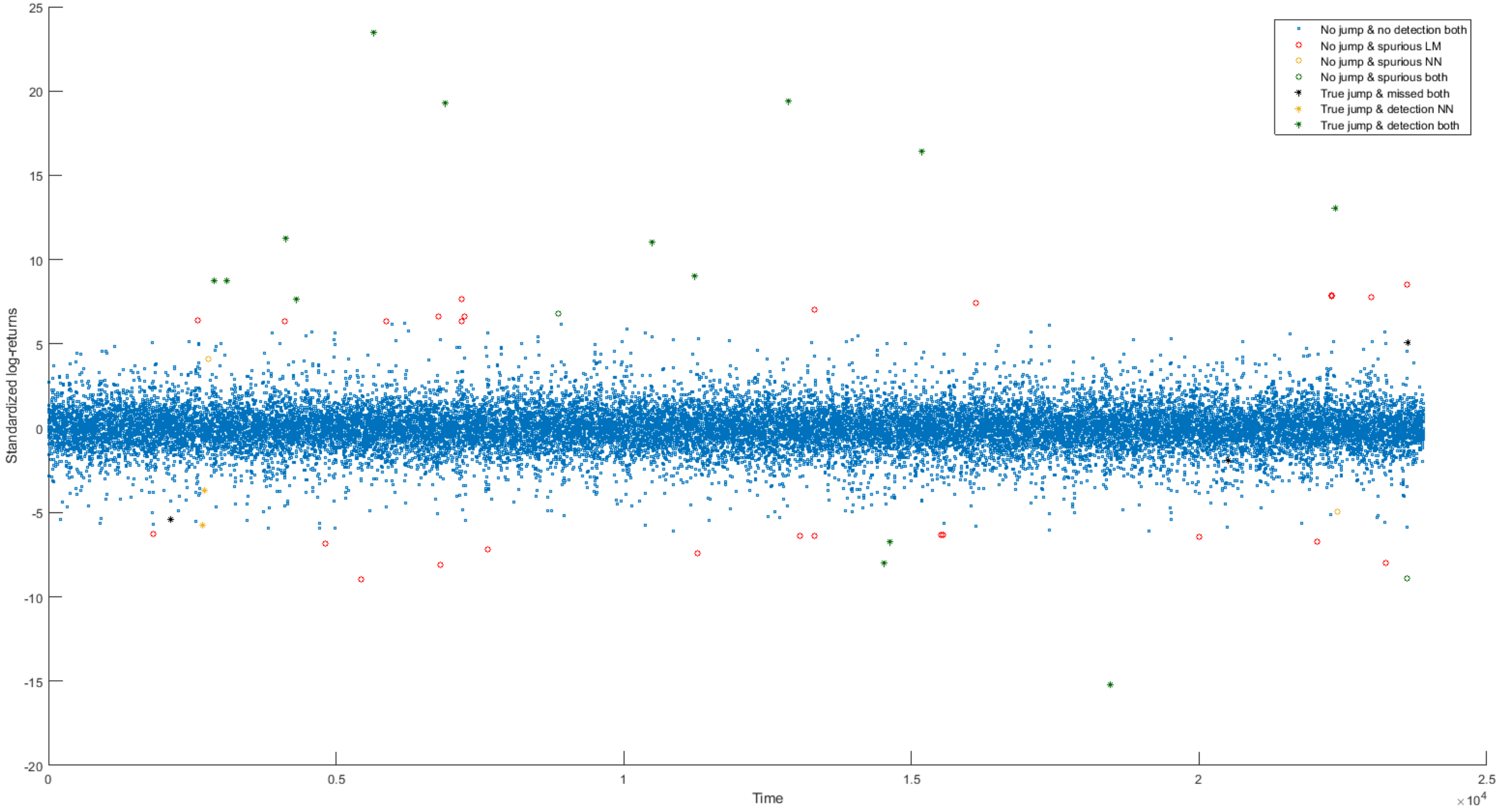


Figure 1.1 – Simulated data. Returns are standardized by the bipower variation and sampled at 2 min frequency for 0.5 year. The spot volatility itself jumps frequently.

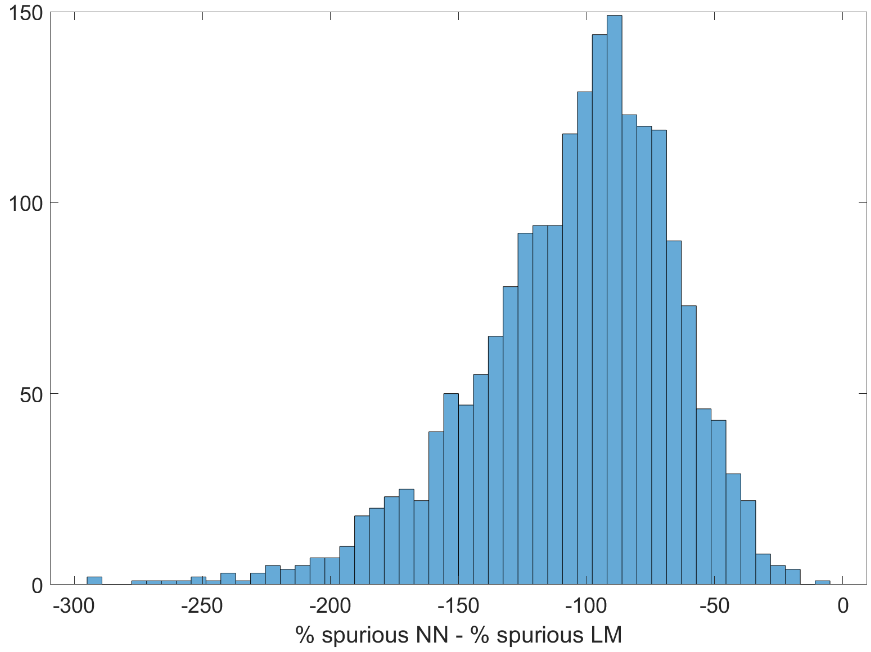


Figure 1.3 – Histogram of the percentage of spurious detection by the LSTM network minus the percentage of spurious detection by the estimator of Lee & Mykland over the same simulated path, repeated multiple times on independent draws.

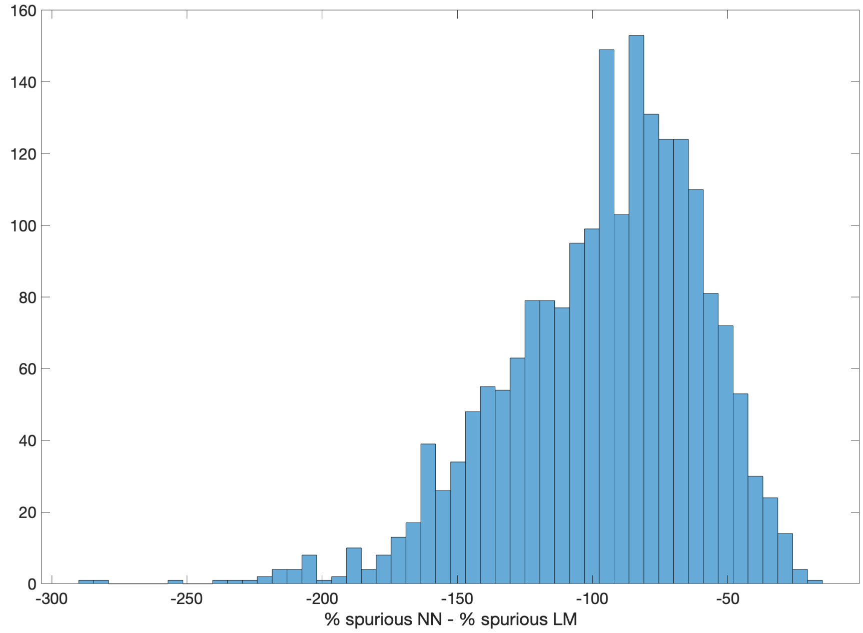


Figure 1.5 – Histogram of the percentage of spurious detection by the unidirectional LSTM network minus the percentage of spurious detection by the estimator of Lee & Mykland over the same simulated path, repeated multiple times on independent draws.

1.6 Application to real data

In this section we perform jump classification on real data sampled at 2 minutes obtained from the Trade and Quote (TAQ) database. Here clearly we cannot exactly assess the performance of our test comparatively to the benchmark. However we provide evidence supporting our method through an event study and volatility forecasting.

As a first step we propose a visual inspection of classified returns of one US company and compare with the LM test. Figure 1.6 displays the returns of American International Group (AIG) standardized by the BPV. Similarly to simulated data, the core of the series that is constituted by a multitude of returns close to zero is classified as coming from the continuous Brownian by both methods (in blue). The isolated large (in absolute value) returns are also classified by both our network and LM as jumps (in green).

In this figure, some jumps identified only by LM (in red) cluster in time. The fact that a jump in the price is followed by other jumps over a very short period of time (usually within few minutes) goes against the very definition of what constitutes a jump in equity data. Finite activity jumps should be rare events and the probability of multiple occurring over a short time span is negligible. For this reason we believe that many of those jumps classified only by LM are in fact coming from the Brownian component during a burst of volatility, explaining their size. Also, these very points (in red) happen to locate close to a group of returns similar in magnitude, but identified as no jump. This suggests that the whole cluster of returns represents a volatility burst. It is unlikely to have a jump-generated return, that has the same magnitude as its diffusion-generated neighbors, so probably it is a misclassification. This finding matches the results of Christensen et al. (2014). The other dimension in which we differ are the jumps classified only by our network while they are considered as continuous returns by LM (in yellow). They happen after a sharp drop of spot volatility, hence the LM test appears to miss those jumps due to the reasons we explained in the previous section.

1.6.1 Event study

In a second step, we analyze the data points where our network and the LM test disagree and convey an event study to support the verdict of one of the two methods. This subsection presents an example.

On September, 19, 2008, the AIG stock experienced a log-return of 12.18% within 2 minutes. This event can be seen on figure 1.6 in the center of the orange circle. In order to have a more granular view of this event, figure 1.7 displays the stock price of AIG for the first half of this day. The price changed from \$2.78 at 10h17am to \$3.14 at 10h19am. Our network classifies this return as a jump, while the LM test disagrees with us and attributes this change to the continuous term. Why do we think that this return was indeed caused by the jump?

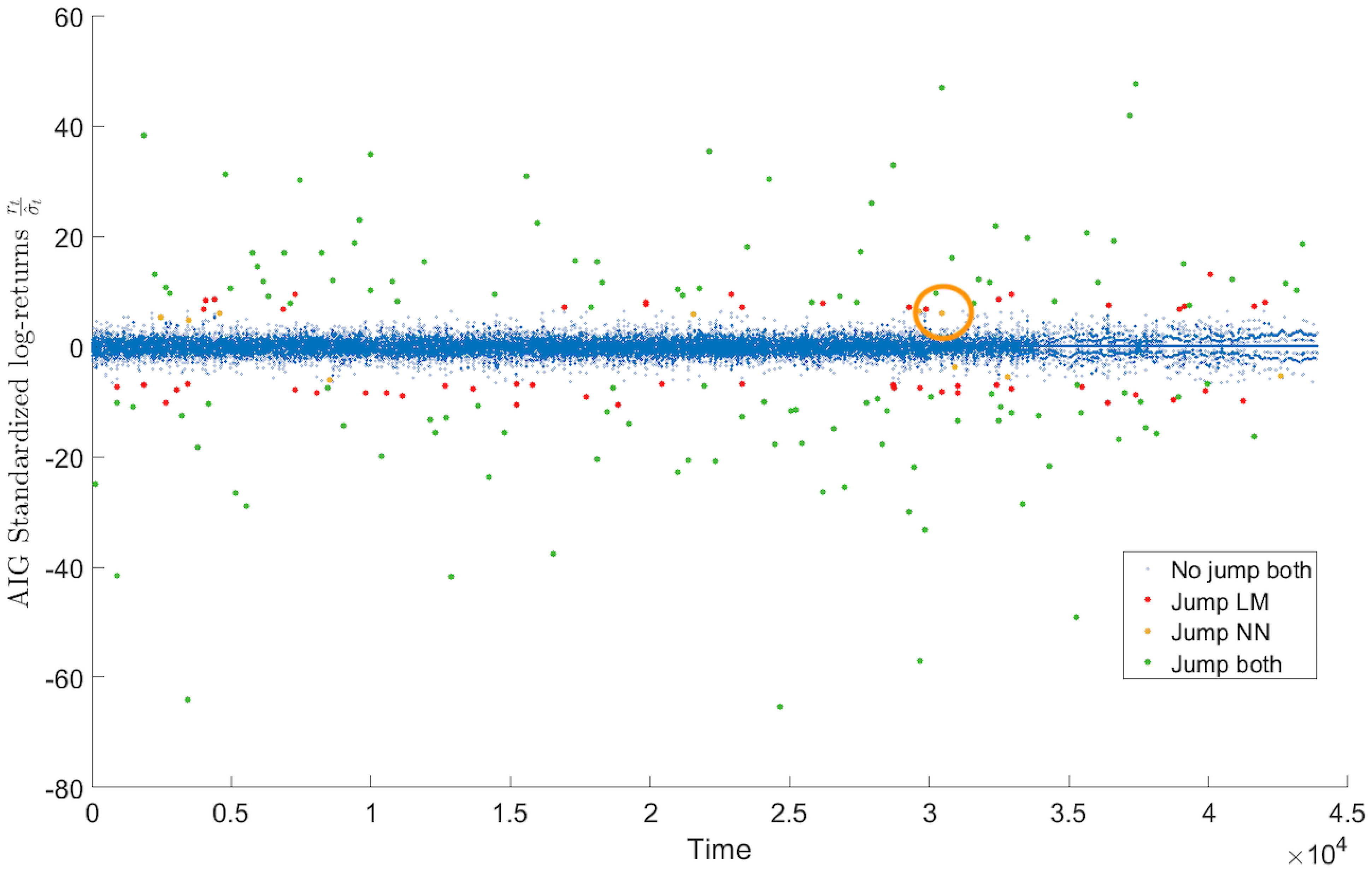


Figure 1.6 – Returns of American International Group (AIG) standardized by the bipower variation at 2 min frequency. Regarding the legend: "No jump both" means that none of the methods detected a jump, "Jump LM" means that a jump was detected only by the test of LM, "Jump NN" means that a jump was detected only by our network, and finally "Jump both" means that both LM and the network agree and detect a jump.

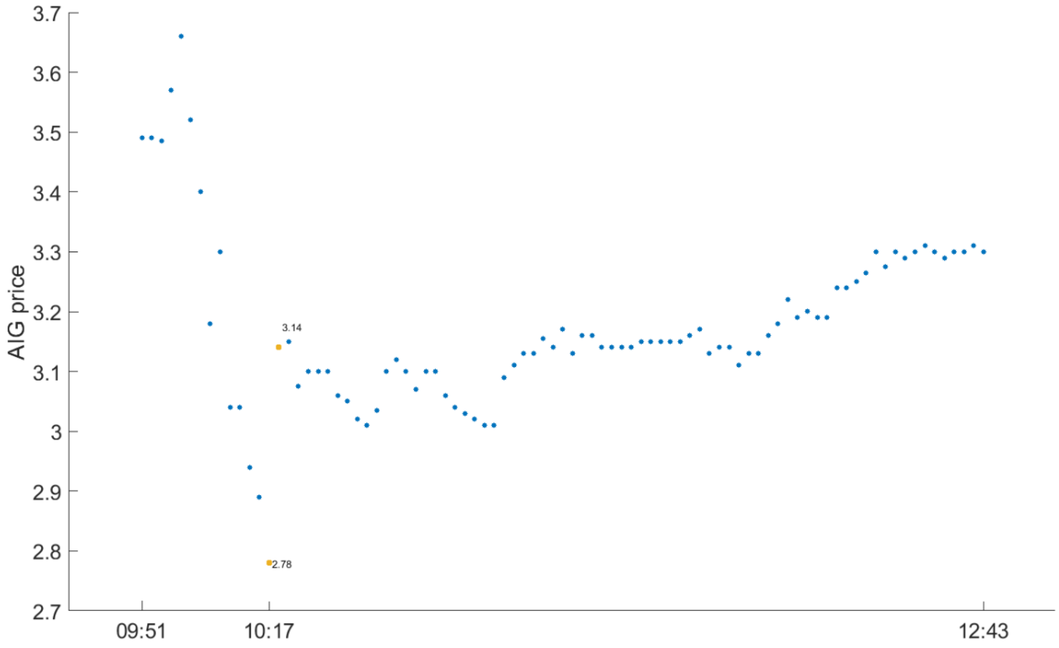


Figure 1.7 – AIG stock price in the morning of September 19, 2008 sampled at a 2 minutes frequency.

The week of September 15, 2008 - September 19, 2008 was in the heart of the financial crisis. On Monday Lehman Brothers filed for bankruptcy, constituting the largest one in US history. That was a period of market instability, featuring high volatility. But on Friday September 19, at 10:05, the Treasury Secretary Henry Paulson announced a number of actions that the state would take to stabilize the financial system. In particular, the money market funds guarantee program was announced. This was perceived as positive news, and pushed markets up. The length on the Secretary's talk was approximately 9 minutes, and it ended at 10h14am. Then, three minutes later, we observe a sudden and big upward change in the price of AIG. Most interestingly, after this big change a calm period begins. Volatility decreases significantly and price balances on the relatively stable level. This refutes the hypothesis of the abovementioned return being generated by the diffusion part and supports the jump nature of this return. This is an example of our previous discussion about the failure of the LM test. In this case we suppose that a sudden drop in volatility violates the LM assumption and thus does not allow their test to detect the actual jump, causing a misclassification.

1.6.2 Volatility forecasting

Finally, we propose an exercise of volatility prediction. The intuition for this exercise is as follows. It is known that the HAR-RV (heterogeneous autoregressive model of realized volatility) of Corsi (2009) does a decent job in forecasting the variance. Andersen et al. (2007a) have developed a HAR-RV-CJ model that disentangles RV into a continuous (C) and jump (J) components and showed that it may improve the prediction. Thus a more accurate jump detection method allows for a better modelling of these two components and in turn would be reflected in an increased performance. This allows us to benchmark our method to others.

The first step is to construct an estimate of the daily, weekly and monthly total volatility using high-frequency data. For this purpose we use the realized variance and denote the daily component by RV_t^d (see appendix B for the formula). The weekly volatility is simply an average of the daily quantities over five days

$$\mathrm{RV}_t^w = \frac{1}{5}\left(\mathrm{RV}_t^d + \mathrm{RV}_{t-1d}^d + ... + \mathrm{RV}_{t-4d}^d\right). \tag{1.8}$$

The monthly volatility RV_t^m is constructed in a similar fashion using twenty two days. The plain vanilla regression (in spirit of Corsi (2009)) to forecast one day/week/month ahead is

$$\mathrm{RV}_{t+f}^f = \alpha_f + \beta_f^d \mathrm{RV}_t^d + \beta_f^w \mathrm{RV}_t^w + \beta_f^m \mathrm{RV}_t^m + \mathrm{error}_{t+f} \tag{$\mathscr{PV}$}$$

where $f \in \{d, w, m\}$.

In order to split RV into a continuous variance C and a jump variance J we compare three methods. The first one simply uses the bipower variation to estimate C and then computes the jump volatility as $J_t = RV_t - C_t$. We call this method $\mathscr{BPV}$. A second approach ($\mathscr{LM}$) is to use the jumps detected by the LM test and remove them from the time-series of stock returns. Assuming the method works perfectly, we obtain a series of returns generated by a pure diffusion process. Then we can compute the realized variance of this newly created series of returns in order to obtain C. A third method ($\mathscr{NN}$) is to perform the same steps but using our neural network instead. Of course the last two methods produce different results since LM and NN disagree about certain jumps.

For each of the three approaches we run the following regression

$$\begin{aligned}\mathrm{RV}_{t+f}^f = \alpha_{f,i} + &\beta_{f,i}^{d,C} \mathrm{C}_{t,i}^d + \beta_{f,i}^{w,C} \mathrm{C}_{t,i}^w + \beta_{f,i}^{m,C} \mathrm{C}_{t,i}^m \\ &\beta_{f,i}^{d,J} \mathrm{J}_{t,i}^d + \beta_{f,i}^{w,J} \mathrm{J}_{t,i}^w + \beta_{f,i}^{m,J} \mathrm{J}_{t,i}^m + \mathrm{error}_{t+f},\end{aligned} \tag{1.9}$$

$i \in \{\mathscr{BPV}, \mathscr{LM}, \mathscr{NN}\}$. In order to prevent over-fitting we set the coefficients in front of the jump variance to zero.[10]

[10] Adding the jump variance terms $J^f, f \in \{d, w, m\}$ improves the in-sample performance but in general deteriorates the out-of-sample results. Jump coefficients being typically insignificant we have decided to remove them.

The comparison of the out-of-sample forecasting performance of the different methods is displayed in table 1.7. We conduct the analysis on the Dow Jones constituents (excluding the financial firms) and the index-tracking ETF (DIA) from 2006 to 2008 included, observed at a 2 minutes frequency. The reported results are the cross-sectional average of the forecasting performance for each method. We can see that at all horizons the neural network reaches the lowest root mean-square error (RMSE) and the highest R^2, beating all the other methods. The differences in performance between the methods might seem small at first glance but we have to remember that jumps are actually rare events and account only for a small portion of the returns. Moreover, the two methods $\mathscr{LM}$ and $\mathscr{NN}$ only disagree on a strict subset of the already small number of jumps. It is therefore normal to obtain performance metrics that are relatively close to one another. Yet, the higher performance of the network hints at the fact that even on real data it is better able to detect jumps and extract a more precise signal of future total volatility.

	Daily		Weekly		Monthly	
	R^2	RMSE	R^2	RMSE	R^2	RMSE
$\mathscr{PV}$	62,41	1,12	69,83	0,88	64,56	0,89
$\mathscr{BPV}$	64,69	1,09	71,75	0,85	65,78	0,88
$\mathscr{LM}$	64,98	1,08	72,42	0,84	65,72	0,88
$\mathscr{NN}$	65,38	1,07	72,53	0,83	65,99	0,87

Table 1.7 – Comparison of the out-of-sample performance for one day ahead daily, one week ahead weekly and one month ahead monthly volatility forecast by four methods on the Dow Jones stocks (financial firms excluded) from 2006 to 2008. All the regressions are reestimated daily. R^2 and root mean square error (RMSE) are used as performance metrics.

