Network analysis and data science for finance: from traditional markets to decentralised exchanges

Deborah Miori

St Hugh's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2024

Acknowledgements

This thesis collects hard work from four years of my life. And since it spans such a long and meaningful period of time, this manuscript is also almost a diary to me. When reading it in the future, I will always retrieve feelings of excitement, stress, relief, and deep personal growth, that accompanied the time I spent on each one of my thesis' chapters. There are several people that have been part of this adventure, and I want to thank them now because they made this journey truly special.

Clearly, nothing would have been possible without the support of Mihai, my main supervisor. He took me under his wing after a first rough year, and gave me constant inputs and support ever since. I believe we were also a good match "schedule-wise", since we would often together span all the 24 hours of a day in work, due to our peculiar sleeping patterns. I then need to express my deep gratitude to my supervisor Rama. He gave me invaluable guidance at every important step of my PhD, and taught me to always meditate on the bigger picture both in research and professional life. Finally, my mentor Tino initiated me to real financial markets, and always reminded me that doing things differently and with imagination is valuable despite not always appreciated. I owe to him a lot of the self-confidence I gained these years.

During these four years, I have found fantastic people at the Mathematical Institute, with whom I shared many coffee breaks, evenings at the pubs, or Formal dinners at each others' colleges. Among them, I want to thank my two office mates Laszlo and Andrea for their never-ending energy, and Marcello for his advice (and constant jokes too). Then, I want to thank Milena (my make-up guru and favourite wine-buddy for well-deserved fun evenings), Felix, Irene and Giada, for having been great friends all the way along this journey. To do things correctly, I should also spend three or four pages describing why I am grateful to my family. But since the reader would skip that, I will keep it short. I thank my mum Cinzia, dad Attilio, grandmum Eliana (and dog Ronphy) for being who they are, and for being my family. They are always there for me, whatever the circumstances, the time of the day or night I might be calling, or the senseless speeches I sometimes really do. I am truly lucky to have them.

Abstract

Research on financial markets often confines itself to in-depth analyses of time series of asset prices, despite we are now in an era of unprecedented wealth of data that offers boundless opportunities for wider investigations. This thesis aims at broadening our understanding of traditional and decentralised market ecosystems, by taking indeed advantage of "unconventional data". The latter are labelled as such either for their origin (i.e. being alternative data), or for their extensiveness (e.g. spanning multiple asset classes). Given the inherent higher complexity of our data, we leverage data science advancements to analyse them thoroughly. Recurrent techniques employed in this work include network science for capturing relationships among entities of interest, and clustering methods for dimensionality reduction and aggregation of information. Within traditional finance ecosystems, we investigate three sources of possible novel market insights, which indeed lead to alternative risk-monitoring tools. The first source lies in institutional investors' holdings, which are found to signal crowding in trades, after aggregating the bipartite network of funds and their assets. Then, we consider a corpus of economic news with available timestamps. By modelling and clustering the interlinkage of concepts discussed within such news, we discover the major narratives of interest over time and map entropy in their state to market dislocations. Otherwise, we study returns of an heterogeneous set of indices belonging to multiple asset classes, and characterise their network of evolving correlations to identify market regimes that are found to have distinguishable macroeconomic features. Within decentralised finance ecosystems, we instead take direct advantage of the extensive and meticulous data-recording of blockchains. The trading activity of agents on multiple tokens is used to construct a network of transactions for each one of them, and clustering the set of such graphs allows us to identify interpretable "species" of traders. Lastly, we analyse data on liquidity provision, consumption, and price formation on competing decentralised exchange venues, to find a model for the prediction of incoming trading volume at block-level.

Contents

Chapter 1

Preface

1.1 Motivation

While the origins of finance are likely to date back to the start of civilisation, well-known exchanges such as the London Stock Exchange or the New York Stock Exchange were only founded in the second half of the 18th century. This past decade has then witnessed the creation of novel blockchain-based decentralised exchanges such as Uniswap or Sushiswap, which are venues that trade assets without centralised permission. With time, increasingly sophisticated technological advancements have also appeared and greatly simplified the tasks of data collection and processing across disciplines. Within finance, we are now able to access extensive data sets of e.g. institutions' holdings, companies' annual reports, consumer transactions, economic and financial news..., beyond having higher resolution of records on assets traded in markets. Blockchains are then the end stage of this process, since they effectively are ledgers that store every user transaction permanently and fully publicly.

Despite the innumerable novel research directions opened by such wealth of data, present-day financial studies still tend to revolve around the mere examination of time series of limited samples of asset prices. Alternative data remain largely under-investigated, and this reality urged us to systematically study a selection of related sets within this thesis, to consequently advocate for their significant endowment of information. We leverage modern data science methods, and especially network analysis and clustering techniques, to address the inherent complexity of such data and deliver novel insights. Our studies contribute to advancing the current understanding of both traditional (TradFi) and decentralised (DeFi) finance ecosystems, from a strongly data-driven point of view.

1.2 Contributions

Throughout this thesis, "alternative data" refers to non-traditional sources of information within financial research, from which we can extract precious insights into markets, companies, and economic trends. Despite the novel wealth of such data, allowed by the advancements within data collection and processing technologies, we believe that an inadequate effort has been put to simultaneously expand the frontiers of financial research and econometrics. We advocate an enhanced focus on such data, which we personally pursue and propose here to demonstrate its downstream benefits. Indeed, our research models and characterises the intrinsic relationships inherent to multiple alternative data sets (e.g. overlapping investments among funds, equivalent trading patterns of agents...), and unravels unique insights within both traditional and decentralised finance ecosystems. More specifically, we often deal with the heterogeneity and complexity of those data by taking advantage of network analysis techniques to model inter-linkages among entities of interest, and clustering methods to reduce the dimensionality of results. However, care is also taken in accounting for the temporal component of the data and investigating the joint evolution of our systems' features.

Chapter 2 proposes an explicative study that considers a set of institutional investment managers and the assets that they hold over time, which directly points to a bipartite graph construct. We then aggregate the trading behaviour of this set of highly heterogeneous US institutional investors (i.e. investment advisers, banks, insurance companies...) over assets, by looking at the variation of their holdings as disclosed in their SEC 13F Form at the end of each quarter. Consequently, we define a measure of imbalance in the buying or selling pressure by such institutions on individual securities, which results in a reliable signal of crowding and over-heating of trades. Crowding is the tendency of different investors to focus on a similar set of factors, strategies, or securities, which increases the risk of liquidity-driven tail events and must be consequently carefully monitored. Related studies are [97], [127], [29], [85], [25]. It is thus positive to see that our simple construct immediately finds evidence of evolving crowding in trades, clearly showing the benefit of analysing data beyond just asset prices. Interestingly, the mentioned diverse classes of US institutions have also been studied in the literature. However, they have been mainly considered on their own, and examples of such investigations are [57], [67], [9] for corporate and Treasury bond funds, [56] and [51] for insurance companies, and [75], [16], [62] for hedge funds. Alternatively, a few research papers such as [21] and [22] do consider the interconnections between asset returns of hedge funds, banks, brokers and insurance

companies, but mainly to identify financial crisis periods. *Note: this Chapter is based on the paper [93] co-authored with Prof. Mihai Cucuringu, published in the Journal of Investment Strategies, from Risk Journals.*

Chapter 3 focuses on studying a well-known type of alternative data, which is news. There is abundant literature proposing to compute statistics over the number and sentiment of news, to gain insights into the valuation of related stocks. However, we take a strongly different approach and model the interconnectedness of news, from which we extract topics characteristic of different points in time and try to identify the related evolving narratives. We focus on a corpus of economic articles from The Wall Street Journal, and introduce an effective network-based framework for systematic but interpretable detection of topics and narratives. "State-of-the-art" topic modelling techniques, namely Latent Dirichlet Allocation (LDA) [124], Top2Vec [7], and BERTopic [60], achieve very questionable results when applied to diverse real-world corpora of text. Indeed, the research in [41] also highlights such point, while further investigating topic modeling in the context of finance-related news impact analysis. In our case, we leverage the GPT3.5 model to extract the principal entities of each article, and encode their co-occurrence among news in weekly graphs. This allows us to define a construct for an holistic view of news space, on which quantitative metrics can be directly computed. We thus create a set of features that characterise the type and structure of news within each week, and map them to moments of financial markets dislocations (i.e. weeks of high volatility across asset classes). Evidence is found that critical market moments are associated to instances of high entropy in the high-dimensional space of interconnected news. A sample of related, despite distant, investigations are [110], [63], and [121]. These studies tend indeed to focus on one-to-one relationships between stocks in the equity market and associated news, chats, or reports, thus lacking any global inference on the landscape of broader matters of concern. On the other hand, [80] investigates how news affect the trading behaviour of different categories of investors in a financial market, and [123] investigates the high-frequency interdependent relationships between the stock market and simple statistics on US economic news. *Note: this Chapter is based on the paper [96] co-authored with Dr. Constantin Petrov, from Fidelity Investments Inc., FMR. The paper is published in the International Journal of Data Science and Analytics.*

Chapter 4 proceeds with an investigation of financial market regimes. Despite we do use data on asset prices here, our set of interest can be labelled as "unconventional" due to its breadth. Indeed, we take an holistic view of markets and consider more than 200 time series of assets belonging to all equity, bonds, FX, commodities...,

asset classes. In this way, we can fully leverage network constructs and clustering methodologies, to better investigate the temporal evolution of relationships among such time series, and consequently characterise the ecosystem. The importance of regime identification lies in its impact on asset allocation, portfolio construction, and risk control, as highlighted in [6]. A review of associated research is available in [88], while the importance of correlation studies of stock returns during periods of market distress is highlighted in [108]. Well-known studies, such as [24] and [135], tend to investigate regimes by clustering correlation matrices of equity returns, or solving community detection on associated minimum spanning trees to enhance portfolio diversification over periods of indeed different market conditions. However, this area of research lacks indeed studies on correlations and causalities among returns for securities belonging to *multiple* asset classes, which we consequently complete and use to identify interpretable regimes over time. We also propose a stability measure for the transition between such regimes, from the clustering of a signed graph based on instruments' evolving correlations. The results can become part of a risk alert system for the need of portfolios rebalancing, or used to inform on recurrent market developments. *Note: this Chapter is based on the paper [92] co-authored with Prof. Mihai Cucuringu, presented at a ICAIF 2022 workshop. It was completed as part of a collaboration with Fidelity Investments Inc., FMR.*

Chapter 5 builds upon our confidence on the power of alternative (and "broader") data, and studies which achievements can be obtained by leveraging on the accurate, complete, and transparent recording of users' transactions on blockchains. By proposing a graph construct to model each trader's transactions on multiple decentralised trading venues, we assess their similarity in behaviour, and find clusters of interpretable species. This is a first step into a better empirical understanding of the dynamics within decentralised finance ecosystems. To achieve the above result, we first propose a systematic workflow to extract a tractable sub-universe of liquidity pools, where the interconnection among such pools is maximised to capture broader dynamics within the ecosystem. Then, we introduce the mentioned graph representation per market participant, and embed such graphs into a vector space via an extension of the graph2vec algorithm of [101] that we propose. Finally, we cluster the results and identify different "species" of agents. Several studies have recently focused on the dynamics of liquidity consumption and provision within decentralised exchanges, often from an optimisation perspective (see e.g. [35], [36], [43], [15]) or a market design point of view (see e.g. [52], [54], [18], [44]). However, we are the first to propose such an impartial approach to cluster traders by simply considering the temporal component

of their executed transactions, and allowing for multiple target pools. Similarly, only few research efforts study related questions on traditional Limit Order Book data, due to problems of accessibility and confidentiality of such data. A couple of interesting examples are [45], [34], and [116], in which the authors extract and analyse clusters of agents from order flow analyses, either thanks to data from a large broker in US equity markets or from Euronext Amsterdam. *Note: this Chapter is based on the paper [95] co-authored with Prof. Mihai Cucuringu, published in Digital Finance (Springer Journals).*

Chapter 6 further leverages on the level of market transparency provided by blockchains, and shows how a simple regression model achieves satisfactory results when predicting incoming volume on a decentralised exchange. Indeed, by constructing features that span from the latest activity on the pair of traded tokens of interest, but also computed from spillover effects either from other traded pairs or from Centralised Exchanges (CEXs) such as Binance, we are able to take a wide enough view that makes the regression successful. This research relates to the studies of [20], [28], [40], [125], which consider liquidity prediction in traditional finance. Interestingly, literature is scarce on the topic, despite volume being a key variable in many financial and economic theories, as well as a practical indicator of movements of prices, slippage, and overall market activities. *Note: this Chapter is based on the paper [94] co-authored with Prof. Mihai Cucuringu, available in the post-proceedings of ChainScience 2023.*

This thesis dives into multiple alternative data sets, to show the unique insights that we can find by modelling and investigating unconventional relationships within elements of interest. With a completely data-driven approach, we study the important phenomena of crowding, narrative influence, and regime shifting in traditional financial markets. Similarly, we cluster traders on decentralised exchanges and model incoming liquidity consumption on such venues, by taking advantage of the excellent breadth and transparency of blockchain data. Thanks to the research completed, we believe to have enhanced our comprehension of both centralised and decentralised ecosystems[1].

Parallelisms to traditional research. For the sake of clarity, we stress here a couple of key points that highlight how our approaches are complementary to traditional methods within financial research. The first clear example is our crowing analysis, which leverages an alternative source of market positions belonging to an eclectic set of investors, and diverges from e.g. martingale-based approaches that

[1]For ease of reproducibility of our results, we shared core steps of our code at `https://github.com/debbih/` within the "phd-ch*.git" repositories.

investigate the relation between trading signals and order flow in crowded markets, or that indeed define financial bubbles as local martingales. Similarly, financial regimes have often been investigated via Hidden Markov Models, or by focusing on a strictly "bull versus bear" dichotomic characterisation. We prefer not to impose any prior ideology on the regimes to search for, and thus deeply dive into what data suggest by testing the asset-driven similarity of market behaviour during different moments in time. On top of that, we also divert from the available literature regarding the characterisation and clustering of market investors. Instead of building ad-hoc measures to depict the essential traits of liquidity takers active in a market, and cluster them according to those, we propose a complementary view that focuses on the sequential structure of their full trading behaviour (as recorded on blockchains). Clearly, our methodologies try to approach well-known problems in finance from alternative points of view, and should thus be considered as additional data-driven tools (not substitutes) to enhance the pre-existent knowledge already established in the field.

Chapter 2

Overlapping Behaviour of Institutions and Crowding in Trades

We begin with an investigation of a comprehensive data set of institutional holdings, namely a collection of SEC Form 13F-HR filings. Any US Institution with more than $100 million assets under management must indeed disclose its long positions into the SEC Form 13F-HR on a quarterly basis. We consider the number of variations in such institutions' holdings between consecutive reporting periods, and compute a normalised measure of the discrepancy (*imbalance*) in the volume of shares bought or sold for each asset. In this way, we quantify the level of coherence in the trading behaviour of our eclectic set of market participants, on each available security. While the results could point to the "best trades" to invest into, we hypothesise that agreement among such a breadth of institutions signals crowding in the market and riskier over-heated trades. To prove the statement, we divide assets into quantiles according to the strength of the associated imbalance, and trade them in opposite direction to the sign of such imbalances. We test the hypothetical tuning of this fictitious strategy, compute the related Sharpe ratios, and compare results to a benchmark that follows a basic price mean-reversion strategy. A significant opportunity for profit is shown to arise, if an external investor is willing to trade contrary to the strongest 13F filings imbalances. Thus, imbalances capture the amount of information already consumed in the market and the related trades tend to be inflated by crowding and herding behaviours. The latter usually unwind between 21 and 42 trading days (i.e. 1-2 calendar months) after the end of each financial quarter, implying that such time window must be monitored for risk control.

2.1 Introduction

Diversification is a double-hedged sword in financial markets. It helps to offset risks in quiet market times but significantly increases the likelihood of being affected by financial contagion during periods of sudden stress [32]. After the financial crisis of 2007-08, regulators strengthened their control on systemically important institutions [126] and promoted higher transparency of markets by increasing requirements on disclosures of holdings. The behaviour and performance of banks and non-bank financial institutions (such as mutual funds, pension funds, insurance companies, hedge funds, etc.) has become an active area of research since then. Examples are the investigations of corporate and Treasury bond funds in [57], [67], [9], of insurance companies in [56], [51], and of hedge funds in [75], [16] and [62]. Interconnections between the returns of hedge funds, banks, brokers and insurance companies are further analysed in [21] and [22], for the US and European markets respectively. Many studies (e.g. [97], [127], [29], [85], [25]) also address the importance of studying crowding, which is the tendency of different investors to focus on a similar set of factors, strategies, or securities. This happens both inside and across institutions' categories, and results in overlapping positions with generally lower returns but for hedge funds.