1.7 Conclusion

We present a new method that uses an LSTM neural network to detect jumps nonparametrically in high-frequency data. The network is able to distinguish between jumps and bursts of volatility without using ultra high-frequency data, avoiding microstructure noise. The network is trained on labelled data generated at virtually zero cost using Monte-Carlo. Using out-of-sample simulated data, our approach significantly outperforms the benchmark of Lee and Mykland (2008) in realistic market conditions (i.e. in presence of jumps inside the spot volatility of the price process). On real data, we provide supportive evidence of the higher accuracy in jump detection of the neural network through an event study and improved volatility forecasts.

2 Deep Learning for Asset Bubbles Detection (co-authored with Alexis Marchal)

2.1 Introduction

Asset bubbles have preoccupied investors nearly since the beginning of financial markets. From the Tulip mania of 1636-1637 all the way to the more recent financial depression of 2008, including many others in between. Recently, bubble concerns have resurfaced with the rise of crypto assets experiencing rapid increase and violent crashes in prices. The concept of a bubble (defined as a deviation of the price from the fundamental value) might be well known for market participants but its detection remains a challenging task because the fundamental value is not observable.

The fundamental value is the expected sum of future cash-flows or benefits offered by the security, appropriately discounted for time and risk. The difficulty comes from the fact that one always needs some assumptions in order to make a statement about it. One approach is to build a theoretical model for the fundamental value but then the bubble detection procedure would suffer from a joint-hypothesis problem (you are simultaneously testing the presence of a bubble and the assumptions of the model). A second approach is to use almost model-free techniques from mathematical finance under the assumption of a continuous-time economy. Essentially, this boils down to classifying stock prices as being either true or strict local martingales. However to do so it is necessary to estimate the volatility function of a diffusion process given discrete time-series data and this method suffers from estimation errors.

This paper aims at providing a bubble detection methodology that outperforms the existing one under a continuous-time paradigm. To the best of our knowledge, we are the first to provide a certain number of findings to the literature. Our first contribution is to show that long short-term memory (LSTM) networks outperform current methods at detecting asset bubbles both in terms of accuracy and computational time. The second contribution is to provide an evaluation of the method proposed in Jarrow et al. (2011) and Obayashi et al. (2017) in a controlled environment. Indeed, by using simulated data we know exactly when an asset bubble occurs and are able to assess the accuracy of their method (and in a second step compare it to ours). Finally, our last contribution is to assess the economic significance of these bubbles. We essentially detect them on real data and build a long-short trading strategy to assess how much a trader could potentially earn by exploiting this risky arbitrage.

The profitability of the strategy provides support for theoretical foundations that use strict local martingales to detect bubbles. It empirically shows that it is valid to consider the price as a continuous-time process that can only be observed at discrete random times (the trading times), as postulated in Jarrow and Protter (2012).

The rest of the paper is organized as follows. Section 2.2 describes the literature. Section 2.3 lays out the theoretical foundations for characterizing bubbles in continuous-time and presents the existing methods for detecting them. Section 2.4 explains the main idea behind the architecture and the training of the network. Section 2.5 assesses the performance of our method on out-of-sample simulated data. Section 2.6 applies the algorithm to detect bubbles using real data and builds a trading strategy that exploits this risky arbitrage. Section 2.7 discusses the implications of a discrete-time paradigm for our results. Finally section 2.8 concludes.

2.2 Literature review

With the focus being mostly on the martingale theory of bubbles, our paper is largely inspired by the insights of Jarrow et al. (2007) and Jarrow et al. (2010). In these studies the authors provide conditions for bubbles to exist in complete and incomplete markets respectively. Importantly, they show that a non-trivial bubble can be of three types. Two of them can exist only in infinite horizon economies, thus being less relevant for empirical applications. The third type of bubbles, that is the only type viable in the finite horizon framework, is proven to be represented as a strict local martingale under a risk-neutral measure. We are interested in detecting bubbles of that third type in data, and it explains our inclination to work with a continuous-time framework. The interested reader may find more details about the mathematical foundation of bubbles in continuous-time in the work of Protter (2013), while a general review of the asset price bubbles is proposed in Jarrow (2015). Hugonnier (2012) provides a theoretical framework for rational bubbles to appear in equilibrium asset pricing.

For diffusion processes with volatility being a function of the price solely, Delbaen and Shirakawa (2002) and Mijatović and Urusov (2012) provide a theoretical test based on the instantaneous volatility to distinguish between a true and a strict local martingale. The difficulty in applying this test for data is that the functional form of the volatility has to be known. Jarrow et al. (2011) try to solve this problem by proposing one parametric and multiple non-parametric methods for estimating it. One limitation of their paper is that the functional form of the volatility has to stay the same over the whole sample period. The recent work of Obayashi et al. (2017) mitigates this constraint by allowing the functional form to vary among different time periods. This results into adding more flexibility to the method, but deteriorating the quality of each period estimation, leaving space for further improvement. We propose a new direction for this literature, departing from the classical statistical test by using deep learning techniques.

Machine learning methods are gaining a huge popularity in financial research, both among academicians and practitioners. With financial data being mostly represented as time-series, the family of recurrent neural networks is the logical choice for the majority of problems in the field. A prominent member of this class is the long short-term memory network (LSTM) proposed by Hochreiter and Schmidhuber (1997). It is able to remember important information that was faced in the far past and forget unimportant recent one. This type of network is so far the most popular for tasks involving complex historical patterns of data.

2.3 Bubbles in continuous-time

There are two paradigms on modeling asset prices: discrete or continuous-time. Often, both frameworks are equivalent in the sense that they describe the same phenomenon. For instance, for derivative pricing a discrete-time binomial model approximates the Black-Scholes formula which relies on a continuous-time framework. However this equivalence is in general not true.

Whether we adopt a discrete or continuous-time approach greatly changes the theoretical framework and assumptions required for bubbles detection. In a discrete-time economy, bubbles can only exist if the time horizon is infinite. Moreover, the only way to detect them is by relying on a model for the unobserved fundamental value, creating a joint-hypothesis problem. In continuous-time, bubbles can exist even in finite horizon economies and we only need to assume a certain stochastic differential equation (SDE) describing the evolution of the price. We can assess whether the SDE accurately describes the price changes independently from testing for the presence of bubbles (hence eliminating the joint-hypothesis problem). In both cases, we are forced to make a relatively strong assumption: either infinite time horizon or continuous-time. In this paper we have decided to make the latter assumption. This implies that we use objects (strict local martingales) which only exist in continuous-time models but not in their discrete-time counterparts.

It is therefore natural to wonder: how valid is this assumption? And how relevant are the results for the real world? In reality it is true that, despite the fact that high-frequency firms trade with incredibly short intervals, it still happens at discrete times. However, a trade can happen at any point on the half-line $[0,\infty)$ that represents time. At the tick level, the time between trades is not uniformly spaced as pointed out by Jarrow and Protter (2012). This contrasts with a discrete-time model where the time grid is fixed and trading is forced to happen at uniformly spaced dates. This is why it is more plausible to imagine that the price process evolves in continuous-time and we are only able to observe its values at random stopping times (when a trade takes place). Moreover, even though our objects of interest only exist in continuous-time, they can be approximated by discrete-time processes (see appendix C.0.3 for more details). A longer discussion on this topic is available in Protter (2013) in section 11.

Nevertheless, if the reader is a firm believer of the discrete-time framework, we can still give meaning to our results. We show that what we detect are true martingales but with significantly skewed distribution. We discuss in depth the link with a discrete-time economy in section 2.7.

2.3.1 Theoretical framework

We essentially use the set-up described in the annual review of Jarrow (2015) which is also common to most of the papers in this literature. We keep the theoretical framework to a minimum to obtain a concise paper but we still ensure that the reader can understand our main contributions.

Let $(\Omega, \mathscr{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space where the filtration satisfies the usual hypothesis (Protter (2001)). Time is continuous and the economy lives over a compact time interval $[0, T]$ with $T < \infty$. As discussed earlier, we only study bubbles that occur in finite horizon economies. There exist two assets. The risky asset price is denoted by S_t and its real world dynamics is described by

$$dS_t = b(t, S_t)dt + \sigma(t, S_t)dB_t, \quad S_0 = s_0 \tag{2.1}$$

where B_t is a one-dimensional $\mathbb{P}$-Brownian motion. We denote by $\tau \leq T$ the random terminal date[1] of this asset. The second asset is locally riskless and pays the risk-free rate r_t. Its value is given by the solution of the following differential equation

$$dS_{0t} = S_{0t} r_t dt, \quad S_{00} = 1. \tag{2.2}$$

The discounted price of the risky asset is defined in the classical way as

$$\hat{S}_t \triangleq \frac{S_t}{S_{0t}}. \tag{2.3}$$

As mentioned before, an asset is in a bubble when the spot price is trading above the fundamental value. So first we need to clarify what the fundamental value is. We assume that the cumulative dividends process $D = (D_t)_{0 \leq t \leq \tau}$ is an adapted càdlàg non-negative semimartingale. Then the fundamental value $S_t^\star$ is defined in a standard way as the expected sum of discounted cash-flows to be paid. Let $\mathscr{M}$ be a non-empty set of equivalent local martingale measures (ELMM). We fix[2] a measure $\mathbb{Q} \in \mathscr{M}$ and write

[1] This date can be thought as a bankruptcy or when the shareholders decide to liquidate the assets of the firm for instance.

[2] For concerns regarding the incompletness of the market we refer to the discussion in Jarrow (2015).

$$\hat{S}_t^\star \triangleq \mathbb{E}_t^{\mathbb{Q}} \left[\int_t^\tau \frac{dD_s}{S_{0s}} + \hat{L}_\tau \right] \tag{2.4}$$

where $\hat{L}_\tau \geq 0$ is the discounted liquidation value of the asset at the terminal date. We formally define a bubble process as

$$\beta_t = S_t - S_t^\star. \tag{2.5}$$

We say that an asset experiences a bubble whenever $\beta_t > 0$. The next theorem characterizes this phenomenon. For simplicity of exposition, we assume no dividends from now on.

Theorem 2.3.1 (Obayashi et al. (2017)) *A risky asset price process S is undergoing bubble pricing on the compact time interval $[0, T]$ if and only if under the chosen risk neutral measure the discounted bubble process $\hat{\beta}$ is not a martingale but is a strict local martingale. This is equivalent to the discounted price process $\hat{S}$ (plus its cumulative discounted dividends) being a strict local martingale since $\hat{S}^\star$ (plus its cumulative discounted dividends) is always a martingale and not a martingale.*

Theorem 2.3.1 provides us with a simple criterion for detecting bubbles by identifying strict local martingales. Now we can fully understand the implication of our assumptions since we aim at detecting strict local martingales which only exist in continuous-time. A benefit of this setting is that we do not need to model the fundamental value, all we need to know is whether the discounted stock price is a strict local martingale or not.

Under $\mathbb{Q}$ the process $\hat{S}_t$ has the following dynamics

$$d\hat{S}_t = \sigma(t, S_t) S_{0t}^{-1} dB_t^{\mathbb{Q}}. \tag{2.6}$$

By Girsanov theorem, the diffusion coefficient stays the same under any equivalent change of measure, so without loss of generality we can in fact work with any $\mathbb{Q}$. Now that we have the risk-neutral dynamics, we need to know how to differentiate between true and strict local martingales. From now on, we assume the interest rate to be constant and equal to zero for simplicity of exposition. This will not impact our results on simulated data. Regarding the application on real data where rates are not constant, analogous results will hold under some technical conditions. A discussion can be found in Protter (2013).

Thanks to the work of Delbaen and Shirakawa (2002) we have a simple condition to distinguish between true and strict local martingales. For a process of the form

$$dX_t = b(X_t)dB_t^{\mathbb{Q}}, \tag{2.7}$$

we first define an integral

$$I(\varepsilon) \triangleq \int_{\varepsilon}^{\infty} \frac{x}{b(x)^2} dx. \tag{2.8}$$

Then depending on whether this integral converges or not two cases can arise:

- X is a strict local martingale (i.e. experiences a bubble) if and only if $I(\varepsilon) < \infty$ for all $\varepsilon > 0$.
- X is a true martingale (i.e. does not experience a bubble) if and only if $I(\varepsilon) = \infty$ for all $\varepsilon > 0$.

At this point, we need to make assumptions about the diffusion coefficient $\sigma(t,s)$ in the SDE (2.6). The bubble testing procedure requires us to obtain this coefficient solely as a function of s, and not as a function of time. However assuming that the volatility function only depends on the stock price and that it never changes through time is unrealistic. This would imply for instance that if a stock is in a bubble, it will stay in this state forever. We therefore follow Obayashi et al. (2017) and assume a regime change model for the diffusion coefficient such that

$$\sigma(t,x) = \begin{cases} \sigma_1(x), \text{ if } MC_t = 1 \\ \sigma_2(x), \text{ if } MC_t = 2 \\ \dots \\ \sigma_n(x), \text{ if } MC_t = n, \end{cases} \tag{2.9}$$

where MC_t is a Markov chain with n regimes. Note that time intervals do not need to be equally spaced. In general, they are random and unknown. Now over a given time interval $(t_i, t_{i+1}]$ we will be able to estimate the diffusion coefficient as a function of s only and perform the integral test locally using (2.8). In a theoretical model, the computation of $I(\varepsilon)$ is (easily) done. However, applying this test on real data is complicated by the fact that $\sigma(x)$ is not known. Its recovery becomes the cornerstone of the analysis. The tricky point of this estimation is that the functional form may vary over the ex-ante unknown time intervals. Another difficulty is that only data on a bounded interval is available for this estimation, while what matters for $I(\varepsilon)$ is the behavior of $\sigma(x)$ when $x \to \infty$.

For the readers familiar with the estimation of historical volatility, it is important to emphasize that the task is *not* to estimate the volatility through time but to estimate the functional form of volatility $\sigma(x)$ with respect to the price x over some time interval. This implies that volatility estimators like realized variance (RV) are of no use for this task.

As another side note, it is worth noticing that $\beta_t \geq 0$ for any t as shown in Jarrow (2015) in theorem 3 for instance. This means that in this framework, the stock price might be above the fundamental value but never below. It implies that when later we build a trading strategy using real data, we will always short stocks that are classified as being in a bubble.

2.3.2 Classical statistical method for bubble detection

In this section, our goal is to motivate the necessity of a neural network to detect bubbles by showing that it significantly improves the detection rate. We do so by constructing a simple example on simulated data where the stock volatility is assumed to only have two regimes. We show that already here the method described in the literature often underperforms. This example is unrealistically simple but sufficient to prove our point. Later we will use a more complex structure for the data-generating process when making the final comparison of both methods.

We now discuss the method presented in Obayashi et al. (2017). The authors decide to estimate $\sigma(x)$ over a rolling window of 21 days and propose two estimators. The first one being a parametric estimator and a second non-parametric one based on the local time of a Brownian motion. In that case, they can analyze when a bubble is born and when it bursts (and ultimately construct a distribution of the lifetime of a bubble).

In our paper we will only present and benchmark our network to the parametric estimator (PE). Given the fact that we will assess the performance of the two methods on simulated data, we always know the true data-generating process. We use the same family of functions for data simulations and for the parametric estimator. Since the PE uses a function from the true family, its performance will be higher than a non-parametric estimator. The latter might lead to better performance on real data where the functional family is unknown. However, with real data we cannot compare the methods because we do not know the true volatility regime.

In order to simulate data, we first assume a functional form for $\sigma(x)$. We choose the well-known power family[3]

$$\sigma(x) = \gamma_0 x^{\gamma_1} \tag{2.10}$$

where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$ is constant over a given time interval $(t_i, t_{i+1}]$ but is allowed to change from

[3] This functional form implies that we model the stock price with a constant elasticity of variance (CEV) model.

one time interval to another. Given this function, the stock price S_t is a strict local martingale for $\gamma_1 > 1$. In this simplified example we simulate price paths by using only two regimes:

R1 When the stock has $\gamma_1 = 0.9$.

R2 When the stock is in a bubble (i.e is a strict local martingale) and we set $\gamma_1 = 1.1$.

A Markov chain decides on the random times when the stock transits back and forth between regimes **R1** and **R2**.

It is important to emphasize that in a model like equation (2.9), even if today the stock is in a state with $\gamma_1 \leq 1$, we cannot conclude that there is no bubble. Because the existence of a positive probability to transition to a state with $\gamma_1 > 1$ would imply that there is a bubble today. The only thing that our methodology is able to achieve is to detect bubbles whenever we are in a regime with $\gamma_1 > 1$.

It is clear that the scaling parameter γ_0 does not impact the convergence of the integral in (2.8) so we set $\gamma_0 = 0.15$ for both regimes for simplicity. Once the data is simulated, we need to estimate $\boldsymbol{\gamma}$.

The above-mentioned paper relies on a parametric estimator originally introduced in Genon-Catalot and Jacod (1993) which is given by

$$\hat{\boldsymbol{\gamma}}^{\mathrm{PE}} = \operatorname{argmin} \frac{1}{n} \sum_{i=1}^{n} \left(\gamma_0^2 S_{t_{i-1}}^{2\gamma_1} - \frac{n}{T} \left(S_{t_i} - S_{t_{i-1}} \right)^2 \right)^2 . \tag{2.11}$$

Since the data is simulated we know exactly when the stock was in a bubble and we compare this information to the one given by the estimators. To do so, we construct the following random variable

$$\xi_t \triangleq \mathbb{1}_{\{1/2 \leq \gamma_{1,t} \leq 1\}} \tag{2.12}$$

which takes the value one when the stock is in a regime that we do not identify as bubble and zero otherwise. Then we denote by $\hat{\xi}_t^{\mathrm{PE}}$ and $\hat{\xi}_t^{\mathrm{NN}}$ the parametric estimator[4] (PE) and the neural network estimator (NN) respectively.

The results for a single price path are displayed in figure 2.1 which contains four subplots. The first one simply shows the log-returns of the price through time during the different market

[4] In order to obtain $\hat{\xi}_t^{\mathrm{PE}}$ we follow the methodology described in Obayashi et al. (2017). More precisely, we first use the estimator described in equation (2.11) and then apply a Hidden Markov Model to smooth the signal and use the probability matrices from their paper.

regimes. The second subplot displays ξ which takes the value 0 when the stock experiences a bubble and otherwise is equal to 1. Here we see that over the whole period, the stock experienced 3 bubbly regimes and 3 “normal” regimes. Then the last two subplots display the estimation of ξ according to the parametric estimator available in the literature and the neural network respectively. A direct visual inspection reveals that the neural network largely outperforms at detecting bubbles (i.e $\hat{\xi}_t^{\mathrm{NN}}$ is a better estimator of ξ).

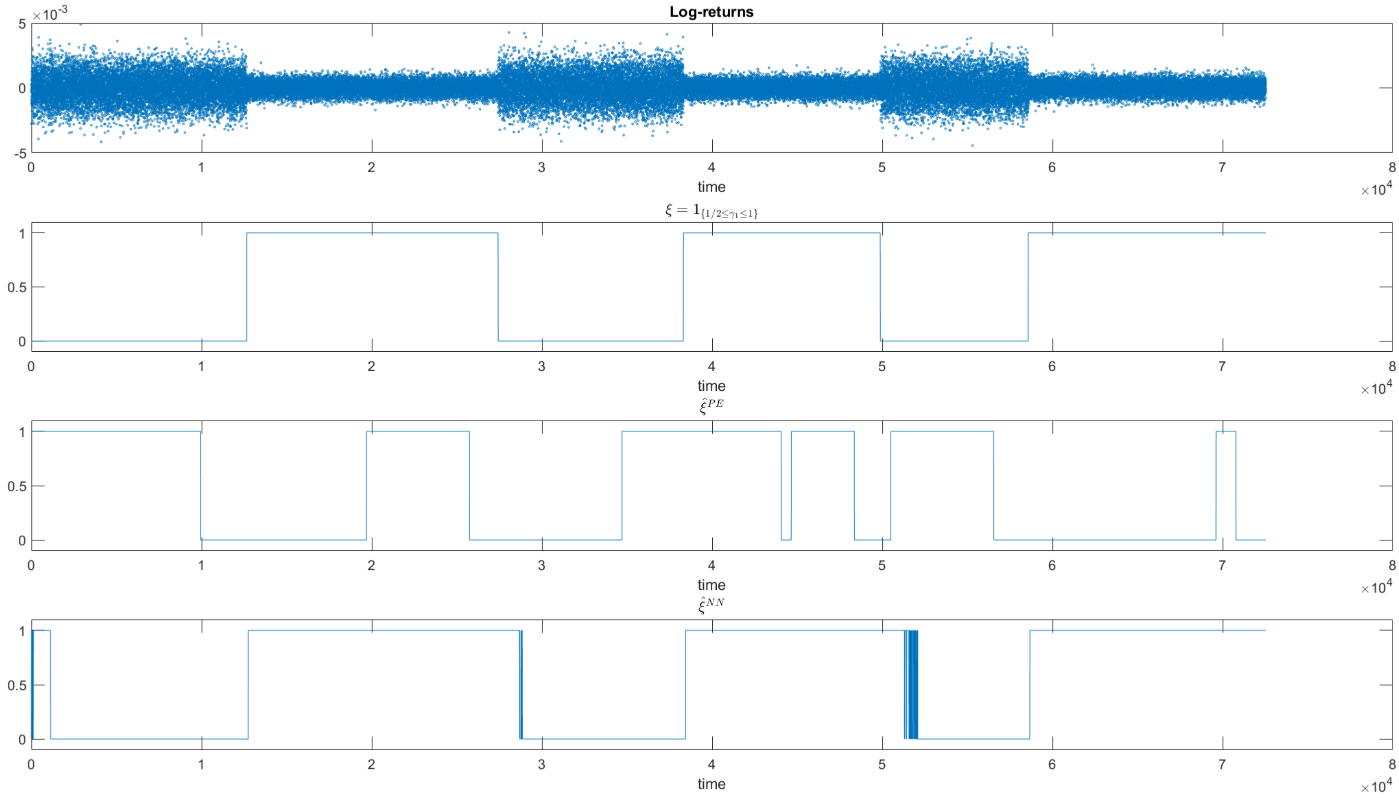


Figure 2.1 – Simulated data showing the drawback of the parametric estimator (PE)

Why is it the case? If S_t stays in one specific regime (either true martingale or strict local martingale) over the whole sample path, then the statistical estimator performs quite well. However the statistical method starts to be challenged as soon as it has to perform the estimation of $\sigma(x)$ during a period when the regime changed. The reason is very intuitive. When using a rolling window around a regime change, the estimator will be using data points generated during the old regime as well as data points belonging to the new regime. This implies that it will fail to correctly estimate (γ_0, γ_1) around these times. Choosing a large window will lead to more stable coefficients however it will fail to identify the exact time of a regime change (or even miss some short lived regime changes). A small window leads to unstable estimations. Obayashi et al. (2017) mitigate this problem by combining their estimator with a Hidden Markov Model used to smooth the bubble signal. Nevertheless, as we see here it still underperforms a neural network.