The information contained in the Security and Exchanges Commission (SEC) Form 13F-HR can be of interest to any study spanning diverse agents' classes, since this filing publicly provides updates on the holdings of the largest US institutional investors on a quarterly basis. Despite being released with a possible (maximum) lag of 45 calendar days after the financial quarter ends and reporting only long positions, Form 13F allows insights on an extremely eclectic set of market participants and has been published with satisfactory quality for at least a decade. Few instances of research have been focusing on it so far, likely due to difficulties in systematically accessing and pre-processing these reports, which are singularly stored on a SEC platform and appear at different times. An example is [130], which leverages these filings to forecast stock returns in mutual funds by "trusting" managers of companies that demonstrated a track record of profitability in the past. Otherwise, [26] shows that 13F filings are strongly used by plain copycat investors even if this has little evidence of long-term benefits. The research in [42] relates SEC Form 13F to SEC Form 10K, which must be filed annually by companies with registered shares and provides a comprehensive summary of their financial performance. As conceivable, the latter author finds that pessimism in Form 10K leads to a decrease in related institutional holdings. Our study

considers a different approach from the above-mentioned ones, and employs Form 13F data to aggregate changes in investors' positions, thus computing a proxy for the flow of money in and out of different assets.

Main contributions. We define a measure of trading imbalance for stocks, which weights the strength of related buying versus selling behaviour between consecutive reporting periods. In this way, we are able to investigate the global view that our diverse set of institutional investors develops on each stock. Our main contribution lies in the extraction of a statistically significant signal from the imbalances computed via 13F filings, which suggests that asset prices move in the opposite direction to their imbalance sign at the end of each quarter. Consequently, our results alert for institutional crowding and can signal over-heating of trades.

Structure of the Chapter. Section 2.2 describes in detail the data used and the pre-processing needed. Afterwards, we highlight the methodology followed to compute imbalances and trading signals in Section 2.3. Preliminary considerations on imbalances and their price impact on contemporaneous returns are then presented in Section 2.4. In Section 2.5, we show our profits when betting contrary to imbalances and describe experiments of conditioning positions to stocks' past prices or to the sector membership of each asset. We conclude our work in Section 2.6, by highlighting the connection between imbalances and crowding, and propose some further remarks.

2.2 Data

2.2.1 Form 13F-HR

The SEC introduced Form 13F-HR ("Holdings Report") to provide some publicly available disclosure of institutions' investments back in 1975. Following its updated rules, the related managers need to report long holdings via the 13F filings when handling US assets under management (AUM) $\geq$ \$100 million. Institutional investment managers can be part of investment advisers, banks, insurance companies, broker-dealers, pension funds, corporations, and so on. Therefore, non-financial institutions are also included in this form. However, 13F disclosures are required only quarterly, filed in number of shares, miss the short positions and must be completed within 45 days (i.e. one month and a half) after the financial quarter ends. Institutions are also allowed to file amendments (Form 13F-HR/A) if they mistakenly reported a position, but without incurring into any consequent fees or alerts. Thus, the most valuable

bets are often kept hidden for a longer time, and deliberately disclosed with a delay, as [33] shows for hedge funds.

Every market participant filing the SEC Form 13F is identified by a Central Index Key (CIK), and investments pursued by different managers are discerned by the "Other Manager" integer number. The division between actual portfolios would be in general described by the SeriesID value, but this is unfortunately not required in 13F filings, meaning that we cannot access this most granular view. Assets that are compulsory to report are part of the SEC Section 13(f) securities list, which introduces a bias towards US funds and investments. The full list of securities covered includes $23,131$ securities as of Q4 2021 (fourth quarter of 2021). These are identified by the US CUSIP number (Committee on Uniform Securities Identification Procedures) and listed on their website[1]. Therein, one can find equity securities that trade on an exchange, certain equity options and warrants, shares of closed-end investment companies, and some convertible debt securities. The shares of open-end investment companies are instead not required.

US CUSIP codes are assigned with format "AAAAAABBC", where "AAAAAA" represents the company issuing the financial security, "BB" is the issue number used to denote e.g. shares with or without voting rights, and "C" is the check digit. The informative part for our purposes is the "AAAAAA", which we denote as CUSIP6. As an example, the AAPL stock has code "037833100", where "037833" is Apple, Inc. Then, "10" denotes Class A Shares and the final "0" is the check digit. It is a convention to assign "10" as first issue number and increasing numbers ("20","30"... or "11","12"...) for subsequent securities.

2.2.2 Data pre-processing

We consider holdings disclosed via Form 13F-HR from 2013-06-30 to 2021-09-30, where earlier quarters are not included due to low quality of reporting. We also do not correct positions for which an amendment was later filed, in order to investigate the information theoretically available at the end of each quarter. However, we do assume that 13F filings are immediately available at the end of each quarter, i.e. we ignore the possible lag (up to 45 calendar days) that companies are allowed in their holdings' disclosure deadline. This implies that related signals are not fully actionable, but we remark that they are still representative of the market exposures filed for the end of each quarter and so useful for the study of crowding. We check the average delay

[1] `https://www.sec.gov/divisions/investment/13flists.htm`

in the filing of reports for completeness, and see that it amounts to ~ 35 calendar days (i.e. ~ 25 trading days) for the most recent quarters of data, but with high standard deviation. Thus, our analyses could be extended to a dynamic setting (i.e. incorporating novel forms as they are filed) to relax the mentioned assumption.

Overall, our 13F dataset shows long investments for $T = 34$ points in time. For each quarter, the number of samples is larger than one million (i.e. $> 10^6$), where each occurrence represents an agent holding a specific security of type "shares" at that point in time. The "Other Manager" key is discarded, while the related number of shares held is summed for each individual institution and security, for every period. Similarly, we aggregate stocks information to CUSIP6 level. The full data show $8,166$ unique CIKs, among which we see influential and diverse institutions such as banks (e.g. Bank of America, Wells Fargo & Co), insurance companies (e.g. American International Group, AXA), pension funds (e.g. Canada Pension Plan Investment Board, Public sector pension investment board), hedge funds (e.g. UBS O'Connor, Renaissance Technologies), market makers (e.g. Cutler Group LP, Group One Trading), and so on.

For each quarter ending at time p, we assume that we have all the related 13F filings immediately available, and build a holdings matrix $\mathbf{H}_p = (F \times A)_p$ where rows are investment institutions $f \in F$ and columns are assets $a \in A$. Each cell is populated with the number of shares that f is holding of asset a at the end of quarter p, or otherwise 0. Then, we compute differences in holdings between each two consecutive reporting periods and consider a new set of $T - 1$ matrices $\mathbf{D}_p = \mathbf{H}_p - \mathbf{H}_{p-1}$, with $p = 2, \ldots, T$. Between $30 - 40\%$ of securities are in every case found to have the related column fully populated with zeros. These are uncommon stocks chosen by only one or two funds that keep the position over longer horizons of time. These data amount to noise for our analyses and the related columns are thus dropped, completing the pre-processing.

2.2.3 Close-to-close price returns

To investigate whether disclosures on holdings can provide novel information on the state of branches of the market, we download close-to-close adjusted daily returns for our stocks from the Center for Research in Security Prices (CRSP) dataset from Wharton Research Data Services (WRDS). The return of the S&P Composite Index is also selected to be able to compute market-excess returns. We map our CUSIP6 identifiers to tickers and obtain returns data for approximately $5,000$ securities per each reporting period.

2.3 Methodology

The set of matrices $\mathbf{D}_p$ describes differences in the amount of shares of stocks $a \in A$ held by institutions $f \in F$ between each consecutive periods $p-1$ and p. We now leverage such variations to construct a "predictive" signal for the assets in question. For each asset a, we proceed by calculating the volume of shares $B^{vol}_{p,a}, S^{vol}_{p,a} \geq 0$ that are respectively bought or sold by market participants. Similarly, we compute the number of buying and selling trades $B^{tr}_{p,a}$, $S^{tr}_{p,a} \geq 0$ on each stock, by counting the number of entries with either positive or negative sign.

We define an *imbalance* of allocations over asset a, between periods $p-1$ and p, by considering the normalised volume discrepancy

$$I^{vol}_{p,a,N} = \frac{B^{vol}_{p,a} - S^{vol}_{p,a}}{B^{vol}_{p,a} + S^{vol}_{p,a}} \in [-1; +1], \tag{2.1}$$

and define similarly an imbalance in terms of trade counts by

$$I^{tr}_{p,a,N} = \frac{B^{tr}_{p,a} - S^{tr}_{p,a}}{B^{tr}_{p,a} + S^{tr}_{p,a}} \in [-1; +1]. \tag{2.2}$$

To increase the significance of our measure, we compute imbalances on securities that have at least N institutions "active" on them. *Activity* of fund f on stock a is defined as having $D^{(f,a)}_p \neq 0$. In this way, we avoid e.g. the unreliable case of having irrelevant strongest signals $I^{vol,tr}_{p,a,1} = \pm 1$ associated to stocks being traded only by one institution. Other normalisations could be considered for $I^{vol}_{p,a,N}$, such as by the total traded volume in the market over the same period, but are here not experimented.

As already mentioned, we assume that all data from Form 13F-HR are made public the day after the end of the financial quarter. At this point in time, we then compute imbalances and gauge them as a possible signal of crowding. We divide imbalances between quantile ranks qr_i for $i \in \{1, 2, 3, 4, 5\}$, that are defined as

$$qr_i = \text{top } [100 - 20 \times (i-1)]\% \text{ imbalances largest in magnitude,}$$

where we neglect imbalances equal to zero. In this way, we are able to focus on the strongest segments of our available signals and reinforce the N threshold. Then, we test whether investing against related assets can become a profitable and statistically significant strategy. Specifically, our *vanilla strategy* leverages either I^{vol} or I^{tr}, and *shorts* each asset for which $\text{SIGN}(I^{vol,tr}_{p,a,N}) = +1$, while it *goes long* assets with $\text{SIGN}(I^{vol,tr}_{p,a,N}) = -1$.

We experiment with multiple hypothetical strategies, which are defined by the choices of: $I^{vol,tr}$, qr_i, and future markout horizons $m \in \{5, 10, 21, 42, 63\}$ trading days ahead of the end of each quarter p (where we assume an average of 21 trading days in a month). The raw future return $fret_{p,a}^{m,RAW}$ of an investment on asset a is computed summing close-to-close adjusted daily returns over the horizon of interest, and then we define the future market-excess return $fret_{p,a}^{m,MER}$ as

$$fret_{p,a}^{m,MER} = fret_{p,a}^{m,RAW} - fret_{p,SPY}^{m,RAW}, \tag{2.3}$$

where SPY is the S&P Composite Index ETF. Market excess returns allow us to remain orthogonal to the market component and hedge our theoretical investments. The related *profit-and-loss* (PnL) from our strategy amounts to

$$PnL_{p,a}^{m,MER} = fret_{p,a}^{m,MER} \times (-1) \times \text{SIGN}(I_{p,a}). \tag{2.4}$$

For clarity, we represent in the diagram in Fig. 2.1 our quantities and timelines of interest.

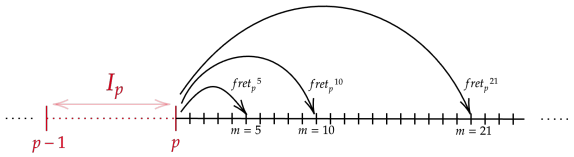


Figure 2.1: Illustration of the timeline on which we extract imbalances and compute future returns for each quarter p.

At each period p, we allocate capital uniformly (i.e. $\$\frac{1}{|A_p|}$) to each one of the stocks $a \in A_p$ chosen by our strategy and to SPY. This choice allows us to have comparable long-short portfolios in terms of capital invested. A different weighting scheme can also be implemented, e.g. by trading stocks proportionally to the magnitude of their imbalance. However, this type of modification is redundant to the quantile approach and can lower the expected profit. Thus, we proceed with the uniform weighting scheme. Then, we iterate the above investment procedure at the end of each financial quarter, i.e. when we assume to see the newly released 13F filings. Finally, we compute summary statistics to judge the actual information provided by imbalances and their effectiveness at different markouts. We compute the PnL for each period p as

$$PnL_p = \sum_a PnL_{p,a}^{m,MER}. \tag{2.5}$$

The full performance of our strategy is then given by adding each result in time, i.e. $PnL_{tot} = \sum_{p=p_i}^{p_f} PnL_p$ where $p_{i,f}$ are the first and last quarter with available imbalances' data. To keep track of the *profit-per-trade* (PPT), we compute

$$PPT = \text{AVG}_p\Big(\frac{PnL_p}{|A_p|}\Big) \tag{2.6}$$

for each configuration. This metric is useful to qualitatively compare the profits after execution costs for each portfolio. Finally, we also compute the *Sharpe Ratio* S as

$$S = \frac{\text{AVG}_{p=p_i}^{p_f}(PnL_p)}{\text{STD}_{p=p_i}^{p_f}(PnL_p)} \times \sqrt{4}, \tag{2.7}$$

to value the profit achieved by unit of risk taken. Since S is usually reported annualised, we process it accordingly by considering that we are investing once per quarter in each year.

The works of [14], [78] and [89] further highlight the importance of checking the significance of Sharpe Ratios when back-testing a sample of hypothetical strategies. The general motivation lies in the reality that, as one tests more and more strategies at the same significance level, the overall probability of choosing at least one poor strategy grows (known as "multiple testing problem" and related to the problem of backtest overfitting). The authors of [14] hence consider a set of strategies $k \in K$, referred to as *trials*, with associated Sharpe ratio estimates $\{S_k\}$. Then, they suppose that these trials follow a normal distribution with mean $E[\{S_k\}]$ and variance $V[\{S_k\}]$. This claim is supported by the fact that the concept of strategy class, of our interest, implies its trials to be bounded by some common characteristic pattern. By leveraging [12], the authors proceed to show that the expected maximum of $\{S_k\}$ when $|K| >> 1$ can be approximated as

$$E[max\{S_k\}] \approx E[\{S_k\}] + \sqrt{V[\{S_k\}]}\Big((1-\gamma)Z^{-1}[1-\frac{1}{|K|}] + \gamma Z^{-1}[1-\frac{e^{-1}}{|K|}]\Big), \tag{2.8}$$

which makes indeed explicit how the number of independent trials $|K|$ affects the expectation of the maximum Sharpe ratio achieved. In the above, $\gamma \sim 0.5772$ is the Euler-Mascheroni constant and Z is the cumulative function of the standard Normal distribution. Equation (2.8) is finally used to propose a statistic that corrects a Sharpe ratio for both the number of trials tested (as just discussed) and the fact that Sharpe ratio returns are generally sourced from non-Normal distributions (see [13]). This statistic becomes a confidence level C_k of the performance of each one of the strategies $k \in K$, computed as

$$C_k = Z\Big[\frac{(S_k - S_0)\sqrt{L-1}}{\sqrt{1-\gamma_3 S_k + (\gamma_4 - 1)S_k^2/4}}\Big], \tag{2.9}$$

and which we assess to gauge the significance of our own results. In the above, S_k is the Sharpe Ratio of the specific strategy we are testing, L its sample length, γ_3 the skewness of the related returns distribution and γ_4 its kurtosis. Finally,

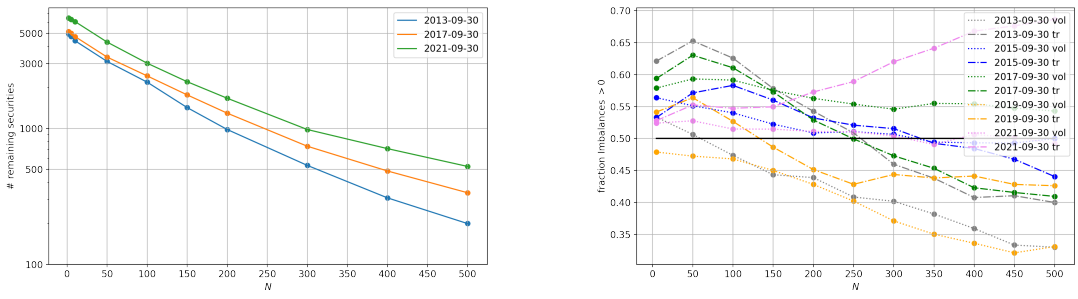
$$S_0 = E[max\{S_k\}] - E[\{S_k\}] \tag{2.10}$$

is the term that deflates our Sharpe Ratio to account indeed for selection bias and overfitting due to multiple testing.

2.4 Preliminary analyses of imbalances

2.4.1 Considerations on threshold N

Before investigating whether 13F imbalances carry a profitable signal, and can thus be interpreted as an enhanced signal of crowding, we check the implications of choosing a different threshold N. For each period p, we vary $N \in [0, 500]$ with step size $= 50$ and count the remaining number of securities, i.e. how many assets have at least N funds active on them at that point in time. We also compute the related imbalances, following Eqs. (2.1) and (2.2). Figure 2.2a shows the number of remaining securities for different N in Q3 of 2013, 2017 and 2021. This is a log-linear plot and the witnessed linear relationship implies that we can approximate the number of surviving securities via an exponential decay dependent on the threshold N. The shift between the three temporal lines expresses the increase in active institutions over the years, as intuitively expected from the expansion of markets, but this does not influence the shape of the relationship.



(a) (b)

Figure 2.2: We vary the threshold N on the x-axis and (a) look at how many securities (i.e. number of imbalances) are left to possibly trade, (b) compute the fraction of volume and trades imbalances with positive sign.

In Fig. 2.2b, we decompose the above relationship between remaining securities with positive or negative imbalance and compute the related ratio. We also look at the evolution of this ratio in time in Figs. 2.3 and 2.4, for imbalances computed in terms of volume or trades, respectively. We observe that there is a predominance of buying (selling) behaviour with decreasing (increasing) N. However, there is also a stable tendency of reversion towards the balanced proportion. This is especially true for I^{vol}, while I^{tr} shows a more persistent bias in time. This difference is due to the two natures of the quantities. While the former considers the actual amount of shares traded in the market, but can include special strong long niche bets from highly knowledgeable investors, the latter relates specifically to the number of buying or selling trades and will strongly react to market conditions. We can thus infer that investors tend to start or increase their common positions at different points in time, but they will decrease or exit them more synchronously. This result is again in agreement with intuition. Finally, the oscillations of Fig. 2.3 can be related to liquidity injections or withdrawals.

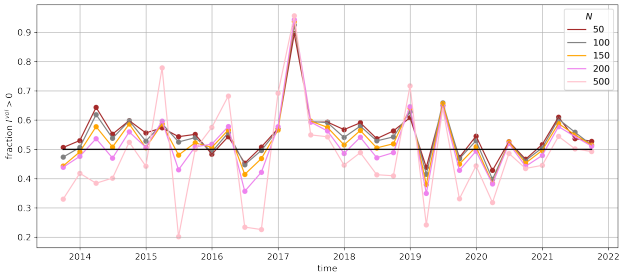


Figure 2.3: Evolution in time of the fraction of volume imbalances with positive sign for different choice of threshold N.

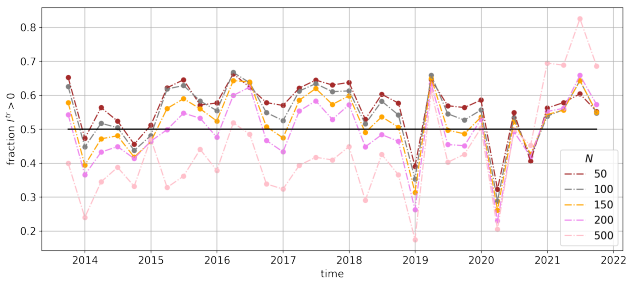


Figure 2.4: Evolution in time of the fraction of trades imbalances with positive sign for different choice of threshold N.

For the purpose of our analysis, we will consider and compare imbalances generated by $N \in \{50, 150, 500\}$. The case $N = 150$ will be the least unbalanced one in terms of sign proportions, but also $N = 50$ does not divert too strongly from equilibrium. These two options allow us to consider more than $1,000$ or $3,000$ securities, respectively. We are also interested in investigating the results of trading those securities on which investors are most active on, and to this end, we further consider $N = 500$. Due to the biases witnessed, we also build a data set where we cross-sectionally demean the signals from imbalances, in order to investigate a possible related impact on the performance of the strategies.

2.4.2 Price impact of imbalances on contemporaneous returns

Our investigation tests for systematic signals of crowding, which is expected to unwind in the future and cause a significant movement in prices. But for completeness, we also assess the impact of imbalances over contemporaneous returns, in order to weight the extent to which fund managers, which are entering into a position over a given quarter, impact the price return of the asset over the same quarter. We perform a linear regression of raw returns (and market-excess returns) onto the related imbalances $I_N^{vol,tr}$ with $N \in \{50, 150, 500\}$, at each quarter p. Quarterly returns are calculated summing the adjusted close-to-close daily returns covering the three months within each quarter, but these are first winsorized to control for the lowest and highest 10% of values, which could include noisy outliers. Finally, we compute the R^2 of each regression resulting from the different combination of variables and quarter p considered.

In Table 2.1, we report the average R^2 over the whole set of periods p, i.e. $\forall p$, for the above combinations. To investigate periodical patterns, we also report the average R^2 aggregated across each specific quarter (Q1, Q2, Q3 and Q4), over the years available in the study. Imbalances computed over trade counts generally attain higher average R^2 compared to the volume ones, and their magnitudes are particularly enhanced for Q3 with $N = 500$. One possible explanation of the latter result is that major investors might act very similarly in entering and exiting positions during Q3, following the yearly Russell Index Rebalancing. All R^2 values are fairly low, which is somewhat expected since we are dealing with noisy financial data, and are considering long periods of 3 months. On top of that, this suggests that the considered institutional inflows and outflow do not foment momentum in such trades, and we can thus further expect the unwind of crowding after the financial quarter ends.

Returns	N	$\mathbf{I^{vol,tr}}$	$\mathbf{R^2}$%, $\forall$p	$\mathbf{R^2}$%, Q1	$\mathbf{R^2}$%, Q2	$\mathbf{R^2}$%, Q3	$\mathbf{R^2}$%, Q4
Raw rets	500	I^{vol}	0.90	1.46	0.67	0.65	0.86
Raw rets	500	I^{tr}	2.40	2.64	0.87	4.40	1.46
Raw rets	150	I^{vol}	0.47	0.31	0.58	0.53	0.44
Raw rets	150	I^{tr}	1.08	1.22	0.86	1.76	0.40
Raw rets	50	I^{vol}	1.05	1.22	1.47	0.90	0.61
Raw rets	50	I^{tr}	1.98	2.01	2.71	2.37	0.80
MERs	500	I^{vol}	0.86	1.16	0.83	0.69	0.77
MERs	500	I^{tr}	2.59	2.79	1.78	3.91	1.71
MERs	150	I^{vol}	0.45	0.21	0.63	0.58	0.35
MERs	150	I^{tr}	1.09	1.23	0.87	1.57	0.61
MERs	50	I^{vol}	1.04	1.14	1.62	0.91	0.52
MERs	50	I^{tr}	1.89	1.91	2.67	2.19	0.77

Table 2.1: For each combination of return type and imbalances I_N^{vol} and I_N^{tr}, we report the average R^2 in percentage of the related regressions over the whole set of periods p. To investigate periodical patterns, we also show the average R^2 calculated for each quarter (i.e. Q1, Q2, Q3, Q4) over the years

2.5 Strategy construction: betting on crowding unwind

2.5.1 Vanilla strategy

Imbalances in volumes and trades are computed for $N \in \{50, 150, 500\}$, following the methodology described earlier and without de-meaning at first. We invest on horizons of $m \in \{5, 10, 21, 42, 63\}$ trading days and compute the cumulative PnL from future MERs. The full sets of results are reported in Figs. A.1, A.2 and A.3 in Appendix A.1, while we show the performance on the 21-days horizon in Fig. 2.5.

The striking feature, which is common to all the simulations and for which Fig. 2.5 acts as a representative, is that trading against the sign of imbalances tends indeed to lead to positive PnL for market excess returns. Therefore, we proceed to compute the PPT and PnL of each strategy, and show results in Fig. 2.6. Figure 2.7 reports the significance levels of the Sharpe Ratios if less than 0.05, where strategies that did

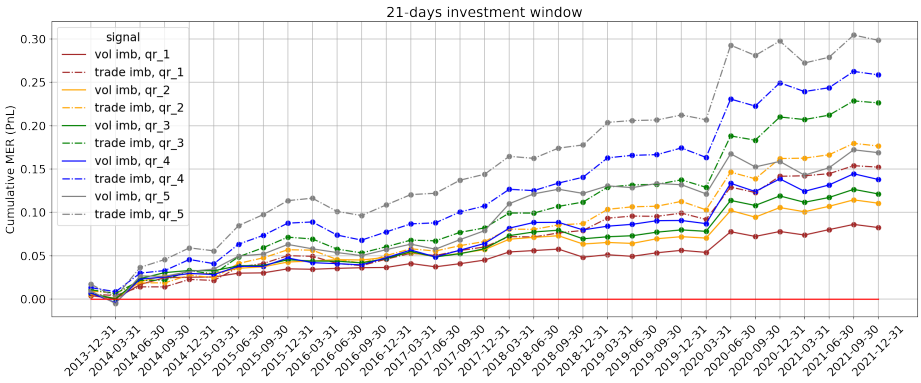
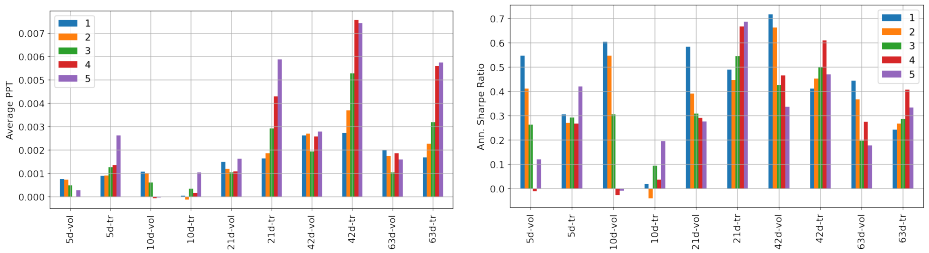
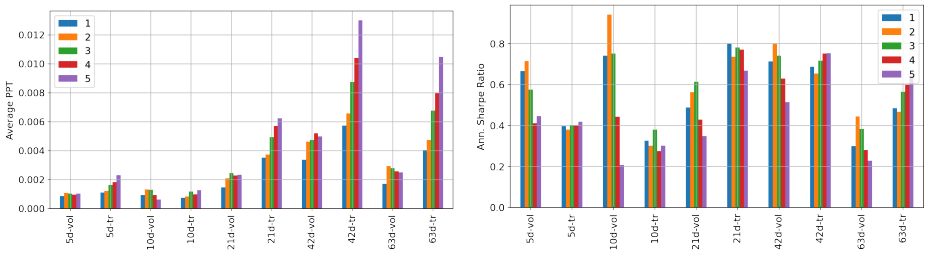


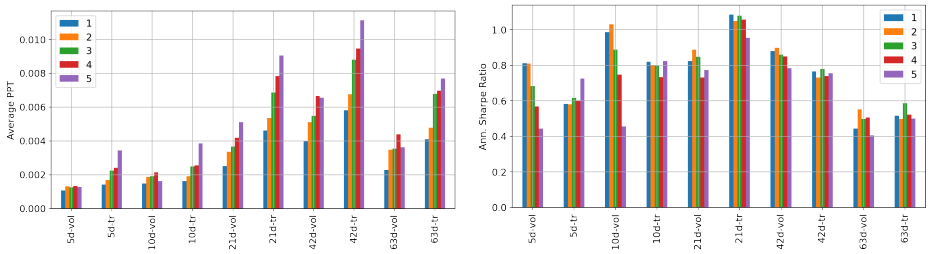
Figure 2.5: Cumulative PnL from MERs on a 1 month horizon, when trading against imbalances with $N = 50$.



(a) $N = 500$, average PPT (b) $N = 500$, Sharpe Ratio

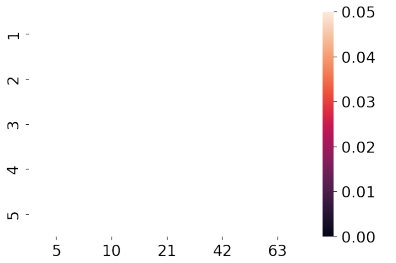


(c) $N = 150$, average PPT (d) $N = 150$, Sharpe Ratio

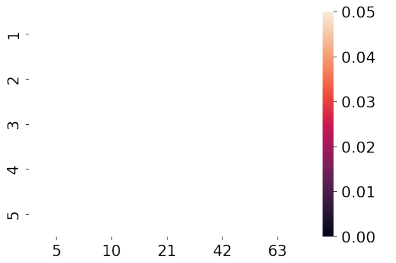


(e) $N = 50$, average PPT (f) $N = 50$, Sharpe Ratio

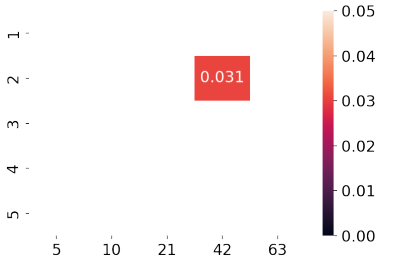
Figure 2.6: Average PPT and Sharpe Ratio statistics on our vanilla strategy for $I^{vol,tr}$. The different colors relate to the different quantile ranks as highlighted in the legend.



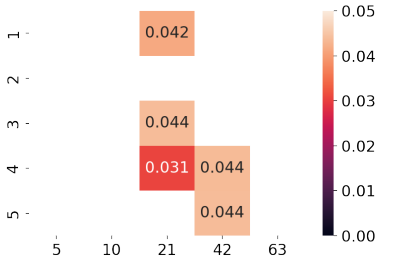
(a) $N = 500$, vanilla strategy on volume imbalances



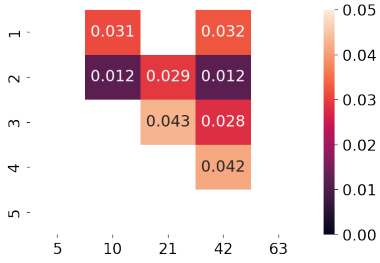
(b) $N = 500$, vanilla strategy on trade imbalances



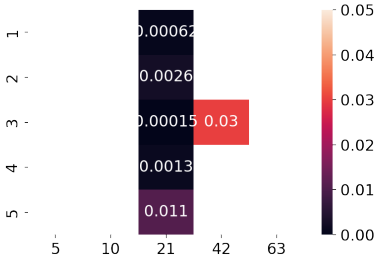
(c) $N = 150$, vanilla strategy on volume imbalances



(d) $N = 150$, vanilla strategy on trade imbalances



(e) $N = 50$, vanilla strategy on volume imbalances



(f) $N = 50$, vanilla strategy on trade imbalances

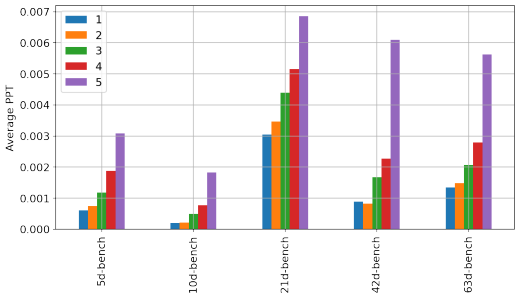
Figure 2.7: Results when testing the significance levels of our Sharpe Ratios. Strategies rejected by the test are reported with null value for ease of visualisation. In each heatmap, the x-axis is the horizon of the strategy and the y-axis is the quantile rank used for trading.

not pass the significance test are reported as null values for ease of visualisation. We observe that no strategy has significant Sharpe Ratio when $N = 500$, and proceed to experiment with de-meaning the signals from imbalances. However, no significance arises at $N = 500$ and the performances and significance at $N = 50, 150$ are negatively affected. We conclude that de-meaning is not suitable on 13F filings imbalances.