How does the network manage to overcome this problem? Intuitively, it is first able to detect the times of regime changes before estimating the parameters (γ_0, γ_1). By doing so, it does not use data from another regime. In a given identified regime, it includes all the data points from this regime for the estimation, using potentially more appropriate data than with a rolling window and thus improving the estimation.

This simple example is easy to understand but is far from reality. For instance the time-series of returns in figure 2.1 does not resemble real returns. This is why a thorough comparison of both methods in realistic conditions will be done in section 2.5 after introducing the neural network and the training process. However already in this toy example the statistical method struggles. In a more sophisticated environment its performance will deteriorate even further.

2.4 Network architecture and training

The long short-term memory network we choose for bubble classification is a special type of recurrent neural network that was introduced in Hochreiter and Schmidhuber (1997). By construction, LSTM networks are well suited for remembering long-term dependencies and thus for handling time-series data. We use a standard LSTM, motivated by the main finding of Greff et al. (2016). They present the largest comparison study for different modifications of LSTM networks and show that improvements over the plain vanilla LSTM are marginal. Our network consists of six layers: an input layer, followed by two bidirectional LSTM layers with 100 hidden neurons each, a fully connected layer, a softmax layer, and a classification output layer. An interested reader may refer to appendix A to find a brief intuition about neural networks in general and LSTM in particular, as well as a detailed layers description. Here we train the network with around 50 data points per parameter to prevent overfitting.

Now that the structure of the network is chosen, it has to be trained to detect strict local martingales. Our methodology is conceptually simple. We first generate training data and since it is simulated, we know exactly when the stock was a true martingale (TM) or a strict local martingale (SLM). Then we label each regime as “TM” or “SLM” and train the network on them. Finally the network is ready to classify new time-series.

To create the training data, we simulate paths of the price process (2.6) by discretizing the stochastic differential equation governing it. The training data set as a whole can be composed by multiple time-series possibly generated from different processes by assuming different functional forms for $\sigma(x)$. Using a variety of them allows the network to concentrate only on the specific feature (TM or SLM) of the data point, preventing over-fitting. Another benefit of this method is that we can generate virtually an unlimited quantity of labelled data to train on, enhancing the network performance at no cost.

In order to incorporate stylized facts of equity data, we have decided to model $\sigma(t,x)$ from (2.9) with a Markov chain. Locally, every $\sigma_i(x)$ is modeled using the power function (2.10). To each state of the chain corresponds a different value for $\boldsymbol{\gamma}$. For half of the states we impose $\gamma_1 \leq 1$ while for the other half we set $\gamma_1 > 1$. The Markov chain will control when to switch from $\sigma_i(x)$ to $\sigma_{i+1}(x)$. The transition matrix is made time-varying to prevent the network from learning the probabilities instead of estimating $\sigma(x)$.

Neural networks are considered by some people as a black box. One of our aims is to make the usage of a network as transparent as possible. In section 2.6 we build a trading strategy and try to predict future price changes. However we do not simply feed past price data to a network and ask for a prediction. By doing so, we would not know what factors does the network use in order to make predictions. This approach would be totally model-free but would also be a sort of black box. Instead, we impose some structure on what the network learns by combining it with a model for the stock price. First we assume that the price is governed by a stochastic differential equation. We impose that the network recognizes certain functional forms for the diffusion coefficient (see eq. (2.10) for instance) and we link these functions to the mathematical theory of asset bubbles. Given the prediction of the theory, we can build a trading strategy.

2.5 Performance on simulated data

In this section we assess the performance of the neural network using out-of-sample simulated data. We start by generating new time-series, using parameters[5] that the network was never trained on. Then we classify the data by using the parametric estimator (2.11) and the neural network. Both methods were tested using 150 Monte-Carlo paths, each individual path consisting of data generated at a 2 minutes frequency for 3 years. The results are displayed in table 2.1. The row "% detection" refers to the percentage of data points that were correctly classified. Those include classifying the generating process as being a TM when it was a TM and classifying the process as being an SLM when it was an SLM. "% spurious" refers to the percentage of points when a method was mistaken. Those include classifying the process as being a TM when in reality it was an SLM and classifying the process as being an SLM when in fact it was a TM.

	Network	Obayashi et al. (2017)
% detection	83.64	49.71
% spurious	16.36	50.29

Table 2.1 – Comparison of the accuracy of the neural network and the statistical estimator from the literature for bubbles detection.

The results show that the neural network outperforms the existing statistical estimator by

[5] The new parameters are the values for (γ_0, γ_1) as well as transitions probabilities for the Markov chain governing the volatility regime changes.

displaying a higher rate of correct and a lower rate of spurious detections. We have tried multiple different network architectures and tuned the hyperparameters so that to maximize the performance on out-of-sample simulated data. Gu et al. (2019) find that more shallow networks outperform deeper ones at forecasting financial time-series due to the low signal-to-noise ratio. We obtain a similar result although we use a network for a classification problem.

At this point, it is worth emphasizing that once the network is trained, it is much faster at classifying data. The parametric estimation (since it relies on an optimization) took approximately 43 hours to accomplish the required task while the network did it in less than one hour.

Just like in the previous chapter, we assess the empirical performance of the LSTM network by constructing a histogram of the accuracy on each independent simulation. For each simulated path, we compute the % detection as defined previously for the network and for the parametric estimator and we subtract them. The result is displayed on figure 2.2. Most of the mass is over the positive region of the real line meaning that most of the time, our network has a higher percentage of correct detections than parametric estimator for the *same* simulated path.

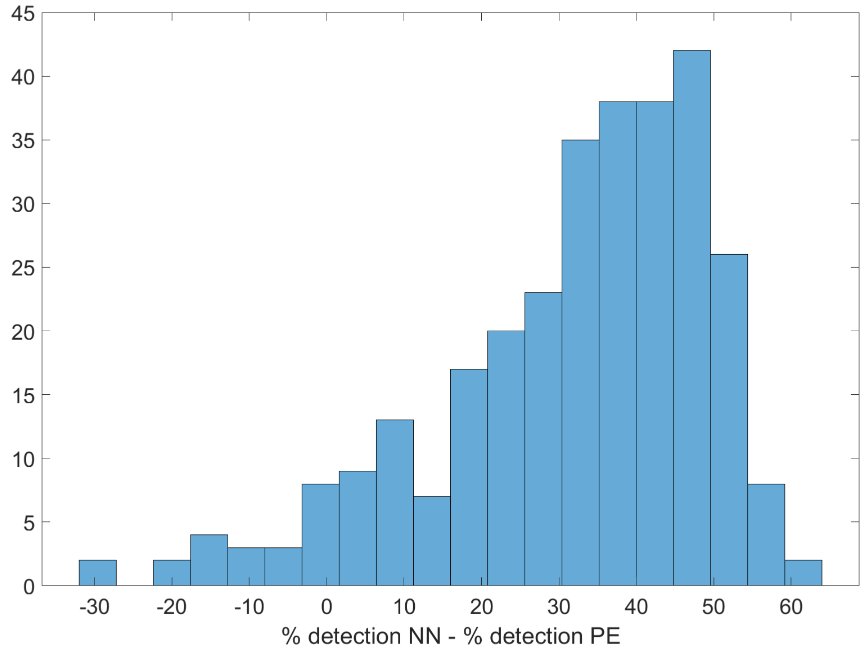


Figure 2.2 – Histogram of the percentage of correct detection by the LSTM network minus the percentage of correct detection by the parametric estimator over the same simulated path, repeated multiple times on independent draws.

We also build a similar histogram but for % spurious. The result is on figure 2.3 and this time most of the mass lays over the negative region. This means that on the *same* path, our network is much less often mistaken.

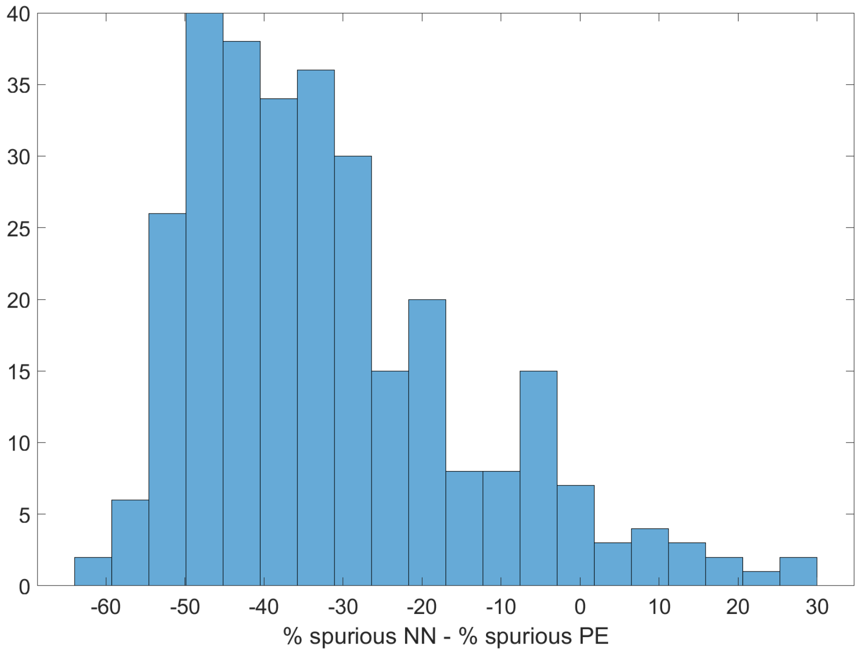


Figure 2.3 – Histogram of the percentage of spurious detection by the LSTM network minus the percentage of spurious detection by the parametric estimator over the same simulated path, repeated multiple times on independent draws.

In this chapter, using a unidirectional or bidirectional network will not significantly change the results of table 2.1. This is because the points living around regime changes constitute a small subset of all the data. This can also be seen by looking at the histograms in figures 2.4 and 2.5 that repeat the above procedure but with a unidirectional LSTM instead of bidirectional.

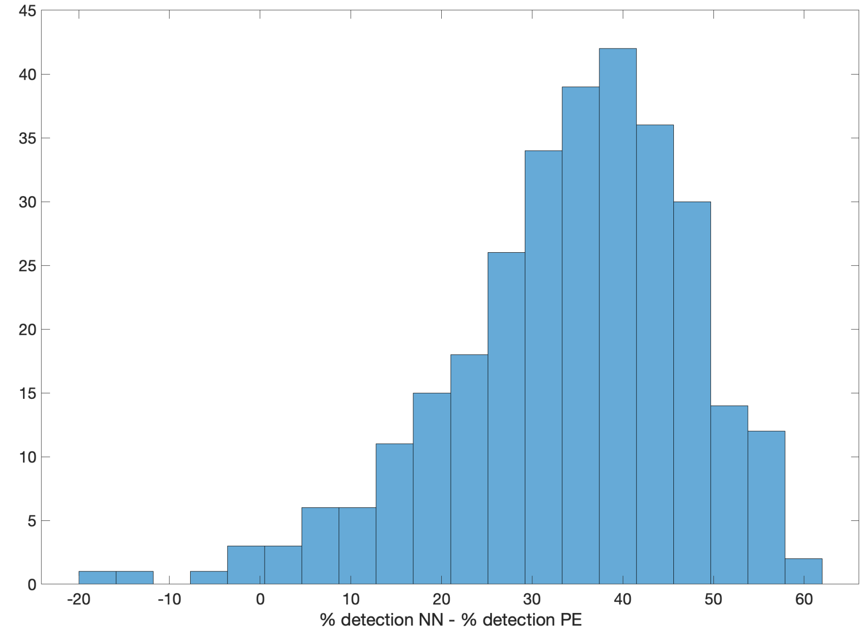


Figure 2.4 – Histogram of the percentage of correct detection by the unidirectional LSTM network minus the percentage of correct detection by the parametric estimator over the same simulated path, repeated multiple times on independent draws.

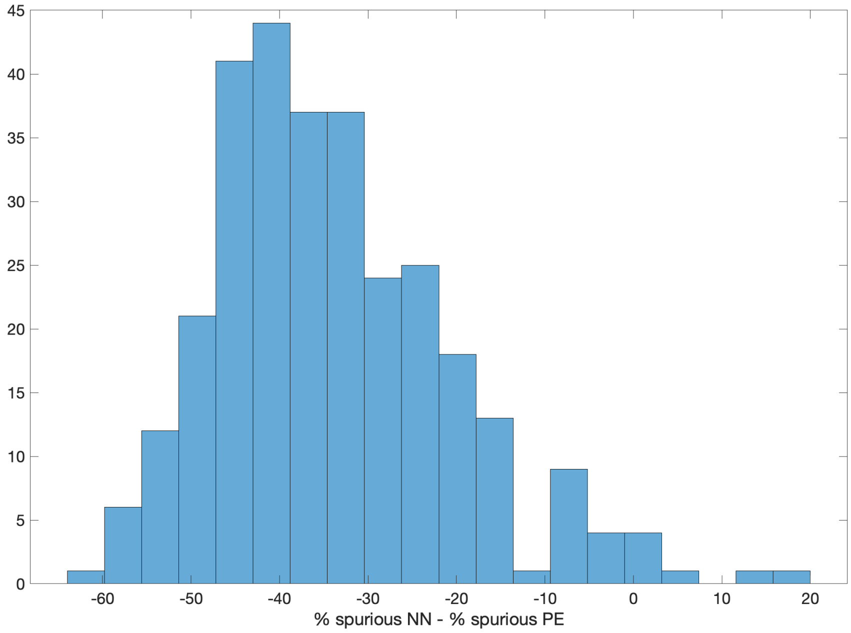


Figure 2.5 – Histogram of the percentage of spurious detection by the unidirectional LSTM network minus the percentage of spurious detection by the parametric estimator over the same simulated path, repeated multiple times on independent draws.

2.6 Application to real data

In this section we perform bubble classification on real data obtained from the Trade and Quote (TAQ) database using our neural network. The dataset consists of 30 individual stocks included in the Dow Jones index (plus the index itself) over a 3 years period (from 2006 to 2008). We select a sampling frequency of 2 minutes. This is high enough to have sufficiently many data points to precisely estimate the volatility function but yet low enough so that the data is not contaminated with microstructure noise. Here clearly we cannot exactly assess the performance of our test comparatively to the benchmark since the true data-generating process is unknown.

It is important to emphasize that up to now we only considered the dynamics of the discounted stock price $\hat{S}_t$ under the risk-neutral measure. By definition of the measure, this process has no drift. We did that because to determine if a stock is in a bubble all we need is the functional form of the diffusion coefficient in order to compute the integral (2.8). The theory tells us that when the stock price is in a bubble, its discounted value is a strict local martingale under $\mathbb{Q}$. Because it is bounded from below by 0, the discounted price is a $\mathbb{Q}$-supermartingale. This means that $\hat{S}_t$ is expected to decrease (under $\mathbb{Q}$). However, when trading stocks in the physical world, the processes describing their evolution likely have a non-zero drift under $\mathbb{P}$ (see equation (2.1) for the real world dynamics of the stock). Therefore to profit from the existence of bubbles, it is necessary to go long and short in different assets in order to cancel any drift effect.

This leads us to a natural way to test our detection method and assess the economic magnitude of asset bubbles in the equity market. For that we build a risky arbitrage trading strategy that shorts an individual stock when in a bubble and at the same time goes long in an asset that replicates its fundamental value. Therefore the short-leg at any time t of our strategy consists of all the individual stocks that are experiencing a bubble at t. The long-leg is the Dow Jones index (which is a proxy for the fundamental value of the short-leg). We invest the same dollar amount in the long and short legs so that we obtain a zero investment strategy. Given that our data covers the beginning of the financial crisis of 2008, the long-short strategy allows us to shield our portfolio from any market-wide shock (we have a zero net directional exposure).

Some classifications of time-series are are displayed in figure D.1 in the appendix. The performance of the trading strategy is presented in figure 2.6 where the vertical axis depicts the dollar value of the portfolio.

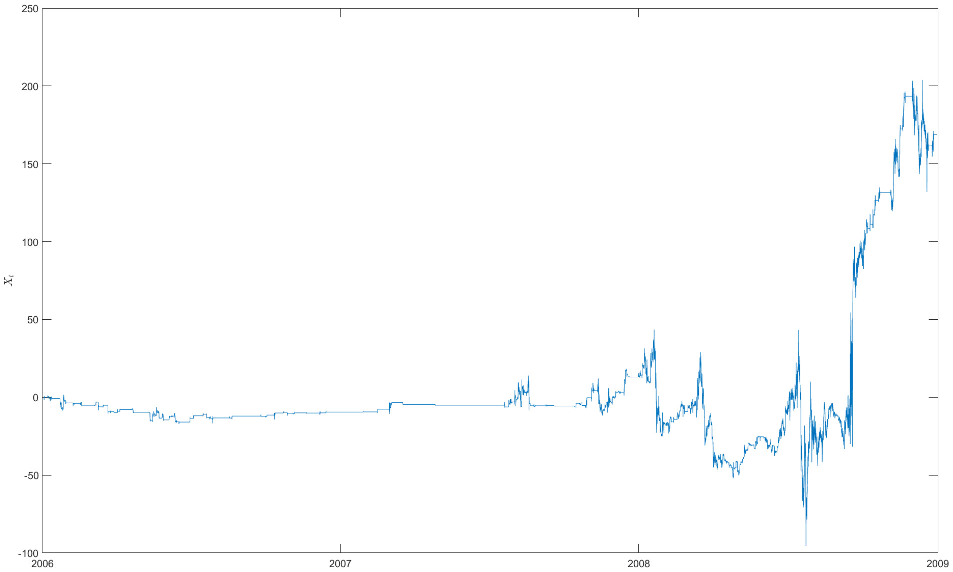


Figure 2.6 – Portfolio value of our long-short trading strategy with zero net exposure.

It is worth emphasizing that our trading strategy is not simply to short high volatility stocks. Suppose that a stock (call it A) has a high volatility but is a true martingale (if γ_0 is large but $\gamma_1 \leq 1$). It is possible that a second stock (call it B) has a lower volatility than stock A but is a strict local martingale (if γ_0 is low but $\gamma_1 > 1$). Our strategy would short stock B but not A.

2.7 Discrete-time paradigm

As discussed in section 2.3, strict local martingales are a continuous-time phenomenon and do not exist in discrete-time. Thus every method (not only our neural network) detecting bubbles by relying on the theory of SLM is subject to the assumption of time continuity. We have discussed previously why we consider such assumption as not only appropriate but in fact preferred over the discrete-time paradigm. However, if the reader is a strong believer in the latter, in this section we explain what would we detect if the true price process indeed evolved in discrete-time.

In this case we assume that the stock price follows the equivalent of our continuous-time process. For this purpose, let us first time-discretize the stock price (2.6)[6] as

$$S_{t+\Delta t} = S_t + \gamma_0 S_t^{\gamma_1} \sqrt{\Delta t}\, Z \tag{2.13}$$

where the process innovations $Z \sim \mathrm{N}(0,1)$ are iid under $\mathbb{Q}$. In discrete-time, S_t is trivially a $\mathbb{Q}$-martingale for any $\boldsymbol{\gamma}$. However for $\gamma_1 > 1$ the mass of its transition density shifts to the left making it positively skewed. This implies that when $\gamma_1 > 1$, the stock price will often decrease (by a small amount) and rarely increase (but by a large amount), so that the conditional expectation is kept equal to the current value of S_t.

In this discrete-time setting since the time horizon of the economy is finite, bubbles cannot exist. Therefore, instead of detecting asset bubbles, our network will detect periods when the stock price frequently decreases by a small amount and rarely increases by a large amount. However, since $S_t = S_t^\star$ at all times, the long-short trading strategy previously explained would not generate any profits. The fact that the strategy is profitable (as seen in figure 2.6) provides additional support for the continuous-time paradigm.

2.8 Conclusion

Relying on the martingale theory of asset bubbles, we show that a long short-term memory (LSTM) network is able to detect them and outperforms the current statistical estimator. On a technical level, the network performs the detection by being able to determine if a given time-series has more likely been generated by a strict local or a true martingale. The algorithm does so by estimating the functional form of the diffusion coefficient of a stochastic differential equation and identifying market regime changes. We then deploy our methodology to US equity data and show that there were multiple bubbles between 2006 and 2008. Finally, using this information we construct a zero net exposure trading strategy that shorts assets experiencing a bubble to assess the economic significance of this phenomenon.

[6] Recall that we assumed a risk-free rate constant and equal to zero for simplicity.

Time continuity is a necessary assumption for the type of bubbles we detect. However this hypothesis is a source of debate among researchers. The profitability of our strategy is therefore an indirect support for the validity of using continuous-time processes to model asset prices.

3 Bitcoin Price Factors: Natural Language Processing Approach

3.1 Introduction

The recent blossom of natural language processing (NLP) algorithms is transforming the perception of linguistic data. The mainstream finance research departs from considering it as an alternative and exotic type of data and starts to widely rely on this information source. As an example, the Annual Review of Financial Economics dedicates one of the issues of 2020 to the textual analysis in finance (Loughran and McDonald, 2020). In particular, a significant share of research concentrates on the ability of textual narrative to explain the state of the economy or specific market (Bybee et al., 2019). However, to be able to retrieve a meaningful relation, proper care should be taken to exclude texts which simply describe the realized moves of the variable of interest itself. For instance, some news articles summarize past market performance, without providing any new information. So far, the most efficient way of filtering such articles out is simply ignoring the big chunks of news coming during the trading hours (Glasserman et al., 2019). Of course, this approach has its drawbacks: for one, a big part of non-endogenous news is missed in this case. Moreover, for the assets traded with the markets open 24/7 it is simply not feasible.

Cryptocurrency is one of the most prominent examples of such never-closing markets. Last year this asset class was in the stage light and experienced a radical shift of concept. Digital currencies departed from being a niche product for technology enthusiasts. Their wider adoption happened simultaneously in the two following directions. Acceptance of cryptos as a mean of payment - not only by tech-oriented flagmen like Microsoft and PayPal but also by daily-oriented businesses like Starbucks, - and further endorsement as a store of value, with the first Bitcoin ETF launched last year in the US, turned them into a widely discussed and closely watched financial instrument.

In this study I concentrate on Bitcoin, being both the most famous cryptocurrency and the one with the highest market capitalization. Bitcoin dominates the total of roughly 1.8$ Tn cryptocurrency market, with more than 42% of total capitalization coming from this coin.[1] While Bitcoin capitalization is comparable with the world's most expensive companies, multiple research challenges arise from the specific properties of this instrument, which is not related to any company or any country. In particular, the unclear intuition behind the underlying fundamental value creates a polarized perception of crypto - from considering it as a purely "hype-based" asset with a fundamental value of 0 (Cheah and Fry, 2015) all the way to a safe-haven asset. Given this dichotomy, explaining Bitcoin wild price movements with a sparse fundamental-based factor model is an even more challenging task than for equities.