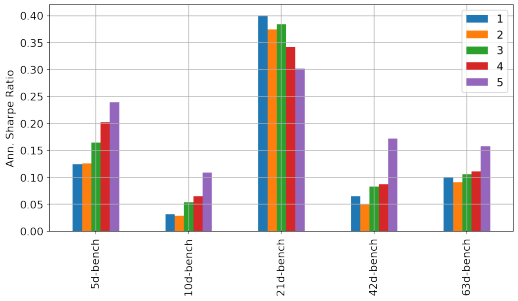
From a pure investment purpose, we compare Sharpe Ratios and average PPT of the strategies passing the significance test. It is indeed meaningful to look at the

profit-per-trade to have a qualitative view of performances when we consider costs of executions from rebalancing. Our vanilla strategy achieves a desirable Sharpe Ratio $S > 1$ if we trade against trade imbalances on a horizon of 21 trading days, for $N = 50$ and quantiles qr_4, qr_5. Otherwise, we achieve good PPT and $S > 0.8$ by trading against volume imbalances on a longer horizon of 42 days again for $N = 50$. This result is in line with intuition, since the above considerations allow us to infer that imbalances are a signal of crowding and over-heating of trades, and trade imbalances are consequently expected to be the most suitable related metric.

To further assess the strength of our findings, we introduce a benchmark strategy trading mean-reversion of prices, since betting against crowding can be seen as an extension of trading against momentum. Instead of using imbalances, the benchmark trades opposite to the sign of past returns over the last quarter. The resultant average PPT and Sharpe ratios are shown in Fig. 2.8 for the same universe of securities as our optimal $N = 50$ case, allowing us to assess whether the information carried by 13F filings does indeed provide a better landscape for investing. If we compare results for the benchmark and our vanilla strategy over the optimal horizons of 21 and 42 trading days, it is clear that imbalances from 13F filings allow us to achieve better investment performance. Thus, we unravelled alternative evidence of crowding and over-heating of trades, which needs to be considered as a point of concern by investors but could also be traded upon. Importantly, 13F filings are here assumed to be all available at the end of each financial quarter, as we already mentioned. The possible delay in publication can be incorporated by dynamically updating imbalances with the arrival of novel filings, but our strategy uncovers anyways strong evidence of the importance of accounting for crowding and over-heating in trades for risk management.



(a) Benchmark, average PPT



(b) Benchmark, Sharpe Ratio

Figure 2.8: Average PPT and Sharpe Ratio statistics on our benchmark strategy that trades against the sign of stocks' past returns, i.e. betting on the mean-reversion of prices. The different colors relate to quantile ranks as in the legend.

2.5.2 Contrary to Market Momentum and to Market Mean-Reversion conditioning

To assess whether we can further enhance the profits achieved by our strategies, and consequently gain more insights into crowding, we condition the imbalances that we trade according to the past performance of the related security. We now hold positions for horizons of $10, 21, 42$-trading days but further consider whether the MER of each security was positive or negative over the past $10, 21, 42$ days. We define our strategies being:

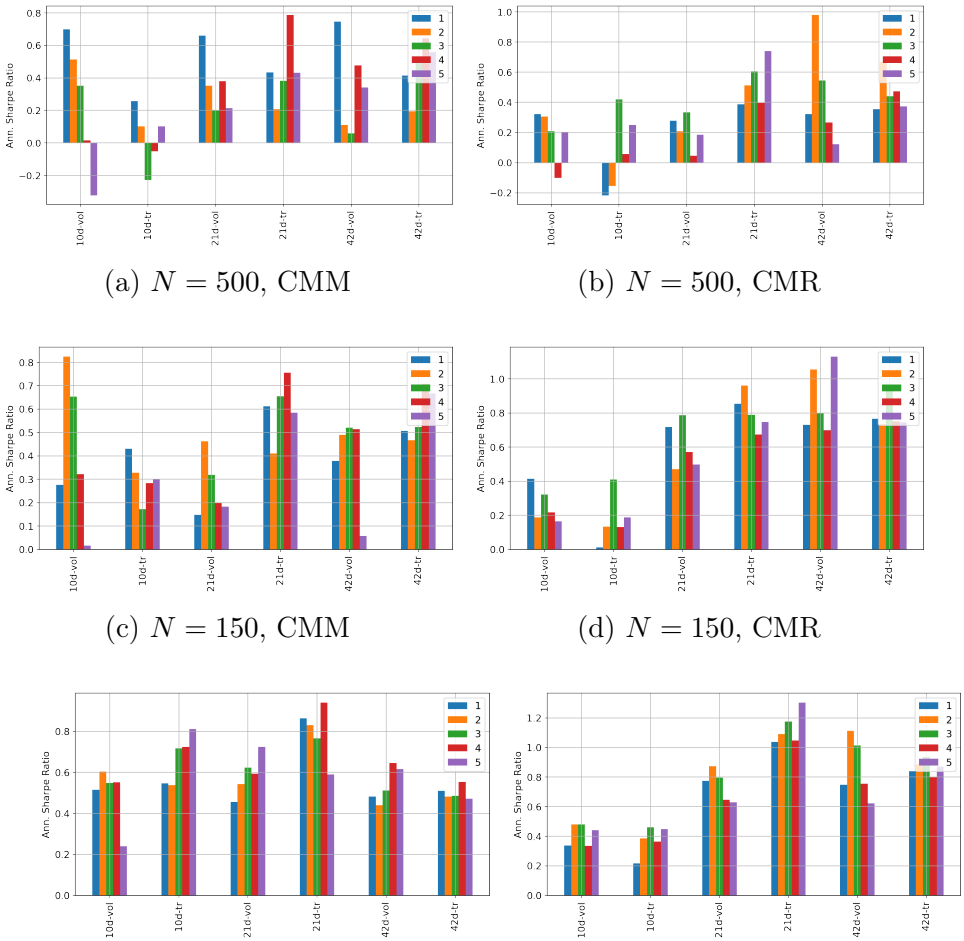
- *contrary to market momentum (CMM)*, if we trade contrary to imbalances that have positive (negative) sign while the related securities showed past positive (negative) performance,
- *contrary to market mean-reversion (CMR)*, if we trade contrary to imbalances that have positive (negative) sign while the related securities showed past negative (positive) performance.

Figure 2.9 shows the Sharpe Ratios following $I^{vol,tr}$ for the 5 quantile ranks, while Figs. 2.10 and 2.11 report the significance levels for CMM and CMR performances for the only interesting case of $N = 50$. We mainly see that high Sharpe Ratios are achieved when trading contrary to trade imbalances on a horizon of 21 days for $N = 50$, for both CMM and CMR strategies. This result suggests possible independence of the performance of imbalances to past prices, hinting to unresponsiveness of the trade to the recent inflows and outflows of investments, and higher likelihood of unwind.

2.5.3 Sector-specific analysis

We perform one final experiment and project imbalances of stocks onto sector memberships, using data from Wharton Research Data Services providing SIC (Standard Industrial Classification) Major Group for a set of securities. This is a code that classifies stocks into 10 sectors and its details can be found on the SIC website[2]. Table 2.2 provides a summary of the number of securities for which we have an imbalance and a SIC code, and assigns to each category a short label that we will use in our plots. We then run our vanilla strategy on each group for horizons of $21, 42, 63$-trading days (i.e. $1, 2, 3$ months). We require only $N = 50$, since a few sectors have already a limited number of securities to possibly trade. The resultant Sharpe Ratios are

[2] `https://siccode.com/page/structure-of-sic-codes`

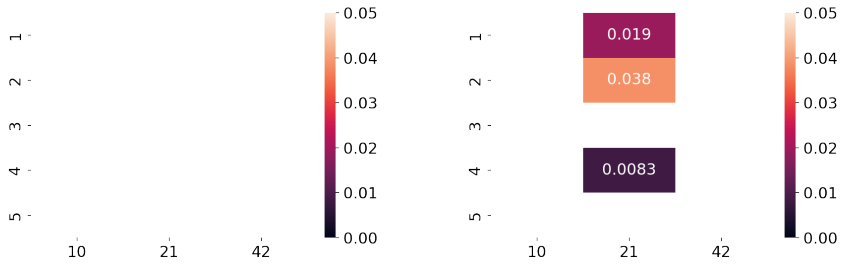


(a) $N = 500$, CMM (b) $N = 500$, CMR

(c) $N = 150$, CMM (d) $N = 150$, CMR

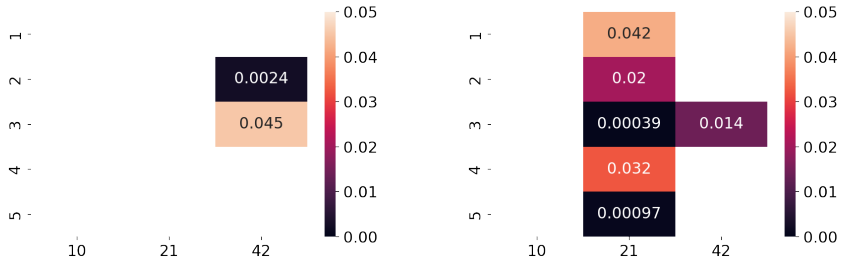
(e) $N = 50$, CMM (f) $N = 50$, CMR

Figure 2.9: Sharpe Ratios achieved for our Contrary to Market Momentum imbalances (CMM) and Contrary to Market Mean-Reversion imbalances (CMR) strategies.



(a) $N = 50$, CMM strategy on volume imbalances (b) $N = 50$, CMM strategy on trade imbalances

Figure 2.10: Significance level tests of our CMM Sharpe Ratios. Strategies rejected by the test are reported with null value for ease of visualisation. The x-axis is the horizon of the strategy, while the y-axis is the quantile rank.



(a) $N = 50$, CMR strategy on volume imbalances

(b) $N = 50$, CMR strategy on trade imbalances

Figure 2.11: Significance level tests of our CMR Sharpe Ratios. Strategies rejected by the test are reported with null value for ease of visualisation. The x-axis is the horizon of the strategy, while the y-axis is the quantile rank.

shown in Fig. 2.12, where subfigures relate to the different highest quantiles. We also compute the related significance levels, and see that only medium performances are acceptable, while the most profitable ones get rejected by the test.

We conclude that the major information that Form 13F carries is a suggestion to go contrary to what imbalances reveal, since further conditioning is not able to uncover any stronger signals. However, we want to mention a further possible research direction for the interested reader. This is the connection between average number (*popularity*) of active funds on the securities of a sector and the related growth and expectation of further inflows. We plot a few initial trends in Fig. A.4 of Appendix A.2 to intuitively show our thoughts. If one manages to increase the proportion of securities for which there are available all imbalances, returns, and SIC data, then it would be possible to monitor variations in the popularity of stocks within sectors with statistical significance. Consequently, one could identify signals of changes in inflows, related crowding, and leverage them to achieve stronger profits, or higher risk control.

SIC CATEGORY	**LABEL**	**No. SECURITIES**
Agriculture, Forestry, Fishing	agric	5/16
Mining	mining	98/352
Construction	constr	20/56
Manufacturing	manuf	521/1227
Transportation & Public Utilities	transp	161/441
Wholesale Trade	wholesale	70/149
Retail Trade	retail	71/194
Finance Insurance, Real Estate	fin-ins-RE	796/5136
Services	services	153/597

Public Administration	pubAdm	75/2897

Table 2.2: SIC sectors on which we investigate the performance of our contrarian vanilla strategy. We report the number of securities for which we have available a SIC code for each sector (denominator in the right column), along with the number of them for which we have also imbalances and returns data (numerator)

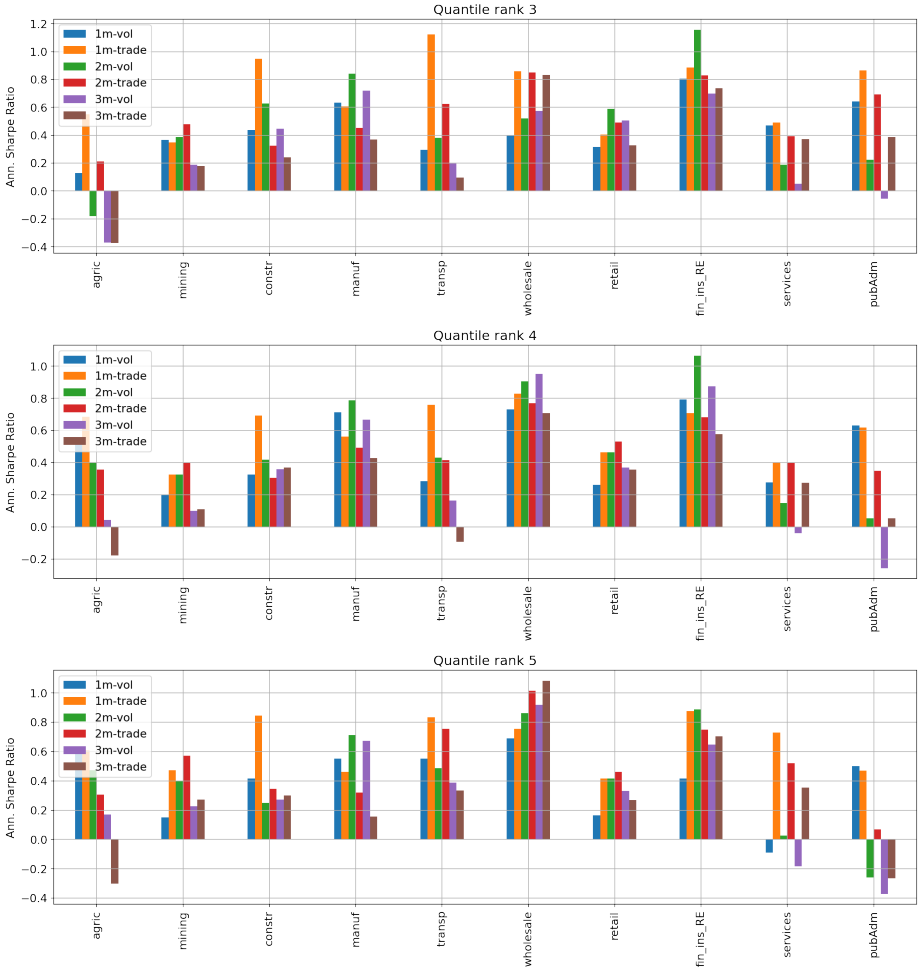


Figure 2.12: Sharpe Ratios achieved by our vanilla strategy when restricting the trading to the different sectors. We use a threshold $N = 50$ and test horizons of $m \in \{21, 42, 63\}$ trading days.

2.6 Conclusions

This study investigates the hidden information that is disclosed within the SEC Form 13F-HR. We leverage the concept of *imbalance* in buying vs. selling behaviour and

show that a possible opportunity for profit arises if an external investor is willing to trade contrary to these 13F filings imbalances. We find that a threshold of $N = 50$ provides a good trade-off between reliability of the imbalance signals and variety of the remaining stocks to possibly trade. Our simplest vanilla strategy achieves a desirable Sharpe Ratio $S \sim 1$ that is significant at the 0.05 level, when trading imbalance quantiles qr_4, qr_5 of I^{tr} to an horizon around 21 trading days. Trading on these two quantiles also shows higher average PPT, and returns prevail over a related benchmark strategy that trades on price mean-reversion.

Beyond such direct implications of our study for trading strategy construction, the results provide evidence that connects our imbalances to the riskiest crowded trades in markets. Imbalances quantify the level of coherence in the trading behaviour of our extremely eclectic set of market participants, and the strongest agreement (i.e. highest quantiles in our strategy) points to the most over-heated trades, for which we expect an imminent unwind. Indeed, the highest profitability of our strategy is achieved when investing opposite to *trade* imbalances, and they are the most suitable metric to depict crowding in the market. The optimal horizon around 21 trading days then highlights the time window of needed care, along which over-heated trades must be monitored by any market participant for risk control.

One possible further direction of research to explore concerns the effect of correcting reports with their amendments (Form 13F-HR/A) prior to running the suggested analyses, and compare the results. It is very likely that we will gain stronger insights into the extent to which fund managers, which are entering into a position over a given quarter, impact the price return of the asset over the same quarter. Yet another direction worth pursuing concerns the investigation of network effects in terms of the cross-impact of the imbalance associated to a given asset, in either the contemporaneous or future return of another asset, in the spirit of the recent work in [46]. On a longer time scale, one could also plan to pursue the described analyses on N-PORT holdings reports, once more history is indeed available. At the beginning of October 2016, the SEC introduced the form N-PORT to modernise portfolio reporting of both assets and liabilities of Registered Investment Companies and exchange-traded funds (ETFs) organised as unit investment trusts. This form became compulsory in May 2019 and provides a further level of market transparency and holdings details, with potentially valuable information to analyse.