In the rational investor framework, new information plays a crucial role in securities pricing. An asset gives its holder the claim on the associated stream of the future unknown cash flows, and the investor evaluates the asset based on her beliefs about these payoffs. Dissemination of the new information, especially about the asset fundamentals, leads to the updates of beliefs, which induces trading activity and thus impacts the price. While for other assets new fundamental information can come through various related channels (as, for instance, through the company reports for equities and corporate bonds), for cryptocurrencies the situation is quite different, as no entity is linked directly to it and it is not backed by any tangible asset.

In this paper, I propose to use the news as a proxy for the fundamental information driving Bitcoin price. I use Sentence-BERT (SBERT) network to obtain a comprehensible and easy-to-understand classification of the articles into a set of predefined topics of interest. My first contribution is an efficient separation of the group of endogenous news (which simply describes the past price movements) from the fundamental news. For example, a news article reporting a recent cryptocurrency exchange hack is likely to influence investors' beliefs about the future of Bitcoin, and thus I consider this news to be fundamental. At the same time, a news article reporting the performance of cryptocurrencies during the previous week is an example of endogenous (descriptive) news. Unlike the currently adopted benchmark, my solution alleviates the endogeneity concern by relying purely on the news content, instead of the time published. This allows to, first of all, work with news for never-closing Bitcoin markets, as well as to keep non-endogenous news arriving during the trading hours. Of course, this approach is not limited to cryptocurrency analysis. It opens wide possibilities for mitigating the endogeneity concern and producing meaningful economic insights when working with narrative data in any economic area. Second, I conclude that the explainability of Bitcoin price with news is at least comparable with the explainability of the traditional financial assets. I reach almost 38% explainability considering the full corpus of news. Moreover, taking care of endogeneity, I am still able to explain more than 16% of monthly Bitcoin return variation with fundamental news. Roll's puzzle (Roll, 1988) formulates the infamous property of the classical, centuries-old equity markets: asset price variation is burdensome to explain (even ex-post) with anything else than other prices. In this light, my result, obtained for the young and wilder market without using any price time series, is quite satisfying. I show that fundamental news has a causal impact on future returns, thus confirming the fundamental information role in

[1] As of March 13, 2022.

cryptocurrency price formation. At the same time, I show that Bitcoin investors mildly diverge in their opinions on the new fundamental information.

I hypothesize that fundamental news can be classified into 4 topics, covering the main aspects of cryptocurrency: Technology, Regulation, Adoption, and Macroeconomic situation. I compare the contribution of each topic to the return explainability, concluding that wider adoption of cryptocurrencies and blockchain technology is the most important topic. Finally, I make an attempt to understand why exactly Bitcoin investors value this cryptocurrency. Accepting the stand that Bitcoin is money leads to the inference that it should fulfill the functions of classical fiat money: store of value and medium of exchange. I assume that Bitcoin investors value this coin for being able to perform these two functions and study which one is dominating. Specifically, I extract two time series, each corresponding to media sentiment about one of these properties. I show that the store of value provides a better explanation for Bitcoin returns than the medium of exchange.

The remainder of the paper is organized as follows. Section 3.2 provides a brief literature review, while section 3.3 discusses methodology and data. Section 3.4 presents the results and section 3.5 offers a robustness check. Section 3.6 concludes.

3.2 Literature review

I build my paper on two strands of economic literature. On the one hand, I find inspiration in papers aiming at explaining financial time series with text data. Frazier et al. (1984) trailblazing work stresses the importance of evaluation of the narrative data on top of quantitative data. The development of the internet with its instant access to content and a further increase in computational power leads to accelerated progress in the field. Tetlock (2007) is among the first ones to study the connection between the content of news reports and stock market activity. García (2013) shows that the sentiment of the news content is able to predict stock returns mostly in recessions. Ahmad et al. (2016) show that media expressed tone, depending on the market regime, can have a different impact on firm-level returns. Manela and Moreira (2017) construct a text-based measure of uncertainty - news implied volatility - and show its relation to the stock market moves. Bybee et al. (2019) depart from the sentiment study and show that attention paid to the specific topic(s) of news is able to explain various economic and financial indicators. I differ from them both in terms of markets studied and methodology. While they rely on the latent Dirichlet allocation (LDA) method, which is only suitable for topic detection, the methodology I use is more versatile, allowing additionally for sentiment detection and meta topic classification. For the interested reader, Guo et al. (2016), Loughran and Mcdonald (2016), Loughran and McDonald (2020), and Marty et al. (2020) offer comprehensive reviews on the field.

At the same time, as attention paid to cryptocurrency increased tremendously over the recent years, the literature aiming at understanding the behavior of this class is also blossoming. Liu et al. (2019) study the cross-section of cryptocurrency returns and show that most of the variation is captured by three factors: cryptocurrency market, size, and momentum. Han et al. (2021) use almost stochastic dominance (ASD) to build 13 factor portfolios that dominate over 4 widely used benchmarks. Liebi (2022) shows that the fundamentals of the underlying blockchain technology are helpful in explaining the price movements. Naturally, the discussed progress in the NLP methods leads to their wider use in cryptocurrency research. For example, Chen et al. (2019) create a vocabulary specific to this asset class and show that the impact of the individual investor sentiment, inferred from social media posts, is conditional on the market regime. Also working with individual investor sentiment, now measured from the online Bitcoin price discussions, Kantorovitch and Heineken (2021) concentrate on the interaction between investor disagreement and market activity. While they center on user-generated content purely dedicated to price movements, I focus on the fundamental information, conveyed in a different source of text data - news. The two works that are the closest to my research are the following. Corbet et al. (2020) study the impact of macroeconomic news, following the announcements of major macroeconomic indicators, on Bitcoin returns. Lyócsa et al. (2020) focus on the impact of various types of news and events on Bitcoin volatility, but only one of them (regulation) is text-based. Both my methodology and research question are different: I concentrate on the full corpora of news, rather than one specific topic, and rely purely on the unstructured text data to extract all my factors.

3.3 Methodology and Data

Finance research concentrates on three main sources of linguistic content: corporate-produced documents, news, and user-generated content (UGC). The decision of which type is the best fit for every specific analysis is based on the unique characteristics of each. With cryptocurrencies being the main focus of this paper, corporate-produced texts have very limited potential. At the same time, it was shown by Leung and Ton (2015) that UGC is a more relevant source of linguistic information for small-cap stocks. Assuming that this property is transferable to the crypto space, USG is better suited for studying altcoins rather than major cryptocurrency players like Bitcoin. On the opposite side, the news is very flexible and universal data that conveys consensus information. Consequently, I concentrate on the news articles' potential in explaining Bitcoin price movements. Unlike using market data to explain the returns, relying on news content allows explaining the price formation with economically interpretable fundamental drivers, which is a particularly hard task for cryptocurrencies.

The relationship between news sentiment and asset return is well-established both by empirical and theoretical research. I take the next step, separating this dependency into purely descriptive and explanatory. I call the articles that describe the realized market movements purely descriptive.[2] On the opposite side, the linguistic content of explanatory news is related to something more than the realized price path of cryptocurrency. This type of news conveys information about what can be considered the fundamentals of the coin. News about the tighter cryptocurrency regulation or wider acceptance of Bitcoin is an example of fundamental

[2] I will use terms "descriptive" and "endogenous" interchangeably when referring to this type of news.

news.[3] While the purely descriptive part offers a sanity check for the used NLP method, the main economic interest is concentrated in the explanatory part. From this, arises the first research question:

Q1: Is the relationship between news sentiment and Bitcoin returns purely descriptive or does it have explanatory power?

I consider a positive answer to the explanatory power question as the demonstration that variation in Bitcoin price has (at least partially) fundamental reasons. To solidify this claim, I perform further analysis, following the reasoning given in Tetlock (2007). The author argues that if the media sentiment is a proxy of the new (or at least not yet priced in) fundamental information about the asset, then the sentiment should help to predict future returns, and the price impact of a sentiment shock should not be reversed in the following days. I conduct this experiment and find that this is indeed the case for the sentiment of the fundamental news. Thus the obtained result is consistent with the information theory of media tone. From that, I conclude that exogenous news reveals the information about the fundamentals of the coin, and thus has causal power on Bitcoin returns.

Furthermore, I decompose fundamental news into a few categories, based on which aspect of cryptocurrency the linguistic content is dedicated to. Naturally, a challenge arises: are all fundamental news equally important, or some can explain price movements better than the others? With 4 fundamental topics being Technology, Regulation, Adoption, and Macroeconomic situation, each covering a separate aspect of cryptocurrency properties, this leads to the following research question:

Q2: Which aspects of Bitcoin are the most significant for price formation?

Together with the different areas of cryptocurrency properties, I consider a complementary dimension. The academic literature has long debated whether Bitcoin can be considered money. The opponents of such recognition (Yermack, 2015) stressed that due to its high volatility, low correlation with widely used currencies, technological risk, and retail inconvenience - multiple zeros after decimal point needed to express the consumer prices - Bitcoin is rather a speculative investment than money. On the other side, adherents of such perception (Hazlett and Luther, 2020) argued that an item can be considered money if and only if it serves as a medium of exchange. The authors argue Bitcoin is money, as they show that its demand is comparable to the demand for many government-issued monies. The recent adoption by El Salvador as a legal tender turns the scales towards the acknowledgment of Bitcoin as a currency. Thus, Bitcoin should satisfy 3 functions of money:

- Unit of account

[3] I will use terms "fundamental", "exogenous", and "explanatory" interchangeably when referring to this type of news.

- Medium of exchange
- Store of value

Previous research used the statistical properties of the coin and transactional data to examine which of the 2 functions: store of value or medium of exchange - is more pronounced. Up to my knowledge, I am the first to use textual data to offer insight on this question, studying the investors' attitude expressed in the news. With the proposed NLP approach, I am able to classify fundamental news into one of the 2 corresponding categories and thus answer the final research question of this paper:

Q3: Which function of money - store of value or medium of exchange - are Bitcoin investors most interested in?

3.3.1 Methodology: news classification

In this section, I present an algorithm to perform a semi-supervised news classification. The SBERT network allows to transform each text into a 768-dimensional embedding vector, and thus the usual algebraic operations could be applied to it. In particular, one is able to find the distance between the embeddings of a given news article and a pre-specified text, which leads to a natural classification rule. This approach allows me to detect the topic of an article, as well as the sentiment. In the following subsections, I will zoom into all the steps of the process.

SBERT: from text to vector

Multiple methods in NLP are focused on mapping textual data into a vector form. One intuitive approach would be a bag-of-words algorithm, which transforms a given text into a vector, with entries being the frequencies of each word encountered in the text. A significant drawback of this approach is the sparsity of the obtained vector: its dimensionality should correspond to the total number of distinct words encountered throughout the entire text corpora. Thus for each separate text most of the entries would be 0s. As a consequence, this method is poorly suited for identifying the semantic similarity between data points: the distance between sentences "We like movies." and "My family enjoys cinema." will be the same as the distance between "We like movies." and "Everyone hates TV shows.". Clearly, it is not an acceptable result, and this impediment is especially pronounced when working with short texts.

More complex NLP algorithms are able to tackle the sparsity problem, for example, the Word2Vec model, introduced in Mikolov et al. (2013). A shallow neural network is used to embed each word into a vector of moderate dimensionality (usually from 100 to 1000). This allows a researcher to incorporate the notion of synonymity between different words: embeddings of "movie" and "cinema" will be located close to each other in the vector space. However, this model does not take into account the context. For instance, the embedding of the word "security" will stay the same whether one means it as "asset" or as "safety".

Recently introduced by Google (Devlin et al., 2019), the BERT model takes care of this issue. So far, the BERT family is considered to be a state-of-the-art technique in Natural Language Processing, which is capable of embedding a text into a dense vector, taking into account the context, the punctuation, and the order of words. I am using the Sentence-BERT (SBERT) (Reimers and Gurevych, 2020), which is a modification specifically tailored for the task of identifying semantically close texts. The produced embeddings of similar texts are close to each other in the vector space, and thus a distance measure (such as cosine similarity, for instance) between the embeddings will reflect the semantic similarity of the initial texts.

Define the news topic

Given the news database, the choice has to be made: which part(s) of an article should be used for the analysis? The most common choices are either the full text or the title of an article. However, I find that neither of these choices is optimal. In the text body, we often encounter deviations from the main topic: authors usually try to paint the global picture, recalling some previous details, mentioning related events or stories, and giving opinions or predictions. This dilutes the main innovative input of an article. On the other hand, using only the title has its own challenges: sometimes the title is just not detailed enough, too short, or too clickbait to convey the article's point.

At the same time, multiple news sources - including the one I am using for the cryptocurrency news - provide a brief 2-3 sentences summary, that is usually a continuation of the title, adding the essential facts to present a condensed overview of the article. Combining the title and the summary (I call this "top") tackles both abovementioned disadvantages. The obtained text is short (usually less than 4 sentences) and is always dedicated to the main point of an article exclusively as there is simply no room for distractions. Also, it is detailed enough to convey the message clearly, unlike the title or summary alone. I embed the tops of all articles using the SBERT network, which results in one 768-dimensional vector per article.

The following step is classifying the obtained vectors into predetermined categories. I assume that all news about cryptocurrency could be classified into 6 essential topics (4 fundamental and 2 endogenous):

- Technology. This cluster unites the news covering technological aspects of the cryptocurrency: mining, security and hacker attacks, hardware and software related information.
- Regulation. This group of news is dedicated to the legal status of cryptocurrencies and blockchain technology in different countries.
- Macroeconomic situation. This cluster includes news about central bank policies and plans for cryptocurrencies, economic situation, or global events, that may not be related to the crypto world directly, but could impact the valuations: inflation, pandemics, central bank announcements, conflicts, and wars.

- Adoption. This cluster covers news about the integration of cryptocurrency and blockchain technology into the society, for instance, authorization for making payments in bitcoins or using blockchain to secure elections.
- Past Prices. Multiple news articles are dedicated to the market analysis and discuss the cryptocurrency's past moves.
- Price Predictions. Similarly to the previous one, this cluster covers crypto market forecasts.

Isolating the 2 last clusters plays a crucial role in mitigating the endogeneity concern. Indeed, news items in cluster "Past Prices" simply describe historical market moves, which naturally leads to the high correlation between the return and the sentiment of this cluster. The cluster "Price Predictions" is subject to the same concern, even though to a lesser extent: when the main topic of an article is the price forecast, such prediction is usually built upon the analysis of the previous price moves rather than on the fundamental reasons.

To identify each cluster, I collect the words, phrases, or sentences that in my opinion characterize the cluster. For example, the set for cluster "Regulation" contains phrases "ban of cryptocurrencies" and "court order" among others, while "malware", "hacker attack" and "mining" belong to the set for cluster "Technology". I embed all the textual items that I consider defining for the cluster and subsequently average the embeddings across each cluster. This leaves me with one average embedding per cluster, and I consider these embeddings to be the representative of the corresponding group. Afterwards, I compute the cosine similarity between the embedding of a news article and each of the clusters:

$$sim_j = \frac{a \cdot e_j}{\|a\| \|e_j\|} = \frac{\|a\| \|e_j\| \cos\theta}{\|a\| \|e_j\|} = \cos\theta \tag{3.1}$$

where a and e_j are 768-dimensional vectors of embeddings of a news article and the cluster j respectively, $a \cdot e_j$ is the inner product in the corresponding vector space and $\|\cdot\|$ defines the euclidean L_2 norm. Cosine similarity is equal to the cosine of an angle between the two vectors and is bounded between -1 and 1. Cosine similarity of 1 means that vectors are fully aligned, which corresponds to maximal linguistic resemblance. Cosine similarity of -1 signifies that the embedding vectors have opposite directions, so the corresponding texts have opposite meanings. I attribute an article to the cluster with the highest cosine similarity.

I conclude this subsection with a short comment about an alternative way of topic detection: latent Dirichlet allocation (LDA). This statistical method has gained tremendous popularity among finance researchers in the area of textual analysis. It is an unsupervised clustering algorithm, which allocates each of the texts in a given corpus into one of the predetermined number of topics. At the same time, it has the following limitations:

- LDA is a statistical model: it detects terms that are often used together and assumes that such set of terms indicates a common topic. This frequentist approach performs poorly on the short documents.
- Often, a big number of topics is needed to obtain a meaningful result. For instance, Bybee et al. (2019) end up with 180 topics. As I am aiming for a sparse model, this is undesirable.
- The LDA method does not take the context into account, which yields lower accuracy. For instance, depending on the context, the word "exchange" could mean either the platform where the assets are traded (as, for example, NYSE), or the direct process of the good changing hands (as in the function of money "medium of exchange").
- In the classical LDA framework, topics are chosen by the algorithm. Its modification, seeded LDA, allows a researcher to predefine the topics of interest by providing the key terms for each topic, transforming the approach into semi-supervised. At the same time, this method can seed keywords (as the whole algorithm is built on the frequencies of the terms), but not phrases. Thus it is challenging to build topics that reflect the specific interactions between the objects of interest. Such a method could be used to classify the global corpus of texts into major topics (for instance, it would be able to select cryptocurrency-related articles from the global news database), but is less sensitive to distinguishing subtopics (such as future cryptocurrency price forecasts).
- Finally, LDA is not well suited to define text sentiment.

These arguments demonstrate that for my task the LDA procedure is not an optimal choice.

Sentiment detection

Apart from belonging to a specific topic, each article carries a sentiment about this topic. With sentiment being conceivably the most used text characteristic in finance, the task of sentiment detection is addressed by multiple NLP algorithms. For instance, the FinBERT network is considered to be on the cutting edge of sentiment detection for financial texts, as it is fine-tuned on the large corpus of labeled field-specific texts. However, the performance of such a sophisticated algorithm is still far from perfect. While it is able to correctly identify the sentiment of simple sentences ("The economic situation is improving." is classified as positive, and "Liquidity has dried up and the bid-ask spread has surged." as negative), slightly harder sentences like "The CDS spreads are increasing." are wrongly classified as positive by the network. In general, this and similar models for sentiment detection that were trained on labeled data are prone to excessive emphasizing of the verbs signifying increase: sentences like "X goes up." are often erroneously classified as positive, without paying enough attention to what exactly is X.

Independently from the mentioned FinBERT imperfection, another concern stems from the nature of cryptocurrency, seen by many as a safe haven asset. Even news that is correctly classified as negative for the classical financial system could be positive for Bitcoin and vice versa. For instance, Corbet et al. (2020) discover a negative relationship between Bitcoin returns and some types of generally positive macroeconomic announcements. On the other side, during certain episodes Bitcoin suffers from global uncertainty together with the classical markets. For instance, on March 11-12, 2020 when the World Health Organization announced COVID-19 pandemics Bitcoin experienced a sharp drop from 7,900 USD/BTC to 4,900 USD/BTC, reacting to the global news along with the stock market.

I propose to overcome these complications using SBERT. Since this NLP method generates semantically meaningful embeddings, the language features are reflected in the vector properties that could be treated algebraically. Let us consider the word "king" as an example (Mikolov et al., 2013). It incorporates multiple information directions (for instance, it is a human being, of the male gender, that has power over its kingdom and so on). What will happen if we switch the gender of the "king" from male to female? We will obtain the "queen": the word that means the same person as the king, but of the opposite gender. And the same holds in the embedding space:

$$\text{emb}(king) - \text{emb}(male) + \text{emb}(female) \sim \text{emb}(queen) \tag{3.2}$$

Thus if we would like to isolate the gender direction we can simply consider the difference between the embeddings of "queen" and "king". It will allow us to ignore other properties, such as belonging to humanity, reigning power, etc., and isolate the specific dimension of interest: gender. Then taking cosine similarity between the obtained difference vector and embedding of any word will allow us to derive inference of gender which this word signifies. Words that convey no information about the gender (like "table" or "jam") would be orthogonal to this direction and thus would score around 0.

Given this logic, I follow the path of Jha et al. (2020) to define the sentiment of the article. I construct two sets: one containing positive sentences about Bitcoin and cryptocurrency, and the second one containing negative.

I average the embeddings within each set and take the difference. The resulting vector defines the positive sentiment direction:

$$pos\_sent = \frac{\sum_{i=1}^{i=n} pos_i}{n} - \frac{\sum_{i=1}^{i=n} neg_i}{n} \tag{3.3}$$

where pos_i is the embedding of i-th phrase in the positive set, and neg_i is the embedding of i-th phrase in the negative set. Then cosine similarity between the defined sentiment direction and a news item assigns sentiment to the article. The positive value of the cosine similarity indicates that the article embedding points along the positive sentiment direction, while the negative value suggests that the direction of the article embedding is opposite to the isolated positive sentiment one. Neutral articles are orthogonal to the given direction in the embedding space and thus their cosine similarity with the sentiment direction is close to 0. I

Positive set	Negative set
Cryptocurrency provides a hedge against inflation.	Cryptocurrency is a risky investment.
Bitcoin guarantees security and privacy.	Bitcoin is not regulated and can be hacked.
Bitcoin is a profitable investment.	Bitcoin is loosing its value.
Cryptocurrency appreciates through time.	Cryptocurrency price is going down.
Government regulation favors bitcoin.	Government prohibits using cryptocurrency.
Global situation is unstable and bitcoin is a safe bet.	Cryptocurrency is useless.
Economy is in recession and crypto market is benefiting from it.	Crypto market suffers from global instability.
Cryptocurrency is popular.	Cryptocurrency is not popular.

Table 3.1 – Positive and negative sentiment sets

provide examples of news articles classified as well as the sentiment scores in the appendix E.

3.3.2 Methodology: price factors

In this section, I transform classified news articles into numeric time series. For each topic, I compute an attention-weighted sentiment score. I use the number of views for each article as a measure of attention to the given article. I rely on the observation that higher attention is associated with a stronger potential impact of this specific article on the price movement. Causality could be ambiguous: either a higher number of views could be explained by the high importance of the information contained in the text, in which case the content is the driver of the return. Or the increased attention could be due to the reasons, not related to the quality of information per se. In this case, attention becomes a self-fulfilling prophecy about content relevance. This also results in the monotonic relationship between the attention to the article and its potential to impact the market.

Formally, I denote by n the total number of news in the whole database. Each news article a_i now is represented by a vector (t_i, c_i, s_i, v_i), where:

- t_i is the date and time when a_i is published;
- c_i is the topic cluster, to which a_i belongs. Denoting the set of all topics as $\mathfrak{A} = \{\text{Technology}, \text{Regulation}, \text{Macro}, \text{Adoption}, \text{Past Prices}, \text{Price Predictions}\}$, $c_i \in \mathfrak{A}$.
- s_i is the sentiment score;
- v_i is the number of views of the article.