Chapter 3

Interconnectedness of Economic News and Market Dislocations

We now shift to another type of alternative data frequently used within finance, namely news. In particular, we introduce a novel framework to study the dynamics of news narratives, by leveraging GPT3.5 advanced text analysis capabilities and graph theory. We focus on a corpus of economic articles from The Wall Street Journal and dynamically extract the main topics of discussion over time, in a completely systematic and scalable fashion. Then, we show how the structure of such topics of discussion has a statistically significant relationship with the contemporaneous state of financial markets, which can be used to construct an investment strategy or monitor financial risks. Our work is based on the intrinsic ability of GPT models to track the context of sentences within a document, thanks to which we can accurately extract a ranking of the most important entities discussed within each article, and evaluate their entity-specific sentiments. Then, we create a graph for each week of data, in which nodes are the entities retrieved and edges are built from the co-occurrence of such entities within articles. Graph centrality measures are computed over time to track the most representative keywords of topics of discussion, which result in an accurate summary view of the evolution of economic narratives. Fuzzy community detection is finally used to cluster linked entities into a more detailed representation of topics. Linking the features of these topics to the relevant financial market time series, we find that high fragmentation within our networks' communities relates to moments of financial markets dislocations (i.e. dates with unusually high volatility across asset classes). This result should thus motivate stronger effort within financial research to move beyond ubiquitous sentiment analysis of news, and delve deeper into broader and more holistic studies of textual data.

3.1 Introduction

In today's fast-paced digital age, the world is inundated with an unprecedented volume of information, particularly from *news* sources. The sheer magnitude of data generated daily has reached staggering proportions, making it increasingly challenging for individuals to parse and process this information accurately solely through human capabilities. News is constantly flowing from countless channels and platforms, and the need for advanced tools and technologies to sift through this vast sea of data has never been more apparent. In the realm of financial markets, the potential gain from efficiently handling such an enormous amount of news data, and extract quantitative signals from it, is even more pronounced. Financial markets are highly sensitive to information, and characterising *narratives* within news is surely one task that can enhance our knowledge on news' impact on asset prices, trading strategies, and investor sentiment.

The aim of this work is to introduce a novel approach, based on both Generative Pre-trained Transformer (GPT)3.5 advanced text analysis capabilities and graph theory, to accurately identify narratives within economic news. We also model the evolution over time of interrelations among related topics, and investigate whether the structure of discussion within news carries relevant information on the contemporaneous state of financial markets. Such line of thought conforms to evidence showing how agents infect each other with "investment ideas" [76], and assesses whether news have a related non-trivial effect on the behaviour of markets. Importantly, we desire to introduce a framework able to handle complex narratives properly (a need highlighted from previous literature [72]), but which is also generalisable and scalable.

As just mentioned, our work greatly relies on GPT advanced text analysis capabilities (especially in summarisation, question-answering, and entity extraction [27], [19]), and on graph theory. Digital text provides a rich repository of information about economic and social activity [55], which is why we leverage GPT to extract the main context-aware entities discussed within any article of interest. GPT and related fine-tuned models have also been assessed in [109] on their zero-shot learning ability, from which we gain further assurance on the satisfying ability of GPT3.5 in natural language inference tasks. Parallel branches of research rely on Semantic Role Labelling (SRL), which is a linguistic algorithm that identifies the action, the agent performing that action, and the patient being acted upon, within any sentence of interest. However, related work (see RELATIO [11] and CANarEx [5]) suggests that high complexity in the input corpus could affect the interpretability of downstream results. SRL is

generally applied to every sentence in each document, and it consequently produces a very wide set of relationships to consider that can dilute knowledge on the strongest drivers of information. For the sake of completeness, we also mention that [129] models topics via GPT similarity embeddings, but leave further details on it to the comparisons of topic modelling approaches of Table 3.1, which is later fully introduced.

In our work, we also leverage graph theory to investigate weekly networks generated from the co-occurrence of entities within articles that are extracted by GPT. Indeed, we pursue nodes' centrality analyses to understand the most important entities over time, and deepen into community detection to identify clusters of entities to map to interpretable topics. Co-occurrence networks have already been successfully used in various contexts. An example is [82], where the task of authorship recognition is investigated by comparing texts modelled as networks of words linked according to textual similarity measurements. Then, [122] uses textual forma mentis networks as quantitative tools to reconstruct the mindset of Twitter users engaging in social discourse. Moreover, [111] assesses the importance of enriching word co-occurrence networks with links that account for the semantic of words, via similarity scores of related word embeddings.

Before proceeding to the details of the proposed approach for narratives' detection within news, we now highlight interesting research on quantitative relationships between news and financial markets. In [123], the authors investigate the high-frequency interdependent relationships between the stock market and statistics on economic news in the US context. Then, [80] investigates how news affects the trading behaviour of different categories of investors in a financial market, while [120] finds evidence that market makers demand higher expected returns prior to earnings announcements, because of increased inventory risks that stem from holding net positions through the release of anticipated earnings news. In [110], the authors measure the correlation between returns of publicly traded companies and news about them, as collected from Yahoo Financial News.

Then, [66] tries to quantify how topics discussed within news influence the stock market. The authors of this paper apply a topic modeling technique called Latent Dirichlet Allocation (LDA) [124], in order to extract the keywords of information (i.e. "topics") that synchronise well with trading activity, measured by the daily trading volume. However, LDA assumes that topics are independent of each other, and its results are significantly affected by the level of care in the initial cleaning and pre-processing of documents [83]. Moreover, LDA topics are probability distributions of disconnected words, which can hinder interpretability of the results, especially with

corpora encompassing complex narratives [72] as in our financial and economic setting. Relatedly, the very recent research in [41] investigates the field of topic modeling in the context of finance-related news impact analysis, and further stresses how very limited literature exists. The authors compare three state-of-the-art topic models, namely LDA, Top2Vec [7], and BERTopic [60], and show that the latter performs best. The framework focuses on extracting topics and related sentiment of specific stocks, whose time series of returns are in connection investigated. However, the actual identification and interpretation of topics is still of questionable level, and the framework limits itself to facilitate efficient news selection based on membership within target topics, but do not properly analyse the latter. LDA is also used in [31], where the authors want to assess how strongly news text can be a mirror of the state of the economy. They find that news topic attention contains substantial information about future economic outcomes above and beyond standard indicators, which provides us with further confidence on the relevance of our study.

On the other hand, the recent emergence of novel Large Language Models (LLMs), such as indeed GPT ones, clearly marks a significant departure from earlier language processing techniques. These models leverage the power of deep learning and vast amounts of text data to achieve unprecedented levels of language analysis and generation, achieving outstanding improvements in Natural Language Processing (NLP) tasks, and opening countless novel paths of research. The paper in [132] presents a study on harnessing LLMs outstanding knowledge on human text construction for explainable financial time series forecasting of NASDAQ-100 stocks. Then, [63] leverages GPT parsing of companies' annual reports, to get suggestions on investment targets. On the other hand, [121] shows the potential improvement of the GPT4 LLM in comparison to BERT for modeling same-day daily stock price movements of Apple and Tesla in 2017, based on sentiment analysis of microblogging messages. Building upon the discussion of relevant research just outlined, we propose in Table 3.1 a comparative analysis of the most relevant competing works to the framework that we are proposing. This should better highlight the benefits of our approach, and further clarify how our work sits in context.

Table 3.1: Comparative analysis of the most relevant competing works on narratives extraction to ours, versus the framework that we are indeed proposing

Category	Approaches	Comparison to our framework
Topic detection via Latent Dirichlet Allocation (LDA).	Examples of related research are [66] and [31]. The first paper tries to extract keywords of "topics" that synchronise well with trading activity, from an LDA analysis of news. The latter paper applies LDA to economic news and tests for a relationship between news-based narratives and "shocks" in numerical economic data.	LDA-based approaches have proven their effectiveness in multiple applications. However, they need to assume that topics are independent, and they cannot easily update dynamically. Our proposed framework is designed to directly update topics over time, and allows for a more complex characterisation of both topics and their interrelations (by studying community detection on the related co-occurrence graphs).
Topic detection via BERTopic.	An example of related research is [41], which proposes evidence for the superiority of BERTopic against LDA for topic modelling in the context of finance-related news impact analysis.	BERTopic can be only applied to small samples of text and assumes that only one topic maps to each document analysed. When aiming to detect narratives within news, such features become strong limitations. With our approach, we allow news' articles of variable length, and enforce no assumption on the inner coherence of the theme of discussion.
Text similarity from embeddings.	The work in [129] is an example of approach that uses GPT to generate embeddings of text, which are then clustered according to their similarity (i.e. to identify possibly associated topics). Importantly, this work focuses on extracting the main topics characteristic of a corpus of abstracts of scientific publications.	The mentioned approach only works with short texts because of input size restrictions of GPT-3 similarity embedding models. Moreover, its authors underline strong sensitivity to outliers. Our interest in modelling complex economic narratives implies that we need to unravel different topics within same news and be resilient to outliers, to achieve an accurate view of the current themes of discussion. Thus, our own approach to extract context-aware ranked entities from each article of interest is able to overcome such limitations.

Continued on next page

Table 3.1 – continued from previous page

Category	Approaches	Comparison to our framework
SRL-based narratives' detection.	RELATIO [11] and CANarEx [5] are state-of-the-art approaches for narratives' detection via SRL. The former considers all sentences in a corpus of documents and, for each sentence, differentiates its characteristic entities and associated actions. Results are clustered both semantically and syntactically, and a directed multigraph is then generated to study the landscape of "narratives" generated. CANarEx builds upon RELATIO, but incorporates a pipeline to allow co-reference resolution. This consequently improves the context-awareness of the relationships of extracted entities.	This advanced branch of research on narratives' detection has proven significant ability to identify representative constructs of discussion. However, such techniques rely on the parsing of every sentence within each document, which can introduce computational cost and complexity of results. Our approach builds on a parallel view, and relies on GPT ability to extract a ranking of the most central entities of discussion within any text. This generates a focused and conservative set of entities from which we build our co-occurrence graphs. When the corpus of interest covers a multitude of broad and complex narratives as in news, our method maintains high interpretability and clear modelling of themes of discussion.
Our GPT-based graph approach.	Due to GPT advanced ability to complete tasks of text summarisation, question-answering, and entity extraction (see [27], [19]), we identify a ranking of the five most important entities discussed within each piece of news of interest. The co-occurrence of entities within articles is then encoded into related graphs, which allow us to identify evolving topics and their mutual relationships over time.	*Not Applicable*

Main contributions. Our research introduces a novel framework for topic identification within economic news. Indeed, we leverage on GPT3.5 to extract the main entities (both concrete or abstract) mentioned within each article in an available corpus, and aggregate entities' co-occurrence among articles in weekly graphs. Studying the resultant set of graphs allows us to identify communities of entities that can be mapped back to interpretable non-trivial topics. Since there is no benchmark dataset for the evaluation of topic detection [72], we rely on different metrics to assess our results. By looking at basic metrics such as degree and eigenvector centrality of nodes, we characterise the evolution over time of central themes of discussion according to our model. And indeed, we show that our results accurately map to the principal topics of economic debate occurring during the years considered. Furthermore, we

randomly sample the communities of entities that we identify over time (i.e. defining topics) and manually check for their inner coherence and representativeness within their articles of origin. Within this piece of work, we also propose to consider the sentiment around main entities of an article as a more accurate proxy for the overall sentiment of such piece of text, and describe a case-study to motivate this choice. One final contribution of our study is the investigation of news' features in relation to financial market dislocations, via a logistic regression. We design attributes that characterise both the local and global structure of news, and their sentiment and interconnections. This allows us to find quantitative evidence of high entropy in the high-dimensional space of interconnected news, when the latter are associated to moments of unusually high volatility across asset classes. Consequently, we also find evidence of the effectiveness of our framework for topic detection, since the successful regression implies that relevant information is carried by our graph constructs about the evolving state of economic discussion and concerns.

Structure of the Chapter. Section 3.2 introduces the data we collect, and some initial related processing. Section 3.3 clarifies the theoretical knowledge that is necessary to understand the approach taken, which is mainly comprised by notions from both NLP and graph theory, and on related embedding techniques. Then, Section 3.4 highlights our analyses on narratives and market dislocations, and describes the results achieved. Finally, we conclude this work with some last remarks in Section 3.5.

3.2 Data

3.2.1 Corpus of news

We download a tractable corpus of news from Factiva[1] data provider by looking for news written in English, which also belong to the "Economics" section of the Wall Street Journal (WSJ). In this way, we aim at having a set of news that carries a low-noise and focused view on the evolution of themes that can be of help for financial markets understanding. We consider approximately four years of daily news, i.e. from January 2020 to October 2023, and aggregate them at weekly level. After pre-processing them to a standard format, we achieve a overall dataset of 197 weeks with a total of $21,590$ news, with 110 ± 21 data points per week (i.e. average number of articles and its standard deviation). Importantly, we consider the week ending

[1] `https://www.dowjones.com/professional/factiva/`

on 14th March 2021 as an outlier and drop it, since we could only download three associated articles from Factiva due to an issue on their end.

3.2.2 Market dislocations

Our aim is to identify and quantify the evolution of narratives within news, i.e. the interchange of topics discussed over time and surrounding nuances, but with the further end goal to unravel consequent relationships to the evolution of financial markets. In particular, we are interested in financial market dislocations, which are often recognised as moments when "financial markets, operating under stressful conditions, experience large, widespread asset mispricings" [106]. However, we decide to adopt a more data-driven definition of market dislocations, and consider them as dates when combined shocks to equity, FX, bond, and macro factor risk premium indices occur, i.e. shocks to all the major asset classes. This is of interest to us, since our studies could then predict strong simultaneous downwards trends during which diversification fails to provide its intended protection.

We begin by downloading the following four indices from Bloomberg L.P. at a weekly frequency, with the proposed descriptions taken from its interface:

1. VIX Index - "The VIX Index is a financial benchmark designed to be an up-to-the-minute market estimate of the expected volatility of the S&P 500 Index, and is calculated by using the midpoint of real-time S&P 500 Index option bid/ask quotes."

2. JPMVXYEM Index (VIX FX) - "J.P. Morgan Emerging Market Currency Implied Volatility Index."

3. MRI CITI Index - "The Citi Macro Risk Index measures risk aversion based on prices of assets that are typically sensitive to risk. A reading above (below) 0.5 means that risk aversion is above (below) average."

4. MOVE Index - "The MOVE Index measures U.S. bond market volatility by tracking a basket of Over-the-Counter options on U.S. interest rate swaps. The Index tracks implied normal yield volatility of a yield curve weighted basket of at-the-money one month options on the 2-year, 5-year, 10-year, and 30-year constant maturity interest rate swaps."

Then, we compute the related z-scores for a rolling window ΔT of three months, i.e. 13 weeks if we assume one year to be made of 52 weeks. The z-score is defined as

$$\text{z-score} = \frac{i - \mu}{\sigma}, \tag{3.1}$$

where in our case i is the current value of the index, μ is its mean over the previous time range ΔT, and σ is the related standard deviation. Intuitively, the z-score shows how many standard deviations above the mean the current outcome is. Figure 3.1 proposes the dates for which all our four indices have positive z-score. According to the strength of these z-scores, broad dislocations across asset classes could be implied, which is why we specifically define market dislocations as follows.

Definition 3.2.1 (Market dislocation)**.** A *market dislocation* is identified as a week when all the z-scores of our chosen volatility indices (i.e. VIX, VIX FX, MRI CITI, and MOVE) are strictly positive, and their average is above 0.5. The z-scores are calculated over windows of 13 weeks.

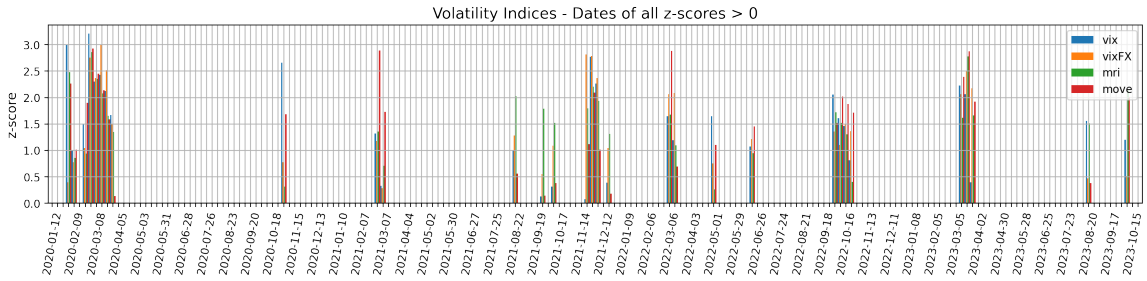


Figure 3.1: Weeks on which our volatility indices have all positive z-scores, with respect to a rolling window of length $\Delta T = 13$ weeks. According to the strength of these z-scores, broad market dislocations can be consequently identified.