Then for the topic $C \in \mathfrak{A}$ during the time period τ the score S_τ^C is defined as attention-weighted average sentiment of all articles attributed to this topic, published during this period. Formally, I have:

$$S_\tau^C = \frac{\sum_{i=1}^n s_i v_i \mathbb{1}_{\{t_i \in \tau\}} \mathbb{1}_{\{c_i = C\}}}{\sum_{i=1}^n v_i \mathbb{1}_{\{t_i \in \tau\}} \mathbb{1}_{\{c_i = C\}}} \tag{3.4}$$

After obtaining the time series for all news classes, I compute total aggregate sentiment for the given period, across all clusters:

$$S_\tau = \frac{\sum_{i=1}^n s_i v_i \mathbb{1}_{\{t_i \in \tau\}}}{\sum_{i=1}^n v_i \mathbb{1}_{\{t_i \in \tau\}}} \tag{3.5}$$

In the similar manner, I compute the average views-weighted sentiment of the fundamental news S_τ^{fund}.

3.3.3 Data

The news dataset is scraped from the website cointelegraph.com. This website is consistently rated among the top-3 cryptocurrency news websites and is currently ranked as the №1 website covering blockchain and crypto assets. As of March 2022, this website averages 17 million views per month, while its close competitor coindesk.com reaches around 13 million.[4] Selecting a cryptocurrency dedicated website instead of screening news by topic from a global database offers an important advantage. The editorial board preselects only the relevant information, and at the same time, potentially impactful global news is also included. The high quality of the content and its credibility is guaranteed by the website's top ranking.

I scrap all news published on this website, starting from the date of website establishment in August 2013 up to June 2021. The website developed its full functionality (for instance, the possibility to share an article to the social network) only in 2015. Starting from this time, each article consists of the following components:

1. Date and time published.
2. Title.
3. Summary of the article. It is a short text following the title, that gives a concentrated overview of the article.
4. Number of views and number of shares.
5. Full content.
6. Category of news.
7. Tags. The website attributes multiple tags to facilitate search and navigation.

[4] Data from https://cointelegraph.com/advertise and https://www.coindesk.com/about/

I disregard non-textual items (with categories "Infographics" and "Podcast") as well as ignore the sponsored content. Also, I ignore digests since the contained information was already published as individual news pieces. Daily bitcoin price data is fetched from the Bitfinex exchange via data provider cryptodatadownload.com, which offers a wide selection of historical crypto price data. I perform the analysis on the time span from February 2015 to June 2021.

3.3.4 Descriptive statistics

The price dataset consists of 77 monthly observations. Bitcoin entered February 2015 with a price slightly above 220 USD/BTC, and slowly rose to reach the 1'000 USD/BTC level at the beginning of 2017. Then the first Bitcoin boom happened: the cryptocurrency increased 19-fold within one year and peaked at 19'783 USD/BTC on the 17th of December 2017. After a rapid crash in winter 2018 a period of relative oblivion and stagnation followed. In 2020, Bitcoin came back to the stage light and reached a price level of 40'000 USD/BTC in early January 2021. It peaked at all-time high level of 64'000 USD/BTC in April 2021 and then rapidly crashed. Bitcoin monthly return during the period of interest averages 6.5% per month, with the standard deviation being around 21%. The returns exhibit slightly negative skewness and slightly negative excess kurtosis.

Variable	Mean	Median	Std.	Skew.	Exc. Kurt.	Min.	Max.
Return	6.48%	6.40%	20.96%	-0.19	-0.22	-45.52%	50.26%

Table 3.2 – Summary Statistics: Bitcoin monthly return

This table reports summary statistics for Bitcoin monthly return. Data from cryptodatadownload.com. The sample period is from 2015-02-01 to 2021-06-30.

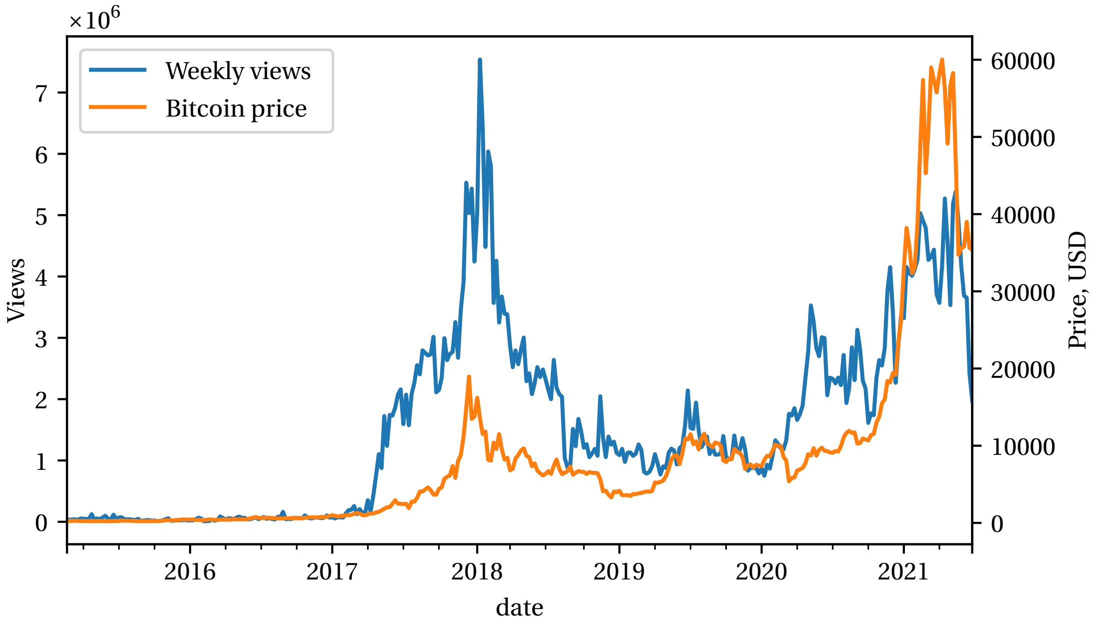


Figure 3.1 – Weekly number of the website views and Bitcoin price.

On the text side, I end up with 33283 news published in the given time span. The popularity of the website is quite low at the beginning of the sample: the total weekly number of views is below 200’000. A sharp attention increase happens in spring 2017. The number of views peaks in winter 2018, at the time of the first Bitcoin wave, reaching 7 million per week. The attention declines together with Bitcoin price fall. Interestingly, the popularity does not return to the previous peak levels even when Bitcoin price reaches an all-time high in spring 2021. The attention distribution in the sample is extremely skewed: while most of the articles have quite low popularity, a small number of articles contribute the most towards the total attention. While half of the articles were viewed less than 9’000 times each, the most popular news article has been viewed almost half a million times.

Variable	Mean	Median	Std.	Skew.	Exc. Kurt.	Min.	Max.
Attention	15’558	8’808	21’095	5.20	51.02	91	471’200
Sentiment	0.06	0.08	0.12	-0.50	-0.15	-0.40	0.43

Table 3.3 – Summary Statistics

This table reports summary statistics for attention (measured as the number of views) and news sentiment, computed by the procedure described in the previous section. Data from cointelegraph.com. The sample period is from 2015-02-01 to 2021-06-30.

In turn, the distribution of sentiment scores exhibits a moderate skewness. The average sentiment in the sample is positive, as well as the views-weighted sentiment mean, with the last one being slightly lower (0.05 compared to 0.06). This fact together with the negative correlation between sentiment and attention supports the hypothesis that pessimistic news receives more attention, as discussed in Ahmad et al. (2016). In general, sentiment is declining towards the end of the sample, but even during the major crush of 2018 it remains positive.

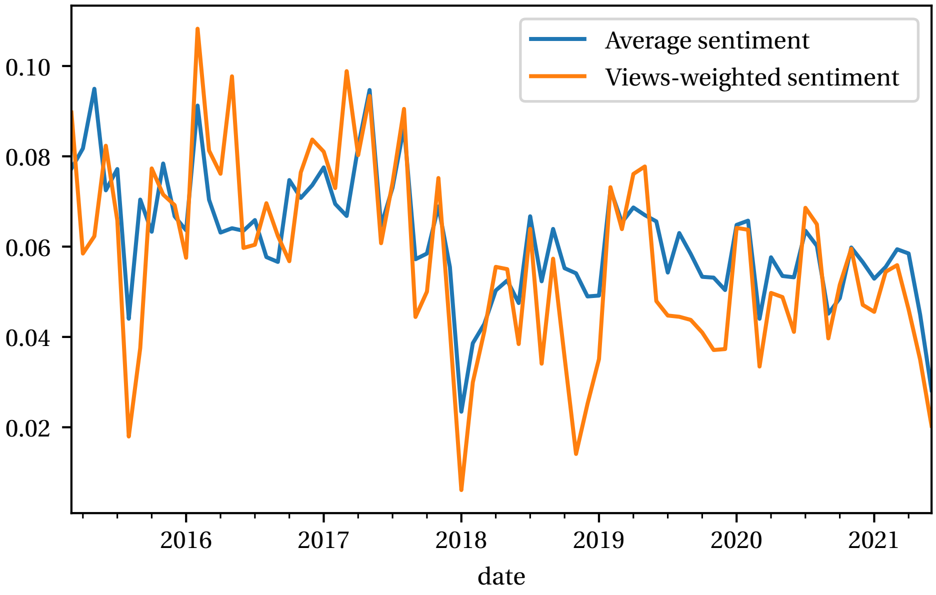


Figure 3.2 – Average monthly sentiment and views-weighted monthly sentiment.

3.4 Results

I perform most of the analysis on a monthly level. This horizon is the optimal choice for the cryptocurrency markets, with their inherently low signal-to-noise ratio. Even though some research follows the Adaptive Markets Hypothesis (Lo, 2004) and shows that Bitcoin market becomes more efficient with time (López-Martín et al., 2021), the consensus states that this market is still quite inefficient. This is related to slower information incorporation, which is additional motivation to use the low-frequency horizon.

3.4.1 Is there a fundamental variation?

To be able to draw statistically meaningful conclusions, I start by examining the constructed sentiment time series. To ensure stationarity, I double-test every data stream using both augmented Dickey–Fuller (ADF) test and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) (Kwiatkowski et al., 1992) test. While none of the processes has unit root, some are trend-stationary. All detrended time series are stationary, as confirmed by both tests. Afterwards, I standardize the detrended sentiments by the corresponding standard deviations. Finally, I apply ADF and KPSS tests to raw Bitcoin returns (plotted on figure 3.3), and ensure that this time series is also stationary.

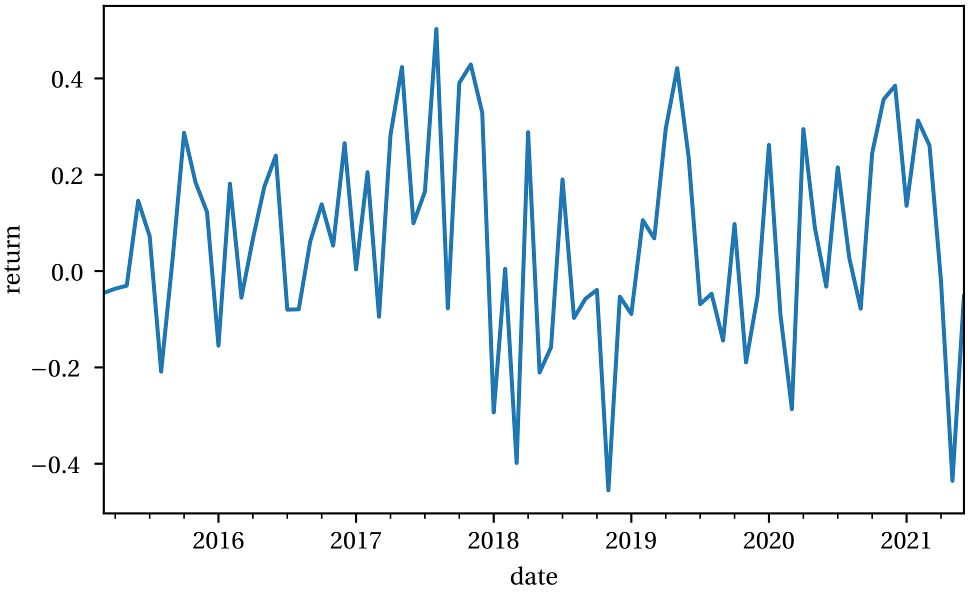


Figure 3.3 – Bitcoin monthly returns.

I start by examining the relationship between the return and sentiment:

$$r_t = \alpha_0 + \beta_0 * \hat{S}_t + e_{0,t} \tag{3.6}$$

where $\hat{S}_t$ is the detrended and standardized aggregate sentiment S_t and $e_{0,t}$ is the error term. I use Newey-West standard errors, robust to heteroskedasticity and autocorrelation up to 1 lag.

Following the large and ever-growing evidence on sentiment and return dependence, I expect to see a significant positive relationship. The result is repoted in table 3.4 and confirms the conjecture. The average sentiment is able to explain above 33% of return variation. This result is surprisingly close to the results obtained for the stock market in Bybee et al. (2019). In both cases, the endogenous news items are included, and purely text-built factors are able to explain a similar amount of variation, despite the difference in methodology and markets. The observed relationship is statistically significant at the 1% level.

However, further decomposition of the sentiment factor is required to draw a causal inference. With the factors built in the previous section, I project the returns on the space generated by the topic detrended standardized sentiments:

$$r_t = \alpha + \beta_1 * \hat{S}_t^{\text{Tech}} + \beta_2 * \hat{S}_t^{\text{Regulation}} + \beta_3 * \hat{S}_t^{\text{Macro}} + \beta_4 * \hat{S}_t^{\text{Adoption}} + \\ + \beta_5 * \hat{S}_t^{\text{Past Prices}} + \beta_6 * \hat{S}_t^{\text{Price Predictions}} + e_t. \quad (3.7)$$

This approach leads to a moderate increase in R^2, with full results reported in table 3.4. The cluster separation allows me to elevate the explainability from 33% for one aggregate sentiment to almost 38%. The Past Prices cluster and Price Predictions cluster, which aggregate news dedicated to the past and forecasted market movements respectively, alone can explain 31% of the variation in return time series. The coefficients are both positive and statistically significant at the 1% level. It is important to stress here that the factors rely only on textual data, and are constructed without any influence from the price data.

I drop all endogenous news as the next step to answer **Q1**. This leaves me with four clusters - Adoption, Technology, Regulation, and Macroeconomics - which I consider proxies for the fundamental sources of the variation in Bitcoin price. As before, I project Bitcoin returns on the space of the detrended standardized sentiments, but now *only* for exogenous news.

Results of this regression are reported in table 3.5. Variation in fundamental news sentiments is able to explain more than 16% of Bitcoin returns time series. Technology topic is significant at the 5% level, while Adoption, Macro, and Regulation are significant at the 1% level. All coefficients are positive. By construction, none of the news analyzed here is dedicated to the market moves per se (as I have eliminated that news in the previous step). Thus I exclude a purely descriptive channel of impact and can claim the causal interpretation.

But could the reverse causality be a concern? For instance, can high return *drive* the appearance of positive news in the given categories? I consider all clusters one by one and argue that this is not the case for any of them. First, by construction, cluster Macroeconomics contains the global news not related directly to the crypto market performance, but potentially influencing it. Thus the channel for this part of the cluster is clearly one-directional. For the Technology cluster, a plausible reverse causality hypothesis would be that higher returns provoke more hacks and security breaches. But this would go in the opposite direction of the observed positive relationship between the return and Technology sentiment innovation. At the same time, news about other aspects of the Technology cluster would simply cover the events and concepts that take more than 1 month to develop. Thus even if the reverse causality effect is present, it does not contaminate the performed analysis. Conceptually similar reasoning applies to the Adoption, Regulation, and the remaining part of Macroeconomics clusters: even though cryptocurrency moves may trigger the reaction and, for example, will motivate some companies to adopt Bitcoin as a mean of payment, preparation and implementation of such decision would take longer than the horizon considered in the analysis, thus such effect can not be present in the contemporaneous regression.

		r_t	
	(1)	(2)	(3)
const	0.065^{***}	0.065^{***}	0.065^{***}
	(0.020)	(0.018)	(0.020)
All news	0.122^{***}		
	(0.018)		
Past Prices		0.090^{***}	0.090^{***}
		(0.018)	(0.023)
Price Predictions		0.038^{*}	0.057^{***}
		(0.021)	(0.020)
Adoption		0.045^{***}	
		(0.015)	
Macro		0.032	
		(0.021)	
Regulatory		0.041^{*}	
		(0.021)	
Tech		0.012	
		(0.019)	
Observations	76	76	76
R^2	0.341	0.426	0.330
Adjusted R^2	0.332	0.376	0.311
Residual Std. Error	0.171(df = 74)	0.166(df = 69)	0.174(df = 73)
F Statistic	47.867^{***} (df = 1.0; 74.0)	13.669^{***} (df = 6.0; 69.0)	12.204^{***} (df = 2.0; 73.0)
Note:			*p<0.1; **p<0.05; ***p<0.01

Table 3.4 – Bitcoin returns explained by news sentiment

This table reports the results of the three models, each analyzing the explanatory power of the sentiment for the named groups(s) of news. Coefficients correspond to one standard deviation increase in the corresponding sentiment time-series. The dependent variable r_t is monthly Bitcoin return. Standard errors are in parenthesis. I use Newey-West standard errors, robust to heteroskedasticity and autocorrelation up to 1 lag.

	r_t
	(1)
const	0.065^{***}
	(0.021)
Adoption	0.044^{***}
	(0.017)
Macro	0.063^{***}
	(0.024)
Regulatory	0.040^{***}
	(0.015)
Tech	0.053^{**}
	(0.022)
Observations	76
R^2	0.208
Adjusted R^2	0.164
Residual Std. Error	0.192(df = 71)
F Statistic	6.881^{***} (df = 4.0; 71.0)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 3.5 – Bitcoin returns explained by fundamental news sentiment

This table reports the explanatory power of the sentiment for all exogenous news. Coefficients correspond to one standard deviation increase in the corresponding sentiment time series. The dependent variable r_t is monthly Bitcoin return. Standard errors are in parenthesis. I use Newey-West standard errors, robust to heteroskedasticity and autocorrelation up to 1 lag.

On top of the heuristic argument, I conduct an econometric experiment to support the claim made. Specifically, I examine whether the news I consider exogenous indeed carries information about the coin fundamentals. I rely on the summary of the media optimism theories, provided in Tetlock (2007). The three theories make different predictions about the impact of media tone on the market behavior, thus examining the market reaction to the media tone shock allows me to specify what exactly the media tone measures. The sentiment theory suggests that media tone reflects the shock to the noise traders' beliefs and is not related to the asset fundamentals. Assuming that noise traders' beliefs are stationary, this theory suggests that a negative shock to the media tone results in an increased trading volume and a temporary price decrease, which is reverted later. The noise theory suggests that all the information is already incorporated into the prices, thus a shock to the media tone will not impact the trading activity. Finally, the information theory of the media tone suggests that the news carries information about the fundamentals, which is not priced in yet. Thus, a positive shock to the media tone will be reflected in a positive price impact. Moreover, this price impact is permanent. At the same time, the information theory does not make a clear prediction about the impact on the trading volume. If the investors disagree about new information, the volume will react proportionally to the magnitude of the shock (measured as the absolute value of media tone innovation). On the other hand, if the investors agree about the new information, the volume will not react.

Thus, I examine whether the sentiment of the fundamental news is able to predict the returns. In this set up I analyze daily data, as the monthly or even weekly horizon is too low-frequency to gauge the predictive power. I use a vector autoregressive (VAR) model to estimate the relationship between returns, news sentiment, and volume.[5] Additionally, I control for the previous day's volatility.[6] I use the Bayesian information criterion to determine the optimal number of lags. This results in the selected model with 3 lags, and the equation for the return is:

$$r_t = \alpha_1^{VAR} + \beta_1^{VAR} * L3(r_t) + \gamma_1^{VAR} * L3(\hat{S}_t^{fund}) + \delta_1^{VAR} * L3(Vlm_t) + \lambda_1^{VAR} * vol_{t-1} + \epsilon_{1,t}, \quad (3.8)$$

where $L3$ is a three-lag operator (so that $L3(r_t) = (r_{t-1}, r_{t-2}, r_{t-3})$ for example), $\hat{S}_t^{fund}$ is the detrended standardized sentiment of the fundamental news, Vlm_t is the detrended log volume, vol_{t-1} is the lagged volatility of Bitcoin returns, and $\epsilon_{1,t}$ is the error term. My primary interest is the coefficient vector β_1^{VAR} and I report the results for this coefficient estimations in the table 3.6. All coefficients are positive, and the first sentiment lag coefficient is significant at the 10% level. The second sentiment lag coefficient is significant at the 5% level. The estimation results suggest that the new information takes two days to get fully incorporated,

[5] For the sake of stationarity, I use a detrended log volume. A trend is computed as the moving average of the past 60 days of log volume. The results are robust to the choice of this window: I also try 30 and 360 days moving average as a proxy for the trend.

[6] I proxy the volatility as squared demeaned return minus the moving average of the past 60 days of squared demeaned returns. The results are robust to the choice of this window: I also try 30 and 360 days moving average as a proxy for the trend.

	r_t
	(1)
$\hat{S}^{fund}_{t-1}$	0.001414*
	(0.000842)
$\hat{S}^{fund}_{t-2}$	0.001956**
	(0.000840)
$\hat{S}^{fund}_{t-3}$	0.000032
	(0.000843)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 3.6 – Predicting one day ahead Bitcoin returns with fundamental news sentiment.