3.3 Framework

3.3.1 Natural Language Processing for Topic Modelling

Natural Language Processing (NLP) has undergone significant advancements in recent years, revolutionising the field of computational linguistics. One of the noticeable but challenging applications of NLP is topic analysis, a technique that involves the extraction of latent themes and subjects from extensive textual corpora. Among the first methodologies proposed for such task, LDA is one of the most famous ones [124].

LDA first transforms each document of a corpus into a set of words with related Term Frequency - Inverse Document Frequency (TF-IDF) value, which is a measure of relative importance of words across documents. This is computed as

$$\text{TF-IDF} = \textsc{Term Frequency (TF)} \times \textsc{Inverse Document Frequency (IDF)}, \tag{3.2}$$

where TF is the count of a target word in the current document considered (normalised by the total number of words in the document), and IDF is the inverse of the count of occurrences of such word in the whole set of documents (normalised by the total number of documents). Intuitively, the TF-IDF weights of words per document are lower if the word appears in more documents and so does not carry strong characterisation power, but higher if instead it appears often in just one specific document. Then, LDA assumes that each document is generated from a collection of topics in some certain proportion, and that each topic is itself a group of dominant keywords with specific probability distribution. Consequently, we can try to back-engineer the root topics used to generate a large collection of documents, by analysing the co-occurrence of individual tokens and looking for the joint probability distribution of our given observable and target variables. However, the task of assigning a meaningful label to each set of terms (i.e. forming a topic) must be done by the user.

Then, a big advancement in topic modeling arose with the introduction of BERTopic [60], built upon the Bidirectional Encoder Representations from Transformers (BERT) language model. BERTopic generates document embedding with pre-trained transformer-based language models, clusters these embeddings, and finally, generates topic representations with a class-based variant of the TF-IDF procedure[2]. However, multiple weaknesses are still characteristic of this technique. The model assumes that each document contains only one single topic, and words in a topic result to be often very similar to one another and redundant for the interpretation of the topic itself. Polysemy (i.e. when words have multiple meanings in different contexts) is also a further challenge. Thus, reliable and systematic downstream applications of BERTopic are considered difficult to achieve.

GPT models represent a groundbreaking advancement in the realm of NLP and artificial intelligence (AI), due to their remarkable ability to simulate human-like text. A GPT model is a decoder-only transformer model of a deep neural network, which uses attention to selectively focus on segments of input text that it predicts to be the most relevant. It is extensively pre-trained and has been proven to achieve

[2] `https://maartengr.github.io/BERTopic/api/ctfidf.html`

groundbreaking accuracy on multiple multimodal tasks [1]. Despite there have not been related advancements on systematic topic modeling yet, we propose a novel methodology that uses it to achieve indeed such scope.

GPTs and downstream extraction of topics and narratives. As introduced in Section 3.2.1, our news data are descriptive textual content, that we instead aim to analyse quantitatively. Our goal is to extract information on the development of topics and narratives in a systematic but interpretable way. We did experiment with the early topic modeling methodologies described above, but found strong limitations (e.g. labeling the topics from LDA and BERTopic is oftentimes highly subjective and challenging, and such topics can extensively overlap, making it difficult to distinguish between them). Thus, we decided to leverage the proven ability of the GPT3.5 model to complete tasks of summary composition, sentiment analysis, and entity extraction [113], to achieve our goal. We could of course have directly questioned GPT models for topics and narratives characteristic of news articles, but their inherent randomness and lengthiness in the formulation of answers makes it difficult to identify comparable and reliable results, to then use for downstream tasks.

GPT3.5 is thus used on our corpus of news, to first do data reformatting. For each one article, we extract:

1. a ranking of the five most important "entities" discussed in the text, with related sentiment scores $\in [-1, +1]$,
2. a ranking of the five most important "concepts" discussed in the text, with related sentiment scores $\in [-1, +1]$,
3. the overall sentiment of the article, both as GPT sentiment score and from the basic VADER technique [69],
4. a one sentence summary of the article,
5. an abstract of the article up to twenty sentences long.

The keywords "entities" and "concepts" are chosen to either focus more on common and proper nouns, or adding a tuning on abstractions that describe categories of objects, respectively. Importantly, we employ *prompt engineering techniques* to generate more accurate and useful responses, and lower GPT temperature parameter to 0.2 to make the outputs more structured. We also iteratively refine our requests, and manually compare a sample of results to the actual news text, to assess the quality of GPT parsing. As an example, we propose here the main prompt used:

- *prompt0 = " I am an International Economist. Give me the top five entities mentioned in this text, from the most to least important. Reply as a numbered list using one or two words per entity at most. Next to each entity, provide the sentiment about this entity after a /-symbol strictly as a number between -1 and +1. Do not use words or brackets."*

Our hypothesis is that focusing on such keywords of news (that become our fundamental building blocks), and the inherent interconnections that we can define by their membership to one same article, will allow us to identify topics and narratives within news. Thus, we now introduce the relevant concepts from network analysis to model and analyse such interconnections.

3.3.2 Network Analysis

An undirected graph G is a pair $G = (V, E)$, where V is a set whose elements are called vertices or nodes, and E is a set of paired vertices that encapsulate related relationships. The elements $(u, v) \in E$, with $u, v \in V$, are called edges or links, and are characterised by some weight w_{uv}. In an unweighted graph, $w_{uv} = 1$, $\forall (u, v) \in E$, while a weighted graph generally has $w_{uv} \in \mathbb{R}^+$, $\forall (u, v) \in E$. If $(u, v) \notin E$, then $w_{uv} = 0$. The *degree* of node v is denoted by $deg(v)$ and computed as

$$deg(v) = \sum_{u \in V} w_{uv}, \tag{3.3}$$

which is often the first measure used to gauge the importance of the different nodes in a graph.

On the other hand, one further well-known measure of centrality (i.e. importance) of nodes is the *eigenvector centrality*. Relative importance scores are assigned to all nodes in the network, based on the concept that connections to high-scoring nodes contribute more to the score of the node in question. Thus, a high eigenvector score means that a node is connected to many nodes who themselves have high scores. If we first introduce the adjacency matrix of the graph G, which is $\mathbf{A} = (w_{uv})$, then the relative centrality score x_v of vertex v can be defined as

$$x_v = \frac{1}{\lambda}\Big(\sum_{u \in N(v)} x_u\Big) = \frac{1}{\lambda}\Big(\sum_{u \in V} w_{uv} \times x_u\Big), \tag{3.4}$$

where λ is a constant and $N(v)$ is the set of neighbours of node v. Equation (3.4) can be rewritten in vector form as

$$\mathbf{Ax} = \lambda \mathbf{x}, \tag{3.5}$$

for which usually there exist multiple values of λ that give a non-zero eigenvector solution. However, the additional requirement for all entries in the eigenvector to be non-negative implies (by the Perron–Frobenius theorem) that only the eigenvector associated with the greatest eigenvalue is the desired centrality measure. Then, the v^{th} component of the related eigenvector gives indeed the relative centrality score of the vertex v in the network.

Importantly, we highlight that it is common to compute the above measures of importance on the *giant component* of a graph. The giant component is the largest connected component of the graph (i.e. the subgraph with highest number of nodes), for which there is a path connecting each pair of nodes belonging to such subgraph.

Community detection - primer. Another important field of research within graph theory is *community detection*, i.e. clustering tightly connected groups of nodes, which is usually achieved by maximising the modularity Q of the partition proposed. Modularity Q is computed following [103] and measures the strength of division of a network into modules. Mathematically,

$$Q = \frac{1}{2W} \sum_{ij} \left[w_{ij} - \frac{deg(i) \times deg(j)}{2W} \right] \delta(c_i, c_j) \tag{3.6}$$

where W is the sum of weights of all edges in the graph and $\delta(c_i, c_j)$ is 1 if i, j are in the same community, otherwise 0. Importantly, modularity allows us to compare partitions of the same network, but it is by no means intended to be compared across different networks. One of the most popular algorithms for uncovering community structure is the Louvain algorithm [23]. The Louvain method consists of two phases, which are iteratively repeated until a local maximum of modularity is obtained. Starting from an initialised configuration in which each node is considered as a separate community, then the algorithm iterates through each node and evaluates the potential gain in modularity by moving it to a neighbouring community. If the gain is positive, then the node is indeed moved to the community that maximises the gain. The second phase implies instead aggregating the communities identified into super-nodes that generate a new and smaller network, on which we go back to apply the first step of the algorithm, and so on.

Community detection - fuzziness. Another important branch of community detection algorithms relies on spectral methods. Within such approaches, it is common to consider the connectivity of a graph via its Laplacian matrix $\mathbf{L}$, i.e.

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{3.7}$$

where $\mathbf{D}$ is the diagonal matrix summarising degrees of nodes. Since such matrix is positive semi-definite, then it can be decomposed into the product of a real matrix and its transpose. The new matrix can be interpreted as an embedding for nodes in the graph, or further clustered to highlight suggested communities. In [133], the authors consider the negative Laplacian $\mathbf{H} = -\mathbf{L}$ of a graph as an encryption of its local structure. In full,

$$H_{ij} = \begin{cases} w_{ij}, & \text{if } i \neq j \text{ and } (i,j) \in E \\ -deg(i), & i = j \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

Then, they apply to it a diffusion equation evolved by an exponential kernel. This is done in order to extract long-range relationships, and reads

$$\mathbf{K} = \exp \beta \mathbf{H} = \lim_{y \to \infty} \left(1 + \frac{\beta \mathbf{H}}{y}\right)^y, \tag{3.9}$$

$$s.t. \frac{d\mathbf{K}}{d\beta} = \mathbf{HK} \text{ with } \mathbf{K}(0) = \mathbb{1}, \tag{3.10}$$

where $\beta > 0$ controls the degree of diffusion and can be tuned by maximising the modularity score of the optimal partition consequently discovered. The resulting matrix $\mathbf{K}$ is symmetric and positive definite, and represents similarities among nodes. It can be then normalised as $K_{ij}^{norm} = \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}}$, and decomposed via Non-Negative Matrix Factorisation (NMF) as

$$\mathbf{K}^{norm} \approx \mathbf{VL}, \tag{3.11}$$

where $\mathbf{V}^{n \times k}, \mathbf{L}^{k \times n} \geq 0$. Matrix $\mathbf{V}$ is thus interpreted as a reduced features matrix [79] (one could equivalently take $\mathbf{L}^T$ due to the symmetry of $\mathbf{K}^{norm}$), whose rows refer to each one of the n nodes in the graph and give associated "membership degrees" to a number k of different clusters. By looking at the strongest membership degree for each node (i.e. highest entry per row), we can then deduce a strict partition of the graph into communities and compute the related modularity score. This is useful in order to tune β and k to values that maximise Q, and consequently deduce a final optimal clustering.

Importantly, we can also compare the 1^{st} and 2^{nd} largest probability values v_i^*, v_i^{**} of each row, to estimate how stable the label of each node is. This allows us to understand which nodes are best representatives of clusters, or more unstable and overlapping among multiple communities. Specifically, we compute the Stable Index S_i for each node i as

$$S_i = \frac{v_i^*}{v_i^{**}}, \tag{3.12}$$

which measures indeed the stability of the node in the community assigned. Care is needed in the analysis, since S_i can blow up if the denominator approaches 0.

3.3.3 Embeddings

As already made explicit, our work relies upon the application of both NLP and network analysis techniques. We extract main entities and concepts characteristic of news thanks to GPT, and these become our fundamental building block for network generation (where the related details will be fully explained in Section 3.4). To thoroughly analyse such landscape of data, we will also leverage upon

1. the word2vec [90] embedding techniques for words,
2. the node2vec [61] embedding techniques for nodes in a graph.

The former methodology considers sentences as directed subgraphs with nodes as words, and uses a shallow two-layer neural network (NN) to map each word to a unique vector. The result is that words sharing a common *context* in the corpus of sentences lie closer to each other. The latter approach focuses on embedding nodes into low-dimensional vector spaces by first using random walks to construct a network neighbourhood of every node in the network, and then optimising an objective function with network neighbourhoods as input.

Words embeddings - benchmark methodology. Word2vec [90] is a seminal method in NLP for word embedding, which operates on the premise that words with similar meanings share similar contextual usage. This approach employs one of two core models, i.e. either the Continuous Bag of Words (CBOW) or Skip-gram one. In CBOW, the algorithm predicts a target word from its surrounding context, while Skip-gram predicts context words given a target word. Utilising neural networks, these models generate and adjust word vectors to minimise prediction errors. The resulting high-dimensional vectors encode semantic relationships among words, and they are widely applied in various NLP tasks, such as sentiment analysis, text classification, and machine translation. Despite dating back to 2013, word2vec is indeed still a state-of-the-art methodology for the task of words (and bigrams...) embeddings, since e.g. GPT models do not easily provide such micro-level embeddings.

We can thus identify vectors for our entities extracted from news, by leveraging on pre-trained word2vec models available online, and cluster them. This provides us with a very first benchmark for topic identification and characterisation of their

evolution over weeks, before moving to study the effect of graph constructs. However, the best pre-trained models available to us tend to be based on old sets of text ($\sim$ up to 2015) and are not frequently updated. Since we do not have the resources to train our own model, this implies that we will surely miss vectors for meaningful words such as "Covid-19", "Bitcoin"..., and that the results cannot consequently be considered for more than initial exploratory investigations. In any case, the two pre-trained models we use to generate embeddings are:

1. Gensim Google News[3], which is a word embedding model based on Google news. A 300-dimensional vector representation is provided for approximately three million tokens ($\sim$ words).

2. FinText[4] Skip-gram [112], which is a financial word embedding based on Dow Jones Newswires Text News Feed Database. Again, a 300-dimensional representation is provided for almost three million tokens.

Node embeddings. Once we introduce an underlying graph structure among entities extracted from news, then a vector representation of the nodes ($\sim$ words) can be generated via the node2vec algorithm [61]. Taking inspiration from the word2vec idea of preserving knowledge of the common context windows of a word into its embedding, the node2vec algorithm learns a mapping for the set of nodes V to a low-dimensional feature space by maximising the likelihood of preserving network neighbourhoods of nodes. Given an undirected monopartite network $G = (V, E)$, node2vec learns a mapping function $f : V \to \mathbb{R}^d$ that produces a low-dimensional representations of nodes, where d is the number of dimensions of our feature space. Now, let $u \in V$ be a source node and $N_S(u) \subset V$ its neighbourhood generated by a sampling strategy S. The objective of node2vec is to preserve such network neighbourhoods, which can be formalised as

$$\max_f \sum_{u \in V} \log Pr(N_S(u)|f(u)), \tag{3.13}$$

and simplified to

$$\max_f \sum_{u \in V} \Big[-\log Z_u + \sum_{n \in N_S(u)} f(n) f(u) \Big] \tag{3.14}$$

with $Z_u = \sum_{j \in V} \exp(f(u)f(j))$ as the per-node partition function. The two assumptions needed for this step are

[3] https://github.com/RaRe-Technologies/gensim-data#models

[4] https://www.idsai.manchester.ac.uk/wp-content/uploads/sites/324/2022/06/Eghbal-Rahimikia.pdf

1. conditional independence of seeing neighbourhood nodes given the feature representation, i.e.
$$Pr(N_S(u)|f(u)) = \prod_{n \in N_S(u)} Pr(n|f(u)), \tag{3.15}$$

2. symmetry of the effect of source and neighbour nodes in feature space, incorporated by choosing
$$Pr(n|f(u)) = \frac{\exp(f(n)f(u))}{\sum_{j \in V} \exp(f(j)f(u))}. \tag{3.16}$$

Then, Z_u can be approximated by negative sampling (see [91]) and Eq. (3.14) is optimised using stochastic gradient descent (SGD).