This table reports the coefficients of lagged sentiment in the VAR return regression (3.8). Coefficients correspond to the effect of one standard deviation increase in the fundamental news sentiment. Standard errors are in parenthesis.

which confirms the hypothesis that Bitcoin market is rather inefficient.[7] The impact of the third sentiment lag is not significant. Figure 3.4 presents the response of the daily Bitcoin return to one standard deviation shock in the fundamental news sentiment over the following 10 days (impulse-response function). The shock effect wears off after 5 periods, and the return response stays non-negative during the whole period. Thus a positive shock in the fundamental news tone leads to a price increase, which is not reversed. The permanent price impact of the sentiment shock is consistent with only one theory of media tone: the

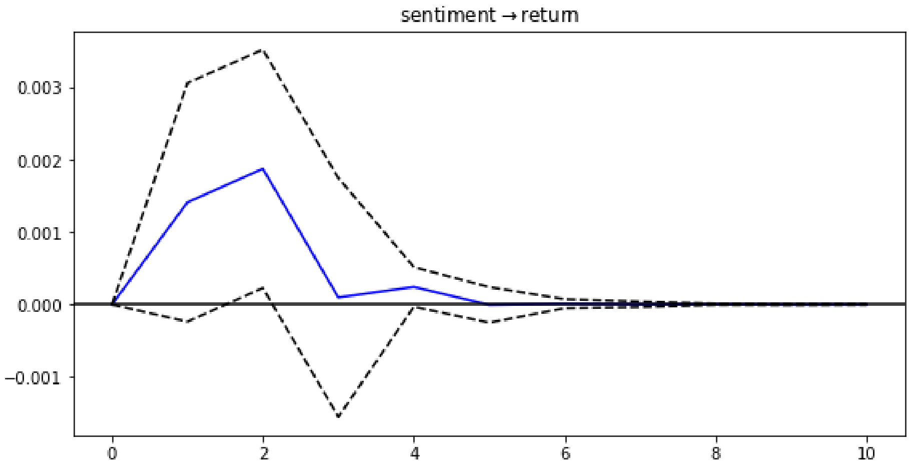


Figure 3.4 – 10-periods response of Bitcoin returns on one standard deviation shock in the fundamental sentiment.

[7] I also run a contemporaneous regression, to check the possibility that information is incorporated at the same day when it is released. It is not the case.

information theory. It follows that the exogenous news indeed carries information about the coin fundamentals.

Q1: Around 16.5% of the return time series is explained by the fundamental factors on the monthly level. Exogenous news has causal power on Bitcoin returns as this type of news reveals information about the coin fundamentals.

Finally, I examine whether the disseminated information creates a disagreement among the investors. To address this question I study the impact of the magnitude of sentiment shock on the trading volume, as my previous results support the information theory of media tone. Thus I augment the considered VAR(3) model with the absolute value of the sentiment, and focus on the equation for the trading volume:

$$\begin{aligned} Vlm_t = \alpha_2^{VAR} + \beta_2^{VAR} * L3(r_t) + \delta_2^{VAR} * L3(Vlm_t) + \\ + \gamma_2^{VAR} * L3(\hat{S}_t^{fund}) + \theta_2^{VAR} * L3(|\hat{S}_t^{fund}|) + \lambda_2^{VAR} * vol_{t-1} + \epsilon_{2,t}. \end{aligned} \tag{3.9}$$

Coefficients in θ_2^{VAR} measure the studied impact and are reported in the table 3.7. Consistently with the previous results, the market volume reaction is concentrated on the second day after the shock realization. The second lag coefficient is positive and significant at the 10% level. This reaction is in line with the hypothesis that Bitcoin investors disagree about the fundamental information contained in the media, even though to a limited extent.

	Vlm_t
	(1)
$\lvert\hat{S}_{t-1}^{fund}\rvert$	0.018717
	(0.012572)
$\lvert\hat{S}_{t-2}^{fund}\rvert$	0.023377*
	(0.012629)
$\lvert\hat{S}_{t-3}^{fund}\rvert$	-0.004144
	(0.012554)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 3.7 – Predicting one day ahead Bitcoin trading volume with fundamental news sentiment.

This table reports the coefficients of absolute value of the lagged sentiment in the VAR return regression (3.9). Coefficients correspond to the effect of one standard deviation increase. Standard errors are in parenthesis.

I proceed with a comment on the choice of the pure Bitcoin returns and not excess returns as a dependent variable in this study. The main reason for this decision is an unclear notion of the risk-free alternative for cryptocurrency assets. On one side, a valid option could be

Treasury Bills, which became a common benchmark in the finance literature. However, the considered period is associated with predominantly low interest rates, and thus choosing excess returns instead of pure returns would not change the analysis profoundly. At the same time, the cryptocurrency universe possesses an alternative supposedly risk-free investment: stablecoins. Such coins aim to offer a constant 1:1 exchange rate with the fiat currency the coin is pegged to. On top of that, some coins that operate with the Proof-of-Stake model can be staked: the holder's assets are borrowed to verify the transactions, rewarding the holder with the staking rate. Thus staking a stablecoin can be considered a cryptocurrency-specific risk-free alternative. However, the deafening crash of the third-biggest[8] stablecoin TerraUSD in May 2022 rises a multitude of questions about the risks associated with stablecoins. The presented ambiguity of risk-free assets in the cryptocurrency space motivates my choice of the dependent variable.

I conclude this subsection with a remark about the role that the attention weighting mechanism plays in the factor construction. For the two global groups of news clusters - exogenous and endogenous - the forces in play are different. For the endogenous news the number of article views plays a less crucial role. Since these articles describe the realized market moves, the corresponding informational content is already known. So the attention paid to each of these news pieces is not directly related to the potential impact on the markets. Thus, if the news items in the corresponding clusters are identified correctly, the smoother attention weighting (such as logarithm of the number of views, for instance) should result in the improved explainability of the returns. I run this exercise, and find that introducing logarithmic weighting boosts the return explainability of endogenous clusters from 31% to 38%. At the same time, for the articles in the exogenous group, the mechanism is quite the opposite. In general, the "informational noise" is dominating in media: by the estimate from Hafez (2009) about 80% of the news is adding noise, while only 20% is relevant. At the same time, as shown in Smales (2014), only the relevant news pieces have a potential for the market impact. Of course, I aim to construct the factors that contain as little noise as possible. Assuming that the noise articles are accompanied by a low number of views and given the highly skewed distribution of the number of views, the linear weighting essentially creates a natural filter, preventing non-relevant articles from playing a significant role in the factor construction.

3.4.2 Are all news created equal?

After establishing the causal relationship between exogenous news and Bitcoin return, a natural question follows: does the coin react more to some news than to the others? In other words, do some specific topics contribute more to the return variation? To answer this question, I project the returns onto the detrended standardized sentiment of each topic

[8] https://www.coindesk.com/markets/2022/04/19/terras-luna-surges-17-as-ust-becomes-third-largest-stablecoin/

separately:

$$r_t = \alpha_i + \beta_i * \hat{S}_t^i + e_{i,t} \tag{3.10}$$

where $i \in$ {Technology, Regulation, Macro, Adoption}. Comparing the models' R^2 and coefficient significance allows me to conclude the degree to which any given topic influences Bitcoin returns.

The results of this approach are presented in table 3.8. I observe that the most important news clusters are Adoption and Macroeconomics. Each of these clusters alone is able to explain around 5.6% in the return variation. While Adoption coefficient is significant at the 1% level, Macro coefficient is significant at the 5% level. The third cluster is Technology: alone, it can explain slightly more than 3% of the return variation and is statistically significant at the 5% level. The last cluster, Regulation, when taken alone, is not statistically significant.

			r_τ	
	(1)	(2)	(3)	(4)
const	0.065***	0.065***	0.065**	0.065***
	(0.024)	(0.024)	(0.026)	(0.025)
Adoption	0.055***			
	(0.021)			
Macro		0.055**		
		(0.024)		
Regulatory			0.024	
			(0.018)	
Tech				0.044**
				(0.021)
Observations	76	76	76	76
R^2	0.069	0.069	0.013	0.044
Adjusted R^2	0.056	0.056	-0.000	0.031
Residual Std. Error	0.204(df = 74)	0.204(df = 74)	0.210(df = 74)	0.206(df = 74)
F Statistic	6.720**	5.444**	1.718	4.213**
Note:			*p<0.1; **p<0.05; ***p<0.01	

Table 3.8 – Bitcoin returns explained by each topic sentiment

This table reports the results of the four models, each analyzing the explanatory power of the sentiment for the named exogenous category of news. Coefficients correspond to one standard deviation increase in the corresponding sentiment time-series. The dependent variable r_τ is monthly Bitcoin return. Standard errors are in parenthesis.

This distribution allows to identify the two primary drivers of Bitcoin valuations. First comes the social network effect. The net positive attention paid to the wider use of cryptocurrency and blockchain technology in general is the leading factor in explaining the returns. The close relation between cryptocurrencies and the outside world global events is highlighted by the Macroeconomics cluster being a close second. At the same time, technological aspects, though still significant, are playing a minor role in the valuations. The Regulation cluster deserves a separate attention in this regard. Being non-significant on its own, it becomes significant and contributes additional 2.6% to the R^2 when added on top of Technology, Adoption, and Macro clusters. This suggests that the effect of regulation news depends on the global situation. For instance, it is natural to suppose that after a major security breach or a loud money laundering scandal involving Bitcoin, a tighter regulation will lead to higher valuations. At the same time, tighter regulation in the absence of negative news and under the condition of wider adoption can result in lower valuations.

Q2: The topic with the largest causal explanatory power is Adoption. This topic is closely followed by Macroeconomic situation.

3.4.3 Which function of money Bitcoin investors care the most about?

While the previous news classification principle was based on the specific aspects of coin properties, in this section I concentrate on a complementary perspective. I take a step to understand why investors value Bitcoin. As Bitcoin is already adopted as a legal tender at least by one country, I conclude that this coin can be considered money, and so it should fulfill the functions of money. I assume that Bitcoin investors extract utility from the 2 functions of money: store of value and medium of exchange - and try to find which one is more important for them. The recent concerns about surging inflation and the government's ability to increase the money supply makes the store of value function of fiat currencies greatly debatable. At the same time, their medium of exchange function is highly valued by economic agents, with the liquidity argument being one of the strongest to explain the cash holdings in the presence of inflation and insured deposits programs.

The layout for Bitcoin is quite different: its supply is limited, and no entity can decide to just create more Bitcoins. Thus many people see it as a hedge for inflation, turning their attention to its store of value function. But high volatility and sudden crashes create an opposing force. At the same time, the supposed anonymity and independence from the global financial system make Bitcoin an attractive medium of exchange for multiple users. Thus it is not clear which function plays the key role in attracting investors to this asset. I use news articles to extract investors' opinions about these 2 functions of money and study which one has a stronger impact on the price formation.

I proceed in the following way: I classify all exogenous news into two meta topics, corresponding to the functions of money studied. While each of the previously discussed news clusters could contribute to either of the functions, the exact formulation of the article's title and summary allows drawing an inference about which meta topic is more pronounced in every specific case. I follow a new algorithm, that is similar in nature to the one described in section 3.3.1.

I use the same news article embeddings as for the topic classification. At the same time, I create two sets of phrases: one corresponds to the medium of exchange function of Bitcoin, and another corresponds to the store of value. Afterward, I define the medium of exchange-store of value direction, which is the difference between the average of the embeddings of phrases from the medium of exchange set and the average of the embeddings in the store of value set. Later, I compute a cosine similarity (which I will refer to as a "function score" in what follows) between the article embedding and the vector that defines this direction. Similar to the sentiment detection algorithm, described in section 3.3.1, this approach allows me to isolate one specific linguistic dimension of the text, corresponding to the property of interest.

The cosine similarity above 0 signifies that the given news article is more related to the medium of exchange function of Bitcoin, while the negative function score, in turn, indicates that the article takes the store of value point of view. Naturally, I consider only exogenous news in this experiment.

The majority of articles in the sample have function scores close to 0, signifying that these news items have a balanced view on the 2 functions of money, without taking a stance on a specific one.[9] I introduce a threshold ρ for the absolute value of the function score, which defines the classification rule: if the function score is below $-\rho$, then the article emphasizes a store of value point of view. The function score in the interval $[-\rho, \rho]$ does not give a particular signal about the function of money, and the function score above ρ signifies that the article pays more attention to the medium of exchange function. I choose ρ on the level of 0.01.[10] Table 3.9 demonstrates examples of articles that were classified in one of the 2 classes of interest: "medium of exchange" and "store of value". The sentiment of the news article is the one computed in section 3.3.1.

I notice that the store of value function consistently receives more attention from the investors than the medium of exchange. Interestingly, the patterns are quite different for the first and second Bitcoin rallies: while at the end of 2017 both functions received a quite similar amount of the attention, the 2020-2021 rally demonstrates the clear dominance of the store of value function, as presented on the figure 3.5. This result is in line with the idea that investors are concerned about inflation and government spendings fueled by the COVID-19 pandemic.

After classifying the news, I compute the views-weighted sentiment for each meta topic. I detrend both time series to ensure stationarity and standardize the obtained data. I project Bitcoin returns on the space generated by the detrended sentiment of each meta topic, mimicking the procedure in section 3.4.2. I compare the results obtained for each meta topic separately. The results are presented in the table 3.10. The coefficient in front of the store of value time series is positive and statistically significant at the 1% level. This function of money is able to explain 8.6% of Bitcoin returns. At the same time, the medium of exchange function

[9] This includes having no view at all.

[10] The results are robust to setting the threshold ρ at the levels of 0.005 and 0.025.

News	Date	Sentiment
Store of Value		
JPMorgan note to clients endorses 1% allocation to Bitcoin as a hedge.	2021-02-26	0.26
VC Expert Gurley: Bitcoin is Incredible Store of Value.	2017-11-18	0.28
Your crypto is not outside government reach, VC firm partner says.	2020-09-02	-0.04
Indian Police Warn Public Against Investing in Cryptocurrencies.	2019-01-03	-0.11
Medium of exchange		
Japan: Regulators Approve Startup's Bitcoin Sidechain Trial for Exchanges.	2019-01-21	0.11
Caricoin Ltd To Transform the Caribbean's Economy with Bitcoin.	2016-06-30	0.12
PayPal Sues Consumer Protection Agency for 'Confusing' Digital Wallet Rules.	2019-12-14	-0.09
Campaign Watchdog in California Bans Political Donations in Bitcoin.	2018-09-21	-0.07

Table 3.9 – Example of the news in 2 meta topics regarding functions of money.

This table presents headlines, date published, and sentiment of the examples of the articles categorized into 2 groups depending on the meta topic stance of the article. Full texts of the articles are available on cointelegraph.com.

of money is statistically significant at the 5% level and explains only 5% of the returns. Thus I answer the third research question:

Q3: The store of value is a dominating function of money for Bitcoin investors.

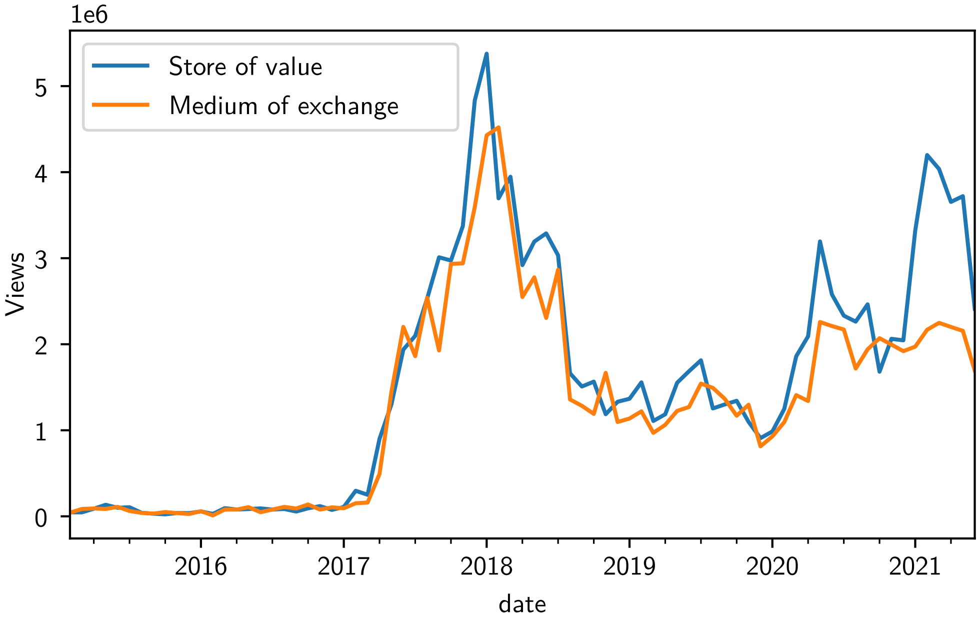


Figure 3.5 – Attention to medium of exchange and store of value functions.

	r_t	
	(1)	(2)
const	0.065^{***}	0.065^{***}
	(0.025)	(0.023)
exchange	0.053^{**}	
	(0.021)	
store		0.066^{***}
		(0.020)
Observations	76	76
R^2	0.063	0.098
Adjusted R^2	0.050	0.086
Residual Std. Error	0.204(df = 74)	0.200(df = 74)
F Statistic	6.389^{**} (df = 1.0; 74.0)	10.625^{***} (df = 1.0; 74.0)
Note:		*p<0.1; **p<0.05; ***p<0.01

Table 3.10 – Bitcoin returns explained by two functions of money.

This table reports the results of the two models, each analyzing the explanatory power of the sentiment for the two functions of money groups of news. Coefficients correspond to the effect of one standard deviation increase in the corresponding sentiment time series. The dependent variable r_t is monthly Bitcoin return. Standard errors are in parenthesis. I use Newey-West standard errors, robust to heteroskedasticity and autocorrelation up to 1 lag.

3.5 Robustness check

I conduct a robustness check for Q1 and Q2 using the following approaches. First, I use a different method to make the sentiment time series stationary. I repeat all the steps for Q1 and Q2 using AR(1) sentiment innovation instead of the detrended sentiment. It doesn't change my results qualitatively. On top of that, I make a change in the classification algorithm, so that some articles belong to 2 topics simultaneously.

More specifically, I introduce a confidence measure that indicates how certain the algorithm is that a given article belongs to the named topic. I attribute the articles with low confidence scores to 2 topics at the same time. The intuition behind this idea is as follows: if in 768-dimensional embedding space the distance from the article's embedding to the center of the cluster it is attributed to is only marginally smaller than the distance to the second-best cluster, then the article potentially can contain information about both topics. In this case, the confidence of the classifier should be low, indicating that the article can belong to both clusters. At the same time, if the first-best distance is significantly smaller than the second-best, then the confidence measure should be high, and the given article belongs only to 1 cluster. Formally, for each article a I compute

$$conf_a = \frac{sim_{j^*} - \max\limits_{j \in \mathfrak{A} \setminus j^*} sim_j}{sim_{j^*}} \tag{3.11}$$

where by j^* is denoted the cluster to which the article is attributed: $j^* = \arg\max\limits_{j \in \mathfrak{A}} sim_j$. Articles with low scores are attributed to both first-best and second-best clusters. I fix the confidence threshold for double attribution at the 5th percentile. The qualitative results remain unchanged: endogenous news explains a significant part of the return variation. I am still able to confirm the information theory of the media tone, and I find the evidence of mild investor disagreement on the released fundamental information. The most important topic is Adoption. The numerical results of the robustness check can be found in appendix F. These results are robust to the choice of the threshold for double attribution.

3.6 Conclusion

In this paper, I propose purely text-based factors to explain Bitcoin returns. A news database collected from a specialized website is classified in a semi-supervised manner using the SBERT network into predefined topics of interest. This algorithm results in efficient separation between exogenous and endogenous news. I demonstrate that exogenous news sentiment is consistent with the information theory of media tone, and thus contains information about coin fundamentals. I show that more than 16% of Bitcoin return variation is explained by the fundamental news, thus rejecting the idea that Bitcoin price is formed purely by speculation. I offer a comparison between the importance of different topics and conclude that wider adoption of cryptocurrencies and blockchain technology is the most important driver of

Bitcoin returns. Together with this, I study which function of money Bitcoin investors care more about by examining the views expressed in the media. I conclude that the store of value property is more valued by the investors than the medium of exchange.

Conclusion

This thesis focuses on the application of deep learning methods in asset pricing. The accent is on sequential data and natural language processing.

The first two chapters are dedicated to the ability of LSTM network to solve the rolling window problem, especially in a regime changing environment. We study two applications, both based on volatility estimation, where the proposed method outperforms the statistical counterpart. It would be interesting to know how generalizable is the obtained result and to study other potential setups that can benefit from the outperformance of LSTM. At the same time, more research is required to understand the exact mechanisms that contribute to the superior accuracy of the proposed deep learning method.

Another potential avenue for future research lies in the alternative network architecture. Recently introduced transformers have already replaced recurrent neural networks (LSTM is a special type of RNN) in the domain of natural language processing. In general, both these architectures are suited for processing sequential data. While RNN processes data points one by one, transformers are able to work with chunks of data due to their attention mechanism. It is curious to assess the capacity of the transformer network to solve the outlined rolling window problem in this light. Thus a comparison between the two architectures could lead to interesting results.

The third chapter uses the SBERT network to extract signals from text data and shows that such signals are helpful to explain the fundamentals of cryptocurrency price formation. Future research may include domain adaptation of the used NLP algorithm and expansion of the analysis to study a wider set of asset properties. Finally, the research on how linguistic properties of the texts are reflected in the properties of the corresponding embedding vectors would be of interest and could open wider possibilities for narrative-based asset pricing.

A Appendix: LSTM networks

This section provides a brief intuition about neural networks and an overview of the structure of the specific network we use.

Artificial Neural Networks (ANN or simply NN) are for now one of the most powerful and widely used tool to tackle complex machine learning problems. In essence, every NN is a sequence of non-linear data transformations. A network consists of units called *neurons*, which are hierarchically organized into *layers*. Each neuron in the network is associated with its own weighting vector W and bias b, which are the parameters that will be updated during the learning process. The neuron performs an affine data transformation (multiplying the input by its weighting vector and adding the bias), and then applies a predetermined activation function[1] to the result. All neurons of layer l take as input the previous $l-1$ layer's output, and in turn pass their own output as an input for the following layer $l+1$. Neurons within one layer use the same activation function and operate independently from each other. Formally, the output of neuron number k in layer l is

$$a_k^{[l]} = \phi(W_k^{[l]'} x + b_k^{[l]}) \tag{A.1}$$

where ϕ is the activation function that is applied elementwise and x is the output of all neurons of the previous layer. Passing the data through the network and obtaining the output is called *forward propagation*.

The goal of network learning is to find parameter values that will result in a minimal error between the network output and the desired output (in our case this is the labelled output). It is done iteratively. First, the input data is fed to the network and the output is obtained. Then an error function is computed, that shows how far is the network result from the target. The chosen activation functions being piecewise differentiable, a gradient descend method is used

[1] Theoretically, any function can be used as an activation. However, a piecewise differentiable function is usually preferred in practice.

to update the parameters. This process is called *backpropagation*. Repeating forward and backpropagation allows to adjust parameters in the way that results in increasing network performance (i.e. lower error function).

An example of a simple network is the logit regression. It has two layers:[2] The first one is the input layer, with the amount of neurons being equal to the number of regressors. The second layer is the output layer with one single neuron, that has the *sigmoid* activation function $\sigma(x) = \frac{1}{1+e^{-x}}$. Parameters of this neuron are just the regression coefficients. This network has no hidden layers (that is, layers other than input and output). Networks that have one or no hidden layers are called shallow, while networks with multiple hidden layers are called *deep*. Figure A.1 provides a visualization of a deep neural network.