The above objective allows for interesting flexibility in the definition of the neighbourhood of source node u, since this depends on the sampling strategy S. Consequently, one can tune the algorithm to either focus on sampling nodes in the same community, i.e. assessing homophily, or nodes with similar roles, i.e. looking at structural equivalence. For the former case, a random walk with Depth-First Sampling strategy (DFS) is initiated from u, which samples nodes at increasing distances from the source. In the latter case, nodes are instead chosen via a Breadth-First Sampling (BFS) strategy, which hovers closer to the source node. Of course, intermediates between these two options can also be defined. Formally, a second order biased random walk $u, v_1, ..., v_{l-1}$ of length l and $u, v_i \in V$ is generated following the distribution
$$P(v_i = z|v_{i-1} = y) = \begin{cases} \frac{\pi_{yz}}{D}, & \text{if } (y,z) \in E \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

where E is the set of edges, D a normalising constant and π_{yz} the unnormalised transition probability between nodes y, z. Assuming that the walk was on node x before reaching y, then π_{yz} is given by
$$\pi_{yz} = \begin{cases} w_{yz} \cdot \frac{1}{p_{emb}}, & \text{if } \varrho_{xz} = 0 \\ w_{yz} \cdot 1, & \text{if } \varrho_{xz} = 1 \\ w_{yz} \cdot \frac{1}{q_{emb}}, & \text{if } \varrho_{xz} = 2 \end{cases} \tag{3.18}$$

where w_{yz} is the weight of the edge between y and z. The shortest path distance ϱ_{xz} is instead computed between nodes x and z. The return parameter p_{emb} defines how likely it is to return to the previously visited node, while the in-out parameter q_{emb} controls how further away from the source node we are inclined to go. The choice $p_{emb} > max(q_{emb}, 1)$ ensures that we are less likely to sample an already visited node,

while $q_{emb} > 1$ biases the walk towards nodes closer to the origin one. Finally, we stress that the number of walks generated from each node should be optimised via hyperparameter tuning.

3.3.4 Logistic Regression

For completeness, we now briefly recall the logistic regression (also known as logit) statistical model. Logistic regression estimates the probability $P(X)$ of a binary event Y occurring or not, based on a given set $X = (X_1, ..., X_n)$ of n independent variables, i.e. predictors, for the instance under consideration. In our case, we can leverage such a framework to test which news' features might relate to moment of market dislocations (as per our definition on z-scores).

In logistic regression, probability $0 \leq P(X) \leq 1$ is defined via the logistic function

$$P(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_n X_n}}{1 + e^{\beta_0+\beta_1 X_1+...+\beta_n X_n}}, \tag{3.19}$$

which can be re-written in its logit version as

$$\log\Big(\frac{P(X)}{1 - P(X)}\Big) = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n. \tag{3.20}$$

In the above, $(\beta_0, ..., \beta_n)$ are unknown regression coefficients that need to be estimated. The method of maximum likelihood is generally chosen for the purpose, in which the intuition is to strongly penalise predictions that lie close to the most uncertain value of ~ 0.5. Mathematically, this results in estimating coefficients $(\hat{\beta}_0, ..., \hat{\beta}_n)$ such that the likelihood function

$$l(\hat{\beta}_0, \hat{\beta}_1) = \prod_{i:y_i=1} P(X_i) \prod_{i':y_{i'}=0} (1 - P(X_{i'})) \tag{3.21}$$

is indeed maximised. Then, predictions are made by estimating

$$\hat{P}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_1+...+\hat{\beta}_n X_n}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 X_1+...+\hat{\beta}_n X_n}}, \tag{3.22}$$

and mapping each $\hat{P}(X) > 0.5$ to outcome 1, while the remaining instances to outcome 0. Importantly, the tested predictors should not be correlated with each other to avoid having false relationships suggested by the regression. Then, we desire the estimated coefficients to be of high confidence, i.e. to be significant at least at the p-value < 0.05 level. Finally, it is often the case that prediction classes are strongly imbalanced, and it is needed to over-sample the minority class in the training set for better results.

In our case, we leverage the well-known SMOTE (Synthetic Minority Over-sampling Technique) algorithm [38] for the purpose, which generates synthetic perturbations of instances in the minority class to achieve indeed a more general decision region for the prediction of these less frequent events.

As a side note, we also mention that other machine learning techniques can be of course used for classification and prediction in a framework such as ours (e.g. classification trees, random forests...). However, logistic regression is indeed the final technique adopted in our work, since we simply aim to complete an initial assessment of whether news structure, especially modelled by our graph construct, can provide any enhancement to our understanding of markets.

3.4 Results

3.4.1 Word2vec benchmark

Our data correspond to a set of 197 weeks, with 110 ± 21 data points per week (i.e. average number of articles and its standard deviation). For every article in each week, we use basic NLP techniques to systematically parse its five main "entities" (and "concepts") with related sentiment, which arise as output of the GPT calls described in Section 3.3.1. This step requires dedicated care, since GPT outputs have some level of instability in their format. Importantly, we also implement a weighting scheme for the sentiment of each keyword k (i.e. either entity or concept), according to its rank of importance within the article. Each sentiment value is thus multiplied by a factor $\frac{1}{rank_k}$, where $rank_k \in \{1, 2, 3, 4, 5\}$.

We begin by considering both the GoogleNews and FinText pretrained word embeddings, and find the vector representative of each one of our keywords (entities or concepts) identified within news. Unfortunately, only $\sim$ 30% of entities are found to have an associated vector in the GoogleNews model, while this percentage increases to $\sim$ 50% with the FinText model. This is due to a mixture of such models being trained on obsolete data, and inadequate complexity in our keywords being sometimes composed by multiple words. Importantly, "concepts" are further found to be strongly higher in complexity and structure rather than "entities", and consequently of lower utility. Thus, our study will focus on "entities" as keywords for any following experiment.

We take advantage of the two selected vector embeddings to build a benchmark on the main topics addressed by our news articles. For each embedding model separately and each week of data, we complete the following steps:

1. We consider the vectors for the three most important entities (by the GPT ranking) in each article of the week. The full set of five entities is not taken, in order to produce more focused results due to the large percentage of missing vectors.

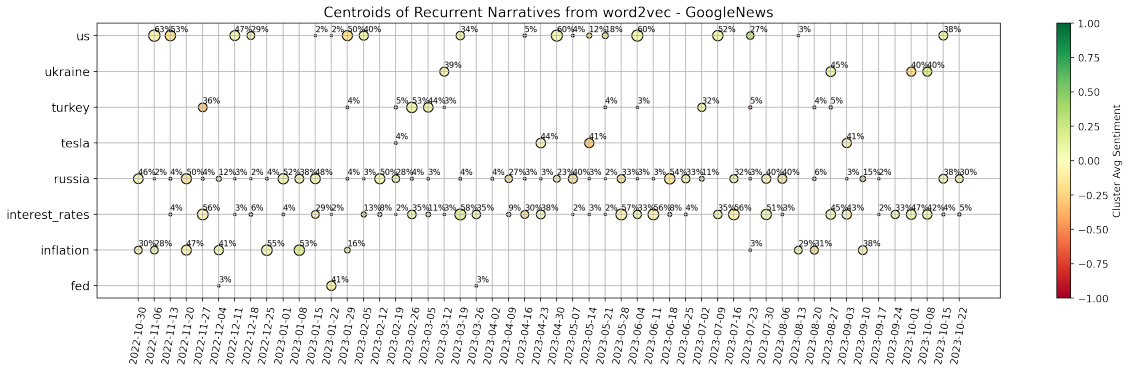
2. We cluster the retrieved vectors by performing hierarchical clustering, which iteratively aggregates input coordinates according to some measure of similarity or distance. Since our vectors lie on a 300-dimensional space, we use the "average" method (UPGMA algorithm) with cosine distance to complete this task (i.e. the distance between two clusters is computed as the average of all cosine distances between pairs of objects belonging to the two clusters).

3. From the above hierarchical clustering, we form flat clusters so that the original observations in each group have no greater a cophenetic distance than d_{coph}. This measure fixes the inter-group dissimilarity at which observations stop to be combined into their parent clusters on the hierarchical tree computed. In our case, we choose $d_{coph}^{GoogleNews} = 0.85$ and $d_{coph}^{FinText} = 0.80$ for the two models, in order achieve a small and tractable number of clusters (i.e. avoid too granular results).

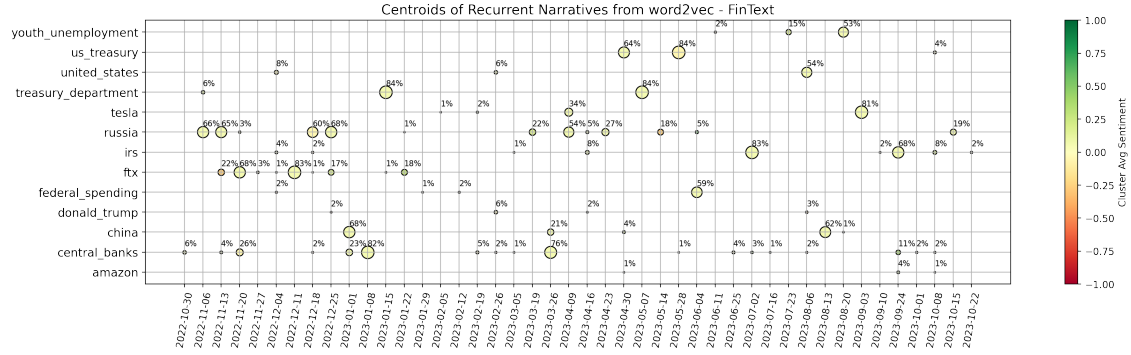
4. For each cluster, we consider all the words belonging to it and compute the related centroid by averaging their vectors. We save only clusters with more than one point belonging to them.

5. To map back centroids to words, we compute the cosine distance between the centroid vector and each one of our word vectors belonging to the cluster. The word that is found to lie closer to the centroid is taken as the representative of the cluster, and by extension as representative of the "topic" discussed within the cluster.

The results for both embedding models are shown in Fig. 3.2, for the latest one-year interval defined by weeks ending on 2022-10-30 and 2023-10-22. There, we can clearly notice some signs of the different training that the models were subject to. The GoogleNews model highlights two main narratives characteristic of the year we are considering, i.e. "Russia" (as for the war with Ukraine and associated consequences), and "interest rates" (due to continuous hiking of central banks). Interestingly, "inflation" is also a topic of main concern especially at the beginning of the data sample, and it is accurate to see that "Turkey" is signalled in February 2023 (when indeed a



(a) Results from word2vec GoogleNews pre-trained embeddings.



(b) Results from word2vec FinText pre-trained embeddings.

Figure 3.2: The proposed plots show the words that are closest to the centroids of clusters of information, which are systemically identified for each week. We plot only centroids that appear more than twice, to produce a clear and more focused representation of the main topics discussed within news. Each point is annotated with the percentage of entities (i.e. words) lying in the related cluster, and coloured by the average sentiment of such entities.

disastrous earthquake unfortunately happened). These results are in line with the fact that the model was trained on a corpus of news. On the other hand, the FinText model provides more disperse results. FinText is focused on financial language and related companies' data, and indeed we see that e.g. the FTX collapse of November 2022 is well identified. "Russia" and "central banks" are also hinted as recurrent and meaningful centroids of information, but the instability and noise within outcomes is significant. For completeness, Fig. 3.3 shows the Principal Component Analysis (PCA) projection of points for sample dates 2023-02-26 and 2023-10-08, where the vectors come from the GoogleNews embeddings. The first two PCA components explain together $\sim 20\%$ of the variance of the data, implying that the proposed representation must be interpreted with care. If we compare the centroids highlighted in Fig. 3.2 with

the current plot, we anyways see some reasonable (despite noisy) structure identified. However, it is clear that the results can be unstable and have difficulties in unravelling deeper shades of information.

Overall, we conclude that the available pre-trained word embeddings allow us to generate an interesting (despite very basic) initial benchmark on expected central words for the identification of topics in the given time range. But as highlighted, many problems come with this methodology. We will soon proceed to introducing our novel graph-based methodology for narratives identification, which further allows us to assess whether a topic exists on its own, or is interconnected with other topics. However, we first briefly provide some meaningful remarks on the choices taken to account for the sentiment within our news.

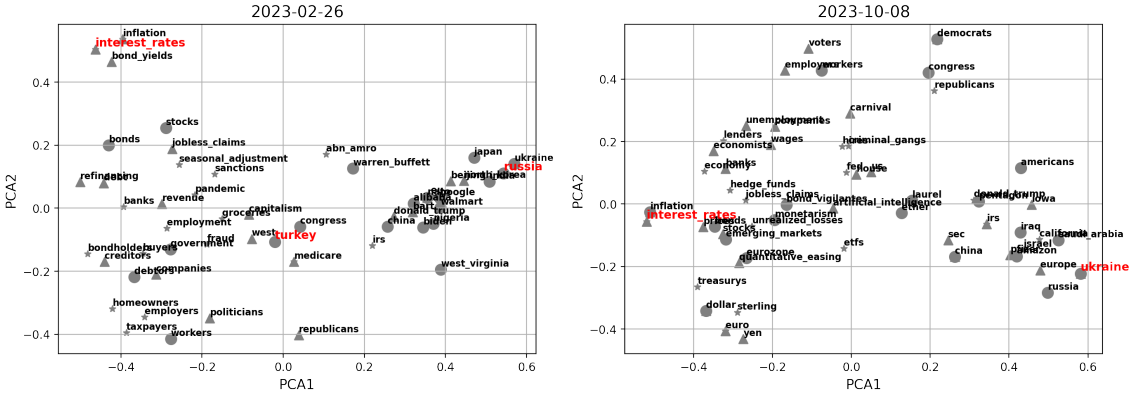
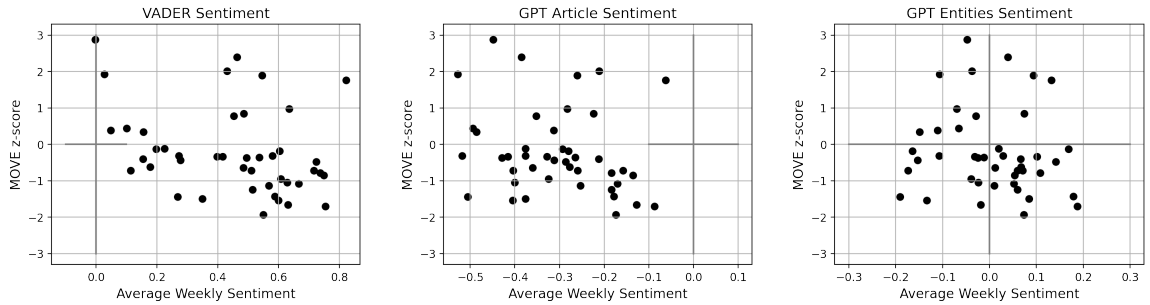


Figure 3.3: PCA projection of the representative words for two sample weeks, where vectors are recovered from GoogleNews embeddings. Words that are entities of rank 1, 2, or 3, are plotted as circles, triangles, or stars, respectively. We highlight in red the centroids suggested for such weeks from Fig. 3.2. Importantly, the variance explained by the two principal components is $\sim 20\%$, while our clustering considers all dimensions of the data.

3.4.2 Sentiment choice

When analysing news, it is of clear importance to be able to accurately assess their related sentiment, especially if one desires to connect them to the state of financial markets. We believe that simply considering the overall sentiment of an article is restrictive, and decide to leverage what GPT extracts as the sentiment surrounding each article's main entities. Thus, we provide now a brief motivating example that supports our approach.

Our latest one year of data, i.e from the week ending on 2022-10-30 to 2023-10-22, concerns a strongly focused period of high (but oscillating) concern on themes such as



(a) Positive bias detected. (b) Negative bias detected. (c) Diagonal relationship.