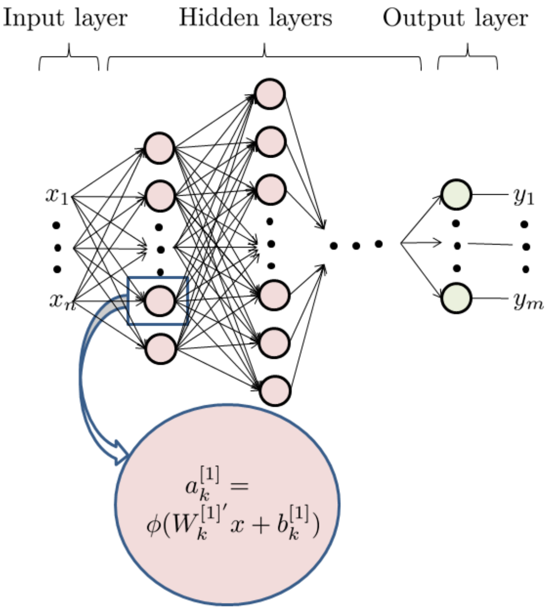


Figure A.1 – Deep Neural Network

The drawback of using a general network as described above is that it is incapable of learning temporal dependencies from time-series data. To address this task a *recurrent neural network* (RNN) could be used. The recursive layer of such network has an analogue of memory, called *hidden state*. It carries the information from the previous time step and is used as additional input for the recursive layer. Formally, it processes time t observation according to

$$h_t = \phi(Wx_t + Uh_{t-1} + b) \tag{A.2}$$

where x_t is the input, h_{t-1} is the hidden state from the previous time step and W and U are the corresponding weighting matrix for the input and hidden state respectively. h_t is the updated

[2] Due to conventions, it is called a one layer network since the input layer is usually not counted.

hidden state, that is kept to treat the next observation $t+1$ and also it serves as the output. Two major issues arise for such plain vanilla RNN. First, the memory is short-lived and not elective. There is no way to keep for a long time the important information and quickly forget the irrelevant one. The second issue, technical in nature, is the exploding/vanishing gradient. [3]

Long short-term memory network (LSTM) is a specific type of RNN, that mitigates both of these problems. An LSTM block has two states. One state corresponds to the working memory and is analogous to the RNN hidden state h_t. It is also the output of a block to the following network layer. The second one is the long-term memory mechanism, called the cell state and denoted by c_t. The block also has three gates (non-linear input transformation), that regulate the information flow inside:

(r) The forget/remember gate coordinates which information from the long-term memory should be kept and which one should be discarded.

(s) The input gate (or sometimes called save gate) decides which information from the input should be saved in the long-term memory.

(f) The output gate (sometimes called focus) controls the updates of the hidden state.

Each gate is in essence just a shallow neural network itself. The parameters associated with the **r**emember, **s**ave, **f**ocus gates respectively are given by the triplet (W_i, U_i, b_i) where $i \in \{r, s, f\}$.

To better grasp the intuition, let us walk along the transformation of the input x_t within one LSTM block. From the previous time step the cell state c_{t-1} and the hidden state h_{t-1} are passed.

1. This first step is dedicated to learning which information of the existing long-term memory will be kept or forgotten. To do so, the remember gate (r) uses x_t and the working memory h_{t-1} to obtain the “remember” vector r_t that is computed as

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r). \tag{A.3}$$

   Here σ defines the sigmoid activation function, that ensures values of the remember vector are between 0 (fully forget) and 1 (fully remember). r_t will later be elementwise multiplied by c_{t-1} to keep only the relevant information.

2. Now we decide which information should potentially be added to the long-term memory. The candidate is formed as

$$c'_t = \tanh(W_l x_t + U_l h_{t-1} + b_l) \tag{A.4}$$

[3] Intuitively, each RNN could be unfolded into a non-recurrent network of the same length than the data series. During backpropagation, the derivative of the error function with respect to weights should be computed for every node. Due to the chain rule, it results in iterative multiplication and thus the derivative may become unstable.

where (W_l, U_l, b_l) are the parameters of the **l**ong-term memory candidate formation.

3. Before it enters the long-term memory, the save gate (s) decides which part of this candidate is worth saving by computing the following quantity

$$s_t = \sigma(W_s x_t + U_s h_{t-1} + b_s). \tag{A.5}$$

As before, the activation function here is sigmoid, that ensures values between 0 and 1 and thus by elementwise multiplication allows to regulate the information flow.

4. We are ready to update the long-term memory by performing the following operation

$$c_t = c_{t-1} \otimes r_t + s_t \otimes c'_t \tag{A.6}$$

where ⊗ denotes an elementwise multiplication. The updated value of the long-term memory c_t consists of the information remembered from the past (first term) and the newly added component (second term).

5. Finally, we can update the hidden state h_t. The focus gate (f) allows to concentrate on the relevant information from the long-term memory. This is done as follows

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{A.7}$$
$$h_t = f_t \otimes \tanh(c_t). \tag{A.8}$$

The figure A.2 allows to represent those transformations visually.

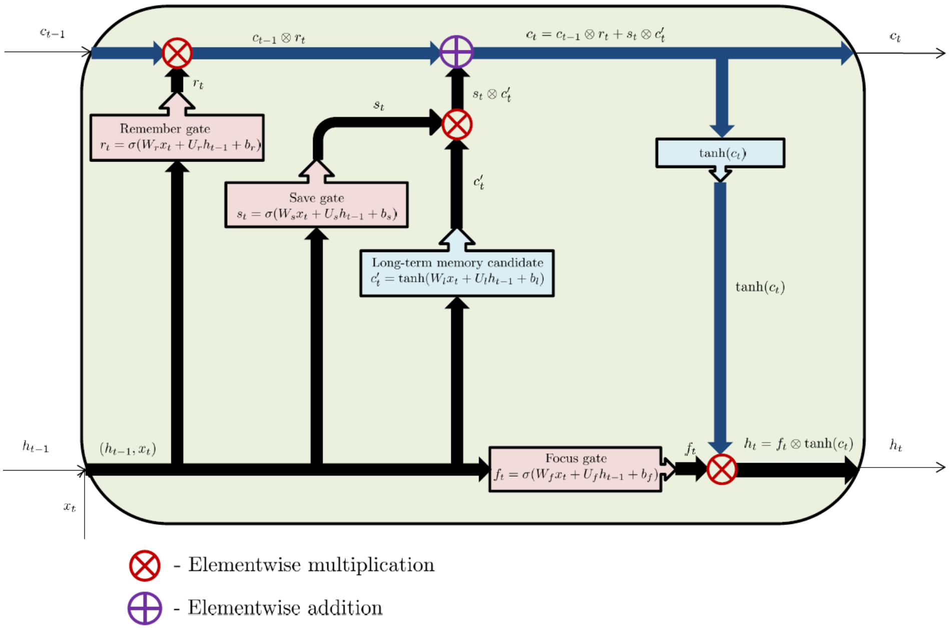

c_{t-1}
$c_{t-1} \otimes r_t$
$c_t = c_{t-1} \otimes r_t + s_t \otimes c'_t$
c_t
r_t
s_t
$s_t \otimes c'_t$
Remember gate
$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$
Save gate
$s_t = \sigma(W_s x_t + U_s h_{t-1} + b_s)$
c'_t
Long-term memory candidate
$c'_t = \tanh(W_l x_t + U_l h_{t-1} + b_l)$
$\tanh(c_t)$
$\tanh(c_t)$
h_{t-1}
(h_{t-1}, x_t)
x_t
Focus gate
$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$
f_t
$h_t = f_t \otimes \tanh(c_t)$
h_t
⊗ - Elementwise multiplication
⊕ - Elementwise addition

Figure A.2 – Transformation of the input inside the LSTM unit

Now that we have a brief intuition about how the LSTM block works, this section concludes with a brief description of our neural network architecture. The network consists of the following layers:

1. Sequence input layer, that inputs the time-series into the network.

2. Bidirectional LSTM layer with 200 neurons. This layer is learning long-term dependencies from the complete sequence

    - An LSTM layer, as discussed before, allows the network to keep track of the valuable information, that was encountered long time ago, forgetting the more recent but unimportant.
    - A bidirectional layer duplicates the LSTM layer, creating two such layers one after the other. The first receives the actual time-series as an input, while the second one receives the reversed copy of the data. This allows the network to use the whole dataset to classify points, including information that comes from the moments after.

3. Fully connected layer with two neurons, both of them being connected to all the neurons from the previous layer. Each neuron multiplies the input by the weight vector and adds the bias. This layer assembles all the features learned by the bidirectional LSTM layer to classify points into "jump"/"no jump" categories.

4. Softmax layer with two neurons, that applies softmax function [4] to the inputs, computing the probabilities of the point belonging to one of the two categories. If the outputs of the previous layer's two neurons are s_1 and s_2, the neurons of this layer will compute $p_i = \frac{e^{s_i}}{\sum_{j=1}^{2} e^{s_j}}$ for $i = 1, 2$.

5. Classification output layer, that computes the cross-entropy[5] for the classification problem for multiple non-intersecting classes in order to construct the error function (that will be minimized).

Schematic representation of our network can be found on the figure A.3.

[4] The softmax function (also called normalized exponential) takes as input a real vector and transforms it into a probability distribution. Formally, for a vector $v \in \mathbb{R}^n$, the softmax function $g : \mathbb{R}^n \to \mathbb{R}^n$ is defined as $g(v)_i \triangleq \frac{e^{v_i}}{\sum_{j=1}^{n} e^{v_j}}$ for $i = 1, ..., n$.

[5] The cross-entropy of two probability distributions $\mathbb{P}$ and $\mathbb{P}^*$ is defined as $H(\mathbb{P}, \mathbb{P}^*) = E^{\mathbb{P}}\left[-\log \mathbb{P}^*\right]$.

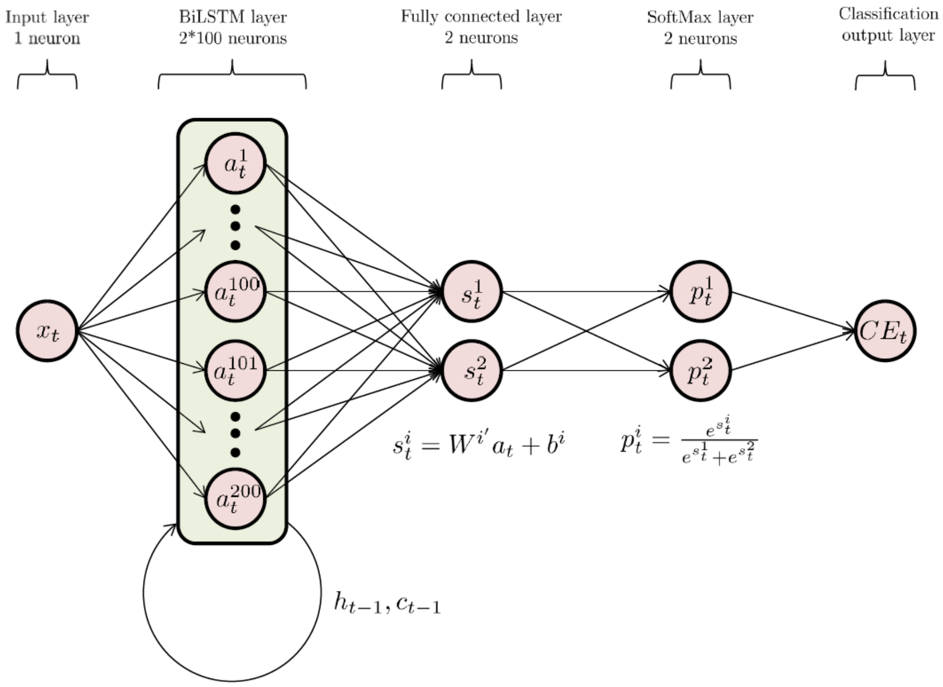


Figure A.3 – Network used in this paper

B Appendix

If the price process is represented by the jump-diffusion model (1.1), then the realized variation is defined as

$$\mathrm{RV}^2_{t+1}(\Delta) \triangleq \sum_{j=1}^{1/\Delta} r^2_{\Delta,t+j\cdot\Delta} \rightarrow \int_t^{t+1} \sigma^2_s ds + \sum_{t<s\leq t+1} Y^2_s \tag{B.1}$$

where Δ corresponds to the chosen frequency and $r_{\Delta,t+j\cdot\Delta}$ is the j^{th} log-return within day t. The realized variance converges (in probability) to the total variance composed by two terms. The first one being the continuous variance (generated by the diffusion term of the stochastic process) while the second is the jump variance.

In order to estimate only the integrated volatility, Barndorff-Nielsen and Shephard (2004) introduced the realized bipower variation which is defined as

$$\mathrm{BPV}^2_{t+1}(\Delta) \triangleq (\pi/2)^{-2} \sum_{j=2}^{1/\Delta} |r_{\Delta,t+j\cdot\Delta}||r_{\Delta,t+(j-1)\cdot\Delta}| \rightarrow \int_t^{t+1} \sigma^2_s ds. \tag{B.2}$$

BPV is a consistent estimator of integrated volatility in the presence of jumps. It is therefore useful to create a jump test.

C Appendix: Discussion on local martingales

In this section we aim at building intuition about true and strict local martingales. We will do so through the famous example of the doubling strategy. Imagine that a gambler is betting on the outcome of a coin toss. If the coin comes out head, he wins his bet. If it comes out tail he looses his bet.

Now assume that he chooses the following strategy: if he looses round n, he doubles his previous bet for the next round. He does so until he wins. As soon as he wins for the first time, he takes his gain and stops playing. Clearly, if the player is allowed to take infinite credit, he can bet endlessly until the coin comes out head and thus is guaranteed to walk away with a net gain of 1 dollar. However, for every finite number of trials it is a fair game: with high probability the player wins 1, and with very small probability he loses all his previous bets, such that the expected net gain is 0. Therefore his personal wealth is a true martingale.

However if the gambler is allowed to bet infinitely fast in a finite time interval, his wealth process turns into a strict local martingale and is expected to increase.

C.0.1 Betting in discrete-time

Let X_t^π denote the dollar value at time t of the portfolio that invests in the strategy π described above. We set $X_0^\pi = 0$ such that the gambler starts with zero initial wealth. Let Z be the random variable that represents the outcome of the coin toss. It takes the value 1 if the coin lands on head and -1 otherwise. The coin is unbiased such that $\mathbb{P}(Z_n = 1) = \mathbb{P}(Z_n = -1) = 1/2$. It follows that

$$X_n^\pi = \begin{cases} 1 & \text{if } X_{n-1}^\pi = 1, \\ X_{n-1}^\pi + (1 - X_{n-1}^\pi) Z_n & \text{otherwise.} \end{cases}$$

As we see, until the first time the coin comes out heads, the value of the strategy X_n^π is negative and the debt grows exponentially. As soon as the coin lands on head, the winning bet covers the previously occurred losses and provides the net gain of 1 dollar. It is easy to see that this is a martingale.

For every finite number of tosses N the probability of loss (i.e the probability of coin landing only on the tails all N times) is $\frac{1}{2^N}$ and the corresponding loss in this case is the sum of all the lost bets, including the current one, so it is $(-2^N + 1)$. The probability of winning is then $1 - \frac{1}{2^N}$ and the net gain in case of victory is 1.

Even though the expected gain stays zero in discrete-time independently of N, it is insightful to analyze what happens to the distribution as the number of bets rises to understand the continuous-time framework in the next section. For that, we simulate 10^7 sample paths of such game, each representing one different gambler. We present the histograms of the players' wealth after n rounds below. In line with the reasoning above, after the first coin toss in half of the cases the player leaves as a winner with \$1 and in the other half the player looses \$1 as displayed in figure C.1. Increasing the number of tosses to 4, we see in figure C.2 a shift in the distribution: most of the mass is concentrated on the winners side, with approximately $\frac{15}{16}$ of players being the winners of \$1 and $\frac{1}{16}$ of gamblers loosing \$15. This phenomenon accentuates as we increase further n: the probability of loosing decreases but the potential loss increases, thus decreasing the skewness of the distribution. The loss probability will disappear when we work in continuous-time, transforming the gambler's wealth into a strict local martingale.

While in theory the expectation of this discrete-time process is always 0, when increasing the number of trials (for example up to 25) we run into the *finite-sample problem.* The probability of loss becomes so small that our number of samples is just not high enough: simulations will tell us that among 10^7 players after 25 rounds there was no losers and everyone eventually walked away with \$1. In this case the average terminal wealth is clearly 1. Here we already approximate a strict local martingale behavior that should only exist in continuous-time. But this picture is misleading and arises solely due to the computational limitations. We should remember that this is still a fair game! The probability of the loss is, however small, still nonzero, and the potential loss is so huge that it balances the mass of winners in expectation.

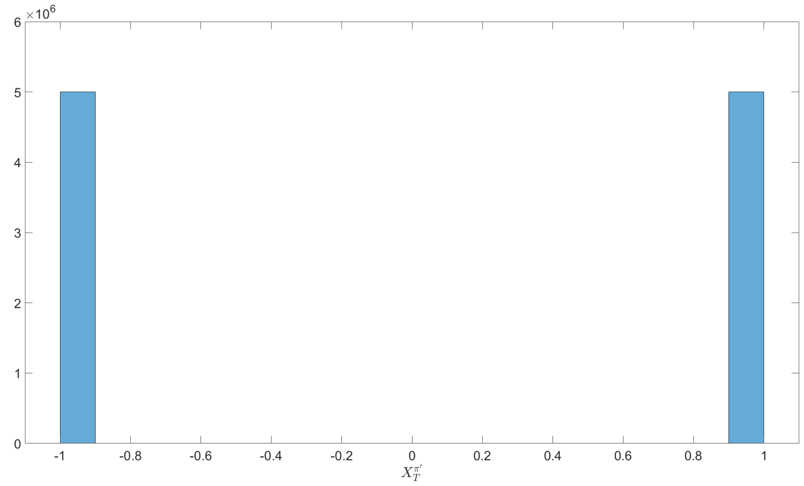


Figure C.1 – Distribution of the terminal wealth of the gamblers after 1 trial.

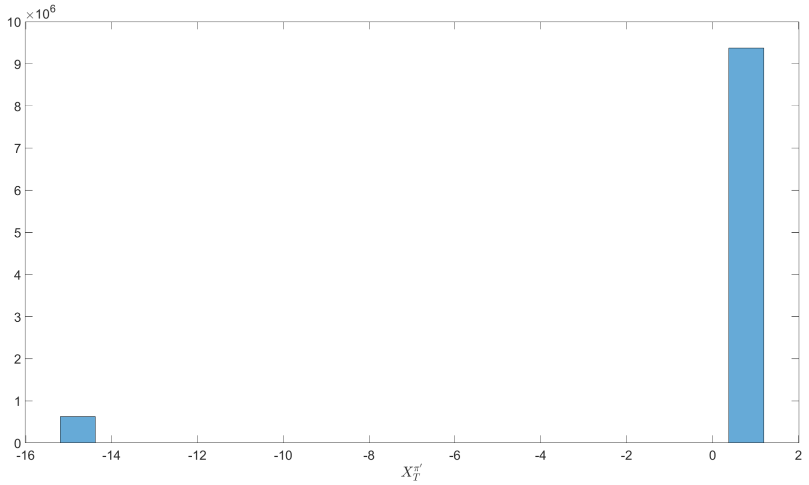


Figure C.2 – Distribution of the terminal wealth of the gamblers after 4 trials.

C.0.2 Betting in continuous-time

The previously described strategy is pathological in the sense that even though for every finite number of trial it is fair, it stops being fair as soon as we allow for infinitely many trials. This pathology is translated into the strict local martingale property of the continuous-time process. Imagine that the player can speed up the time, such that after each lost trial he can bet faster and faster. In such a fantastic world the player can place infinitely many bets (thus eventually win \$1) in a finite time interval (say 1 hour). Mathematically, we construct the following continuous-time process:

$$Y_t = \begin{cases} X_n^{\pi} & \text{if } 1 - \frac{1}{n} \leq t < 1 - \frac{1}{n+1}, \\ 1 & t \geq 1. \end{cases}$$

This process represents the dollar value of a portfolio which employs this doubling strategy. As a result, the player always ends up with the net gain of 1. This process is clearly not a martingale, since $\mathbb{E}[Y_0] = 0 \neq \mathbb{E}[Y_1] = 1$. However, it is a local martingale (under the localizing sequence of stopping times chosen as, for example, $\tau_n = \inf\{t : |X_t| \geq n\}$).

It is important to point that not all strict local martingales arise due to a doubling strategy. As pointed out in Dumas and Luciano (2017), in this example the strict local martingale arises due to the behavior of the agent (which bets faster and faster) while the underlying stochastic process Z (the coin) itself has always the same distribution after each round since the variable is iid. In the bubble detection part of this paper, the process (2.6) itself might be a strict local martingale irrespective of the betting behavior of the investor. Therefore it is important to understand that when we detect strict local martingales (i.e. bubbles) and implement a long-short trading strategy, we do not rely on a doubling mechanism.

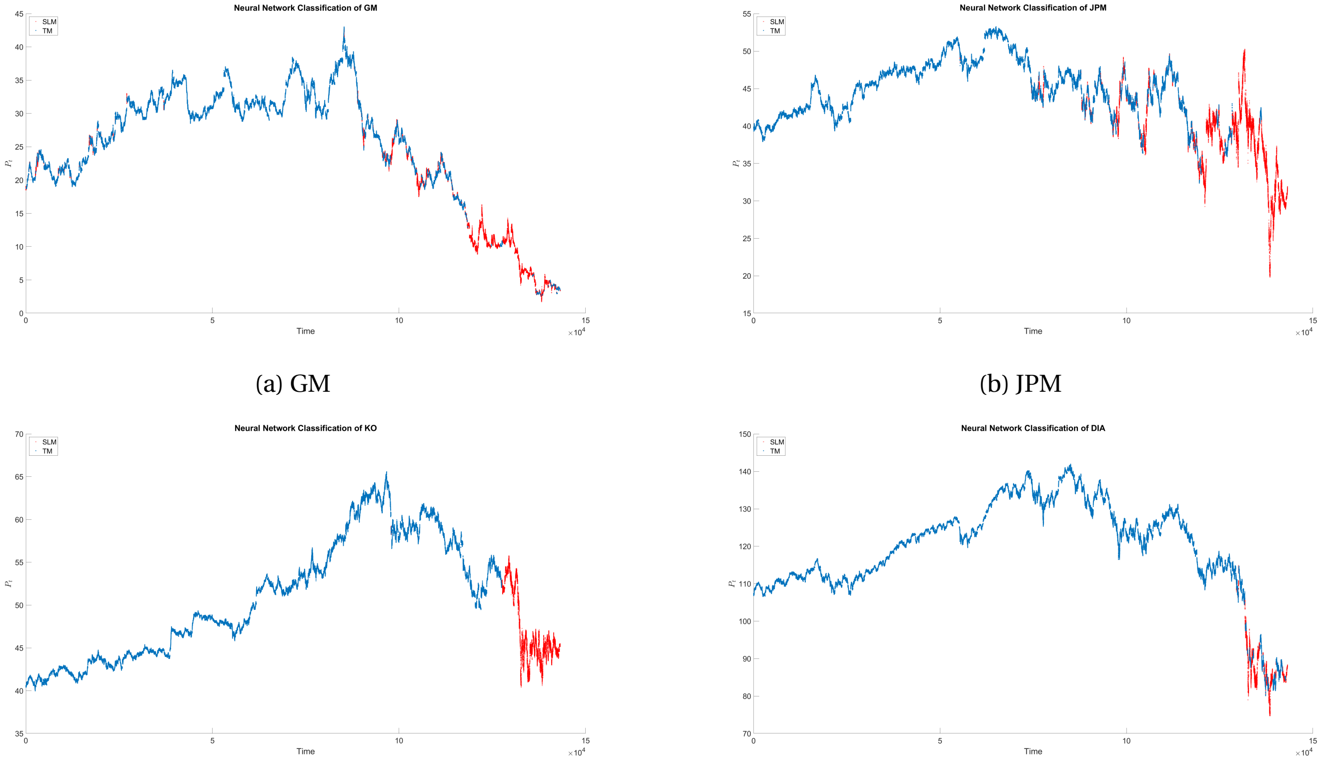
C.0.3 Approximation of strict local martingales by discrete-time processes

The final point of our discussion concerns the simulation of strict local martingales, a phenomenon that exists solely in continuous-time. Since computers cannot simulate continuous variables, we have to rely on a discretization scheme. So in fact, we are left with a discrete-time process, trying to study a phenomenon, that exists purely in continuous-time. However, this appears to cause no complication for our purpose. As we have seen in subsection C.0.1, the probability mass of the gambler's wealth is sparsely distributed. As n rises, a higher probability is concentrated on the winning point, while a very small and further decreasing probability is shifting to the left, corresponding to loosing higher and higher amounts. At some point the probability becomes so small, that among the whole simulation set there is not enough samples for this probability to realise and create at least one loosing path. As a consequence, we end up with a process X^{π} that approximates a strict local martingale behavior. Even though we still work in discrete-time and theoretically this process should be a martingale (staying at 0 in expectation), due to the absence of at least one loosing path the simulated process in fact increases on average.

The same logic applies to our simulations of the stock price S_t. We use Monte-Carlo to generate paths of the discretized version of the price in equation (2.13) where $\gamma_1 > 1$. Its continuous-time counterpart is a strict local martingale which is a supermartingale (since is bounded below by 0) and so has to go down in expectation. However even the simulated discrete-time process decreases on average while in theory it should not. This happens exactly because of an extremely sparse probability distribution as discussed in subsection C.0.1. With very high probability the process goes down (resembling an SLM) while with very small probability it goes extremely high and thus balances the average to stay a true martingale. Since we simulate a finite number of paths, this highly unlikely exploding path just does not appear, leaving us with paths that resemble the SLM.

If we choose a time interval Δt of 10^{-3} with maturity $T = 5$ for the simulations, we observe a decreasing average even if we simulate over 10^8 different paths. Decreasing Δt further for the same T will accentuate this phenomenon.

D Appendix



(a) GM

(b) JPM

(c) KO

(d) DIA

Figure D.1 – Bubbles detection from 2006 to 2008 included (3 years) using our LSTM network. The blue line represents the price under a non bubbly regime. The red line represents periods when the asset is in a bubble regime.

E Appendix: Examples of classified news

In this section I provide examples of positive and negative news for each cluster. With the given title, full text of the article can be retrieved free of charge from cointelegraph.com

News	Date	Sentiment score
	Adoption Positive	
$240 Bln Japanese Company To Pay Employees in Bitcoin.	2017-12-19	0.22
Distributed Lab To Launch a Bitcoin Wallet In Ukraine.	2015-11-10	0.06
PayPal's crypto integration means Bitcoin could triple its user base.	2020-10-22	0.26
	Adoption Negative	
Barclays Shuts Down CoinJournal Bank Account, Outrages Bitcoin Community.	2016-10-27	-0.14
China's Social Media Giant WeChat Blocks a Number of Crypto Media Accounts, Sources Say.	2018-08-21	-0.10
Microsoft president says fintechs should leave currency to central banks.	2021-03-24	-0.11

Table E.1 – Examples of positive and negative news in Adoption cluster.

News	Date	Sentiment score
Tech Positive		
Bitcoin Full Nodes Get Efficiency Injection With Newly-Released ‘Config Generator’.	2017-03-31	0.22
Super Bitcoin Hard Fork To Launch Tuesday To ‘Make Bitcoin Great Again’.	2017-12-11	0.09
Zcash Vulnerability Permitting Infinite ZEC Counterfeiting Fixed and Disclosed.	2019-02-05	0.10
Tech Negative		
Iranian crypto miners using household energy will face large fines.	2021-05-17	-0.13
Aviation Database Struck By Unknown Ransomware Gang.	2020-07-25	-0.08
Reddit user warns of a copy & paste exploit that stole his crypto.	2020-08-26	-0.08

Table E.2 – Examples of positive and negative news in Technology cluster.

News	Date	Sentiment score
Macro Positive		
Central Banks to Hedge Dollar Risks with Bitcoin, Pompliano Predicts.	2019-08-01	0.26
US Federal Reserve Hiring New Manager to Research Digital Currencies.	2019-11-05	0.09
Bitcoin Simply Existing Positively Impacts Monetary Policy: Research.	2019-08-19	0.38
Macro Negative		
Israel’s Central Bank ‘Not Recommended’ to Issue Own Digital Currency.	2018-11-07	-0.15
US Federal Reserve Has No Plans to Introduce Digital Currencies, Says San Francisco Fed President.	2017-12-04	-0.13
US State of Ohio Suspends Service for Paying Taxes With Bitcoin.	2019-10-02	-0.12

Table E.3 – Examples of positive and negative news in Macro cluster.

News	Date	Sentiment score
	Regulatory Positive	
ICOs Can ‘Prove Their Legitimacy’ Under New Crowdfunding Rules, Says EU Lawmaker.	2018-08-14	0.16
Major Lobbyists in DC See a Shift in Regulatory Tone Towards Crypto Since Pandemic.	2020-07-16	0.11
House passes digital asset innovation act to clarify crypto regulations.	2021-04-21	0.13
	Regulatory Negative	
Bitcoin Exchange Bitfinex Exits Washington State In 24 Hours, Licence Problems Cited.	2017-03-01	-0.06
Ukraine: Overregulation Prevents Crypto Development, Says Central Bank Official.	2019-01-09	-0.15
Crypto-friendly trading app Robinhood faces lawsuit from securities regulators.	2020-12-16	-0.13

Table E.4 – Examples of positive and negative news in Regulation cluster.

News	Date	Sentiment score
	Past Prices Positive	
Bitcoin Price Soars Over 5% to Open November; Is Segregated Witness Cause?	2016-11-02	0.10
Bitcoin Price Tackles $12,000 After Breaking Through a Key Resistance Zone.	2020-08-10	0.13
Bitcoin Hits Multi-Week Highs Despite Continuing Altcoin Surge.	2018-01-05	0.08
	Past Prices Negative	
Bitcoin Drops Sharply to Below $3,900, Total Market Cap Sheds $15 Billion.	2019-02-24	-0.25
This unknown cryptocurrency soared by 164,842% in hours, only to crash 99%.	2021-06-15	-0.16
Tuesday Shows Bloodbath for Altcoins, Up to 34% Losses on Top 20 Coins.	2019-09-24	-0.18

Table E.5 – Examples of positive and negative news in Past Prices cluster.

News	Date	Sentiment score
Price Predictions Positive		
3 reasons why Ethereum price is still on track to top $2,000.	2021-01-25	0.25
Most Traders Expect New All-Time High Price for Bitcoin in 2020.	2020-03-28	0.23
Investment Firm Wedbush Predicts $400 Bitcoin by 2016; Advises to Buy GBTC.	2015-07-15	0.19
Price Predictions Negative		
Bitcoin Price Drop to $2.5K Possible if Bulls Fail to Retake $7,350.	2019-11-24	-0.17
Shock survey suggests most investors think Bitcoin won't top $50K by 2030.	2020-12-15	-0.11
Former Facebook Exec: BTC's Price is Either Going to Zero or Seven Figures.	2020-04-07	-0.13

Table E.6 – Examples of positive and negative news in Prices Predictions cluster.

F Appendix: Robustness check

In this section I provide the results of the robustness check performed in section 3.5.

	r_t	
	(1)	(2)
const	0.066^{***}	0.066^{***}
	(0.018)	(0.022)
Past Prices	0.091^{***}	
	(0.019)	
Price Predictions	0.050^{**}	
	(0.024)	
Adoption	0.057^{***}	0.046^{**}
	(0.016)	(0.020)
Macro	0.028	0.054^{**}
	(0.024)	(0.027)
Regulatory	0.038^{*}	0.047^{***}
	(0.022)	(0.016)
Tech	0.004	0.047^{**}
	(0.020)	(0.022)
Observations	75	75
R^2	0.451	0.204
Adjusted R^2	0.402	0.158
Residual Std. Error	0.163(df = 68)	0.193(df = 70)
F Statistic	17.124^{***} (df = 6.0; 68.0)	7.885^{***} (df = 4.0; 70.0)
Note:		*p<0.1; **p<0.05; ***p<0.01

Table F.1 – Robustness check: returns explained by news allowing for double attribution

This table reports the results of the two models, each analyzing the explanatory power of the AR(1) sentiment innovation for the named groups(s) of news. Articles with low confidence scores are attributed to two best-fitting groups at the same time. The dependent variable r_t is monthly Bitcoin return. Coefficients correspond to the effect of one standard deviation increase in the sentiment innovation. Standard errors are in parenthesis. I use Newey-West standard errors, robust to heteroskedasticity and autocorrelation up to 1 lag.

	r_t			
	(1)	(2)	(3)	(4)
const	0.066^{***}	0.066^{***}	0.066^{**}	0.066^{***}
	(0.024)	(0.025)	(0.026)	(0.025)
Adoption	0.055^{**}			
	(0.022)			
Macro		0.045^{*}		
		(0.025)		
Regulatory			0.031	
			(0.020)	
Tech				0.043^{**}
				(0.021)
Observations	75	75	75	75
R^2	0.068	0.046	0.021	0.042
Adjusted R^2	0.055	0.033	0.008	0.028
Residual Std. Error	0.205(df = 73)	0.207(df = 73)	0.210(df = 73)	0.208(df = 73)
F Statistic	6.437^{**}	3.296^{*}	2.430	4.111^{**}
Note:			*p<0.1; **p<0.05; ***p<0.01	

Table F.2 – Robustness check: returns explained by each group of news allowing for double attribution

This table reports the results of the four models, each analyzing the explanatory power of the sentiment innovation for the named exogenous group of news. Articles with low confidence scores are attributed to two best-fitting groups at the same time. The dependent variable r_t is monthly Bitcoin return. Coefficients correspond to the effect of one standard deviation increase in the sentiment innovation. Standard errors are in parenthesis. I use Newey-West standard errors, robust to heteroskedasticity and autocorrelation up to 1 lag.

	r_t
	(1)
$\hat{S}^{fund}_{t-1}$	0.001537*
	(0.000843)
$\hat{S}^{fund}_{t-2}$	0.001960**
	(0.000841)
$\hat{S}^{fund}_{t-3}$	-0.000117
	(0.000844)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table F.3 – Predicting one day ahead Bitcoin returns with innovation in fundamental news sentiment.

This table reports the coefficients of lagged sentiment AR(1) innovation in the VAR return regression (3.8), allowing for double cluster news attribution. Coefficients correspond to the effect of one standard deviation increase in the fundamental news sentiment innovation. Standard errors are in parenthesis.

	Vlm_t
	(1)
$\|\hat{S}^{fund}_{t-1}\|$	0.018488
	(0.012548)
$\|\hat{S}^{fund}_{t-2}\|$	0.021898*
	(0.012621)
$\|\hat{S}^{fund}_{t-3}\|$	-0.001917
	(0.012531)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table F.4 – Predicting one day ahead Bitcoin trading volume with innovation fundamental news sentiment.

This table reports the coefficients of the absolute value of the lagged sentiment AR(1) innovation in the VAR return regression (3.9), allowing for double cluster news attribution. Coefficients correspond to the effect of one standard deviation increase in regressors. Standard errors are in parenthesis.

Bibliography

Ahmad, K., Han, J. G., Hutson, E., Kearney, C., and Liu, S. (2016). Media-expressed negative tone and firm-level stock returns. *Journal of Corporate Finance*, 37:152–172.

Aït-Sahalia, Y. (2004). Disentangling diffusion from jumps. *Journal of Financial Economics*, 74(3):487–528.

Andersen, T. G., Bollerslev, T., and Diebold, F. X. (2007a). Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility. *Review of Economics and Statistics*, 89(4):701–720.

Andersen, T. G., Bollerslev, T., and Dobrev, D. (2007b). No-arbitrage semi-martingale restrictions for continuous-time volatility models subject to leverage effects, jumps and i.i.d. noise: Theory and testable distributional implications. *Journal of Econometrics*, 138(1):125–180.

Andersen, T. G., Dobrev, D., and Schaumburg, E. (2009). Jump-robust volatility estimation using nearest neighbor truncation.

Au Yeung, J. F., kai Wei, Z., Chan, K. Y., Lau, H. Y., and Yiu, K. F. C. (2019). Jump detection in financial time series using machine learning algorithms. *Soft Computing*.

Barndorff-Nielsen, O. E. and Shephard, N. (2004). Power and Bipower Variation with Stochastic Volatility and Jumps. *Journal of Financial Econometrics*, 2(1):1–37.

Barndorff-Nielsen, O. E. and Shephard, N. (2006). Econometrics of testing for jumps in financial economics using bipower variation. *Journal of Financial Econometrics*, 4(1):1–30.

Barndorff-Nielsen, O. E., Shephard, N., and Winkel, M. (2006). Limit theorems for multipower variation in the presence of jumps. *Stochastic Processes and their Applications*, 116(5):796–806.

Boudt, K., Croux, C., and Laurent, S. (2011). Robust estimation of intraweek periodicity in volatility and jump detection. *Journal of Empirical Finance*, 18(2):353–367.

Bybee, L., Kelly, B. T., Manela, A., and Xiu, D. (2019). The Structure of Economic News. *SSRN Electronic Journal*.

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

Chen, C. Y., Despres, R., Guo, L., and Renault, T. (2019). What Makes Cryptocurrencies Special? Investor Sentiment and Return Predictability During the Bubble. *SSRN Electronic Journal*.

Christensen, K., Oomen, R. C., and Podolskij, M. (2014). Fact or friction: Jumps at ultra high frequency. *Journal of Financial Economics*, 114(3):576–599.

Cont, R. and Tankov, P. (2004). *Financial Modelling With Jump Processes*. Chapman & Hall/CRC Financial Mathematics Series.

Corbet, S., Larkin, C., Lucey, B. M., Meegan, A., and Yarovaya, L. (2020). The impact of macroeconomic news on Bitcoin returns. *European Journal of Finance*, 2020(14):1396–1416.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196.

Corsi, F., Pirino, D., and Renò, R. (2010). Threshold bipower variation and the impact of jumps on volatility forecasting. *Journal of Econometrics*, 159(2):276–288.

Delbaen, F. and Shirakawa, H. (2002). No arbitrage condition for positive diffusion price processes. *Asia-Pacific Financial Markets*, 9(3-4):159–168.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186.

Dumas, B. and Luciano, E. (2017). *The Economics of Continuous-Time Finance*. MIT Press Books, edition 1 edition.

Dumitru, A. M. and Urga, G. (2012). Identifying jumps in financial assets: A comparison between nonparametric jump tests. *Journal of Business and Economic Statistics*, 30(2):242–255.

Frazier, K. B., Ingram, R. W., and Tennyson, B. M. (1984). A Methodology for the Analysis of Narrative Accounting Disclosures. *Source: Journal of Accounting Research*, 22(1):318–331.

García, D. (2013). Sentiment during Recessions. *Journal of Finance*, 68(3):1267–1300.

Genon-Catalot, V. and Jacod, J. (1993). On the estimation of the diffusion coefficient for multidimensional diffusion processes. *Annales de l'I.H.P. Probabilités et statistiques*, 29(1):119–151.

Glasserman, P., Li, F., and Mamaysky, H. (2019). Time Variation in the News-Returns Relationship. *SSRN Electronic Journal*.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). LSTM: A Search Space Odyssey. *Transactions on Neural Networks and Learning Systems.*

Gu, S., Kelly, B., and Xiu, D. (2019). Empirical Asset Pricing via Machine Learning.

Guo, L., Shi, F., and Tu, J. (2016). Textual analysis and machine leaning: Crack unstructured data in finance and accounting. *Journal of Finance and Data Science*, 2(3):153–170.

Hafez, P. (2009). Construction of market sentiment indices using news sentiment. *RavenPack International SL*, 1(646).

Han, W., Newton, D., Platanakis, E., Sutcliffe, C. M., and Ye, X. (2021). Cryptocurrency Factor Portfolios: Performance, Decomposition and Pricing Models. *SSRN Electronic Journal.*

Hazlett, P. K. and Luther, W. J. (2020). Is bitcoin money? And what that means. *Quarterly Review of Economics and Finance*, 77:144–149.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hugonnier, J. (2012). Rational asset pricing bubbles and portfolio constraints. *Journal of Economic Theory*, 147(6):2260–2302.

Jarrow, R., Kchia, Y., and Protter, P. (2011). How to detect an asset bubble. *SIAM Journal on Financial Mathematics*, 2(1):839–865.

Jarrow, R. and Protter, P. (2012). Discrete versus continuous time models: Local martingales and singular processes in asset pricing theory. *Finance Research Letters*, 9(2):58–62.

Jarrow, R. A. (2015). Asset Price Bubbles. *Annual Review of Financial Economics*, 7(1):201–218.

Jarrow, R. A., Protter, P., and Shimbo, K. (2007). *Asset price bubbles in complete markets.*

Jarrow, R. A., Protter, P., and Shimbo, K. (2010). Asset price bubbles in incomplete markets. *Mathematical Finance*, 20(2):145–185.

Jha, M., Liu, H., and Manela, A. (2020). Does Finance Benefit Society? A Language Embedding Approach. *SSRN Electronic Journal*, (October).

Jiang, G. J. and Oomen, R. C. A. (2007). Testing for Jumps When Asset Prices are Observed with Noise - A “Swap Variance” Approach.

Kantorovitch, I. and Heineken, J. (2021). Does Dispersed Sentiment Drive Returns, Turnover, and Volatility for Bitcoin? *SSRN Electronic Journal*, pages 1–42.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., and Shin, Y. (1992). 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(1-3):159–178.

Lee, S. S. and Mykland, P. A. (2008). Jumps in financial markets: A new nonparametric test and jump dynamics. *Review of Financial Studies*, 21(6):2535–2563.

Leung, H. and Ton, T. (2015). The impact of internet stock message boards on cross-sectional returns of small-capitalization stocks. *Journal of Banking and Finance*, 55:37–55.

Liebi, L. J. (2022). Is there a value premium in cryptoasset markets? *Economic Modelling*, 109:105777.

Liu, Y., Tsyvinski, A., and Wu, X. (2019). Common Risk Factors in Cryptocurrency. *SSRN Electronic Journal*.

Lo, A. (2004). The adaptive market hypothesis: market efficiency from an evolutionary perspective. *The Journal of Portfolio Management*, 30(5):15–29.

López-Martín, C., Sonia, ·., Muela, B., Arguedas, R., and Muela, S. B. (2021). Efficiency in cryptocurrency markets: new evidence. *Eurasian Economic Review*, 11:403–431.

Loughran, T. and Mcdonald, B. (2016). Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research*, 54(4):1187–1230.

Loughran, T. and McDonald, B. (2020). Textual Analysis in Finance. *SSRN Electronic Journal*.

Lyócsa, Š., Molnár, P., Plíhal, T., and Širaňová, M. (2020). Impact of macroeconomic news, regulation and hacking exchange markets on the volatility of bitcoin. *Journal of Economic Dynamics and Control*, 119:103980.

Mäkinen, M., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2018). Forecasting of Jump Arrivals in Stock Prices: New Attention-based Network Architecture using Limit Order Book Data.

Manela, A. and Moreira, A. (2017). News implied volatility and disaster concerns. *Journal of Financial Economics*, 123(1):137–162.

Marty, T., Vanstone, B., and Hahn, T. (2020). News media analytics in finance: a survey. *Accounting and Finance*, 60(2):1385–1434.

Mijatović, A. and Urusov, M. (2012). On the martingale property of certain local martingales. *Probability Theory and Related Fields*, 152(1-2):1–30.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Mukherjee, A., Peng, W., Swanson, N. R., and Yang, X. (2019). Financial econometrics and big data: A survey of volatility estimators and tests for the presence of jumps and co-jumps.

Obayashi, Y., Protter, P., and Yang, S. (2017). The lifetime of a financial bubble. *Mathematics and Financial Economics*, 11(1):45–62.

Podolskij, M. and Ziggel, D. (2010). New tests for jumps in semimartingale models. *Statistical Inference for Stochastic Processes*, 13(1):15–41.

Protter, P. (2001). A partial introduction to financial asset pricing theory. *Stochastic Processes and their Applications*, 91(2):169–203.

Protter, P. (2013). A mathematical theory of financial bubbles. *Paris-Princeton Lectures on Mathematical Finance*, pages 1–108.

Reimers, N. and Gurevych, I. (2020). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3982–3992.

Roll, R. (1988). R2. *The Journal of Finance*, 43(3):541–566.

Smales, L. A. (2014). Non-scheduled news arrival and high-frequency stock market dynamics. Evidence from the Australian Securities Exchange. *Research in International Business and Finance*, 32:122–138.

Tauchen, G. and Todorov, V. (2008). Volatility Jumps.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62(3):1139–1168.

Theodosiou, M. G. and Zikes, F. (2012). A Comprehensive Comparison of Nonparametric Tests for Jumps in Asset Prices. *SSRN Electronic Journal*, 44(0).

Yermack, D. (2015). Is Bitcoin a Real Currency? An Economic Appraisal. In *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*, pages 31–43. Academic Press.