Figure 3.4: Weekly MOVE Index z-scores against average sentiment of news on each week, for subsets of articles relevant to the bond market. The primitive VADER sentiment analysis technique shows a positive bias, while GPT sentiment calculated for whole articles averages to an opposite negative bias. Since the period considered is not uniformly negative, we are instead satisfied with our proposed entities-based sentiment. This is indeed much more symmetric, despite carries a tail of noise in the third quadrant that we believe spills from the overall negative GPT bias.

interest rates and inflation. Thus, we use such data in the current experiment. We subset related articles to the ones where the first or second most important entities are in the set {"interest rates", "inflation", "Federal Reserve", "European Central Bank", "Bank of England"}. Then, we consider our weekly z-scores computed for only the MOVE Index, which indeed focuses on the bond market. For each week with more than 15 related articles, we average the sentiment (always weighted by rank) of the main entities extracted for the article, and then average over articles. Due to the weighting applied, the maximum mean achievable is $\frac{1+0.8+0.6+0.4+0.2}{5} = 0.6$, and thus we multiply each result by a factor 0.6^{-1}. Similarly, we also average over Vader and overall GPT sentiment on the articles for each week, and visualise the results in connection to MOVE z-scores. Figure 3.4 shows the trends discovered. VADER is one of the most primitive sentiment analysis techniques, and indeed it reveals a strong positive bias. On the other hand, it is interesting to see that GPT tends to constantly assign an overall negative sentiment to the articles related to our theme under investigations, despite this is not representative of the oscillating level of concern in the actual markets and economy. Thus, we are satisfied to see that our weighted GPT sentiment calculated from entities occupies mainly the second and fourth quadrant of the plot, as desired. There is still a tail of negative sentiment with improving bond market (i.e. negative MOVE z-score), but we explain this as a spillover from the overall negative GPT bias.

3.4.3 Graph construction and initial measures

We can now proceed to introducing our novel graph-based methodology for narratives identification, which further allows us to assess whether a topic exists on its own, or is interconnected with other topics. As already hinted, the building blocks of our methodology are the ranked entities extracted by GPT from each article. For each week of news available, we thus generate a related representative graph $G = (V, E)$ as follows:

1. The set of nodes V is constructed from the union of all entities extracted from the articles $a \in A$ of the week. An attribute is also assigned to each node, which is the average of the (rank-weighted) sentiment extracted by GPT around such entity across articles.

2. For each pair of entities $u, v \in V$ that belong to the same article a, an edge $(u, v) \in E$ is created. Such edge has a weight $w_{a,uv}$ given by the product of the inverse $rank$ of the two entities in the article, i.e.

$$w_{a,uv} = \frac{1}{rank_{a,u}} \times \frac{1}{rank_{a,v}}. \tag{3.23}$$

   This choice is taken to better account for the strength of connection among words, implied from their importance and focus within the article.

3. Multiple edges across the same two nodes are aggregated, and the associated weights summed to give the final weight $w_{uv} = \sum_a w_{a,uv}$ of the link among each pair of nodes u and v. A threshold is then applied, and all edges with $w_{uv} \leq \frac{1}{2} \times \frac{1}{3}$ dropped. Explicitly, we are requiring that the sum of weights among each pair of nodes is strictly above the base weight generated among two entities being second and third in the ranking of a same article. While we experimented with multiple different options, it was found that this choice allows us to maintain an interesting structure of interconnections in the graph, while lowering the noise of data and the symmetries intrinsic to our first part of the approach. The related degree distributions are also reasonable.

4. Finally, we save the giant component of the graph.

For completeness to the above, we also show in Fig. 3.5 both the ratio of the size of the giant component versus the total initial number of nodes and the average clustering coefficient of the network, for each week. The latter is computed as the

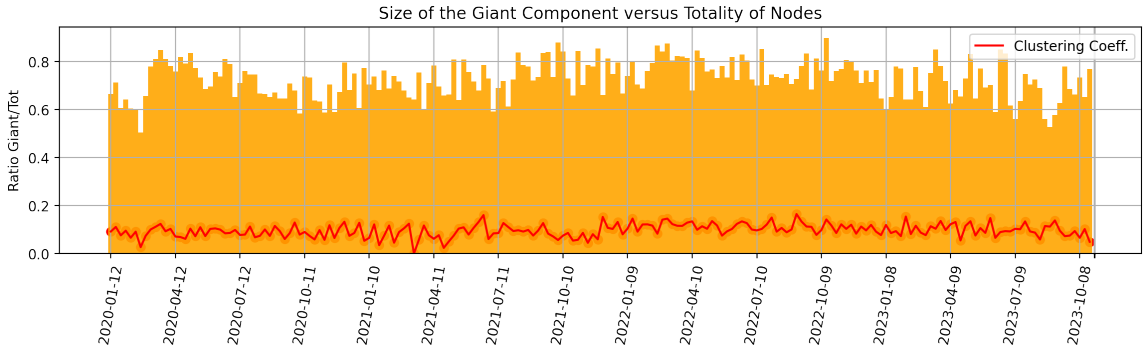


Figure 3.5: For each week of data, we measure the percentage of nodes that lie in the giant component of the related graph and report it in orange. In red, its average clustering coefficient is also depicted.

average among the local clustering coefficient for each node in the graph, which is the proportion of the number of links between a node's neighbourhood divided by the number of links that could possibly exist between them.

The giant components tends to encompass the large majority of the nodes (being of size $\sim 200 - 300$ nodes), meaning that we maintain significant amount of information, nevertheless with some intrinsic variability. When the second largest connected component has higher than average number of nodes, then it still tends to have only $\sim 20 - 30$ nodes. For the sake of curiosity, we looked into a few such scenarios and report here an explicative example. For the week ending on 2022-12-04, the second largest connected component of the related graph is made of the following 21 nodes:

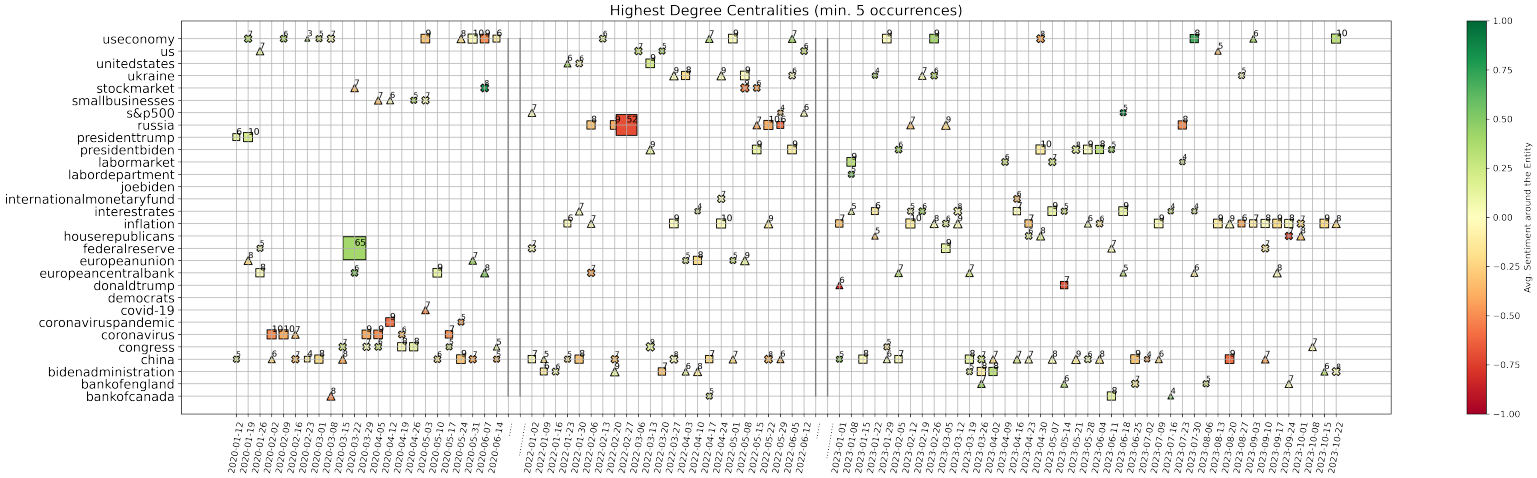
- *sam bankman-fried, us bankruptcy courts, senate agriculture committee, john j ray iii, securities and exchange commission, cryptocurrency markets, alameda research, us trustee andrew vara, ftx, commodity futures trading commission, crypto lenders and hedge funds, valar ventures, us justice department, ledger x, blockfi, monsur hussain, sec and cftc, terrausd, newyork prosecutors and sec, three arrows capital, lehman brothers.*

Clearly, this is a cluster of information related to the bankruptcy of the crypto exchange FTX in November, and its contagion to BlockFi Three Arrows Capital. However, this is disconnected from the giant component. Interestingly, we can thus inspect that e.g. the Securities and Exchange Commission (SEC) has no major other meaningful participation in the news, since otherwise the cluster would be connected to the giant component via that entity.

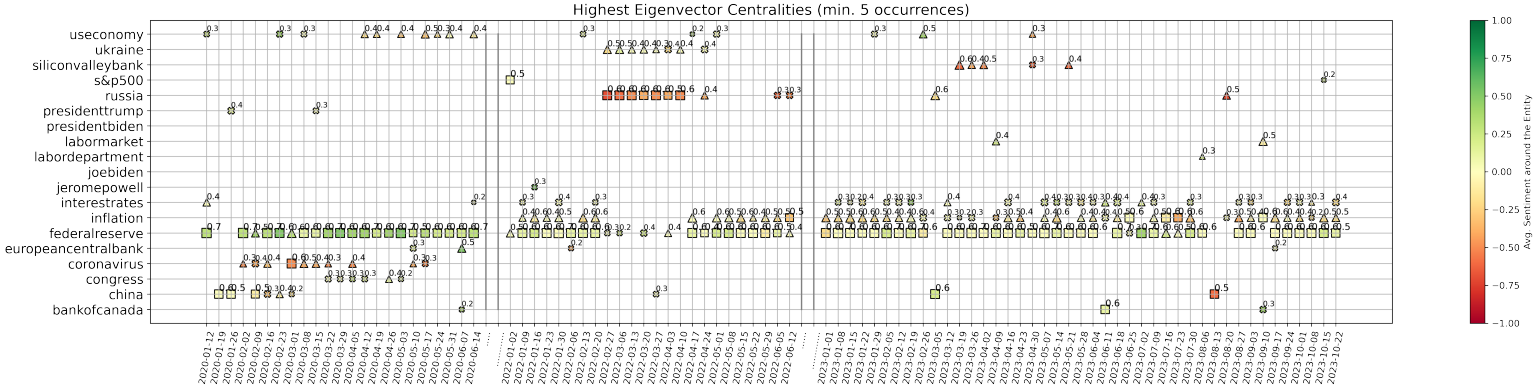
Thus, our approach allows us to focus on the interconnectedness of topics via the giant component of each graph, and to consequently try to extract the main narratives

among weeks and their interrelations. Other "large" connected components can be considered to understand disconnected topics addressed in the news, but will not be of main relevance for the *global* state of the market. The average clustering coefficient of the giant component is then seen to simply oscillate around the value of 0.1, and thus does not provide strong direct insights. On the other hand, the degree and eigenvector centrality of nodes at different weeks will provide valuable information.

Centralities. We compute the weighted degree $deg(v)$ and eigenvector centrality $eig(v)$ for every node v in the giant component of the graph describing each week. Then, we focus on the three nodes with highest $deg(v)$ and $eig(v)$ for the given point in time, and show them in Figs. 3.6a and 3.6b, for the two measures respectively.



(a) Degree centralities.



(b) Eigenvector centralities.

Figure 3.6: Most important entities in the graph of each week, from the degree of nodes or their eigenvector centrality. Degree centrality focuses on the size of neighbourhood of a node (i.e. the topic is addressed from multiple perspectives), while eigenvector centrality highlights the nodes close to other important nodes (i.e. measures the interconnectedness and influence in the global environment). For each week, we plot the 1st, 2nd, and 3rd most important entities as squares, triangles, and crosses, respectively. We size and annotate each point according to the magnitude of importance (rounded), and colour-code it by the average sentiment around that entity. For clarity, we show only entities that appear as highly important at least five times over our sample.

In particular, we focus on entities that are recurrently important, i.e. we subset to entities that have highest centrality values for at least five weeks of our sample. For each week, we then plot the first, second, and third most important entities with the shape of a square, triangle, and cross, respectively. We size and annotate each point according to the actual value of the importance measure (rounded), and colour-code it according to the average sentiment around that entity. When we look at the nodes with highest degree, we are selecting the entities around which many news are focused. On the other hand, nodes with high eigenvector centrality are entities connected to other important entities, meaning entities lying at the center of the global net of interconnections among news.

It is thus interesting to then analyse the different results arising from these two measures. We start from Fig. 3.6b, which shows snapshots of the evolution of nodes with highest eigenvector centrality. Clearly, "Federal Reserve", "inflation", and "interest rates" play the major role in our net of news for the latest one year and a half. This is expected since such period is characterised by broad discussions about the persistently strong inflation, and the constant rate hikes of Central Banks. However, "Silicon Valley Bank" (SVB) also acquires significant importance at the end of March 2023, after it indeed collapsed on March 10th, 2023. The reason why this event has noteworthy eigenvector centrality (while it is not signalled by the degree centrality) is that SVB was the largest bank to fail since Washington Mutual closed its doors amid the financial crisis of 2008, on top of a moment of already strong fear of incoming recession in the U.S. Thus, such shock could have spread and affected current themes of discussion, meaning that our modelled interconnectedness of news was able to capture and highlight these broader concerns. Going further back in time, we see the strong concern around "Russia" and "Ukraine" at the beginning of 2022, when indeed the former invaded the latter. And clearly, "China" and the "Coronavirus" are signalled at the beginning of 2020, when the Covid-19 pandemic originated. Such entities are however not further signalled with the passing of time, meaning that the focus shifted on the consequences of such crisis rather than such topic itself.

We now move to the degree centralities shown in Fig. 3.6a. "Inflation", and "interest rates" play again an important role, as expected. However, we have now more variability due to less interconnected topics but with high surrounding discussion. A negative shock related to "Russia" is clearly proposed in conjunction with the invasion of Ukraine, while a strongly positive one signals the pandemic stimulus approved by the Federal Reserve in March 2020. Interestingly, "China" appears quite constantly during our full sample of data, with e.g. a point of stronger negative

sentiment and concern on the week ending on 2023-08-20. This is when Evergrande filed for U.S. bankruptcy protection as China economic fears mounted, while China also unexpectedly lowered several key interest rates earlier that week. Finally, we can further notice that "President Biden" and its Administration are some times highlighted, which is sensible due to the U.S. focus of our corpus. On top of that, data show indeed that e.g. "Donald Trump" carries strong negative sentiment the final weeks of 2022, when the January 6 committee decided he should indeed be charged with crimes related to the assault on the U.S. Capitol in 2021.

3.4.4 Community detection

The degree and eigenvector centralities of nodes in a graph can point to a concise sample of nodes of major interest, but they do not provide deeper insights on the structure and information within the totality of nodes. We believe that considering the problem of community detection on the proposed graphs will allow us to extract more insights on the topics characteristic of each week of news, and on the evolution of the associated narratives. We slightly distinguish between the words "topic" and "narrative", and consider the former as the broad category or label of a series of events, while the latter as the surrounding information that can evolve over time. As an example, we would refer to the "Covid pandemic" as a topic, while its narrative would evolve from the initial outbreaks, to the development of vaccines, to the posterior implications of the monetary policy adopted...

We do community detection on the graph representative of each week, both via the classic Louvain methodology and the fuzzy spectral method (Section 3.3.2). In this way, we identify clusters of nodes highly interconnected, which we then analyse. The Louvain method automatically finds the optimal number of communities k^{Louv} for each week. On the other end, we need to define the desired number k^{fuz} of output communities when applying the fuzzy spectral method. To find the optimal k^{fuz}, we first compute the modularity Q of Eq. (3.6) of the strict partition arising from fuzzy community detection on each given graph, for possible number of communities $k \in \{2, 3, ..., 14\}$. Then, we automatically choose the *knee* of each resulting plot, i.e. the point after which Q does not significantly increase with further increases in k. Figure 3.7 shows an example of such plot, for the week ending on 2022-11-20. Since the result is based on NMF, we always compute the clustering from multiple random seeds, and then take the mode of the suggested knees (despite the trends tend to be uniform, as the plot hints). Of course, we also investigate the diffusion parameter β in

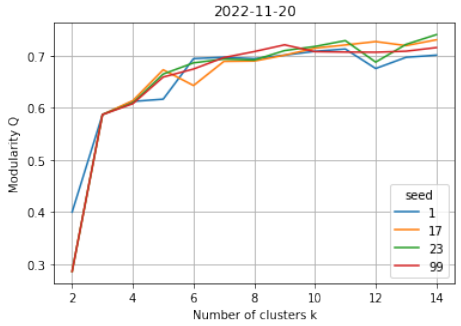


Figure 3.7: Looking for the optimal number of communities k^{fuz} for the week 2022-11-20, to input when computing fuzzy community detection. Multiple random seeds are used to test the stability of the results against possible perturbations during NMF. A knee $k^{fuz} = 5$ gets here selected.

parallel, and fix it to $\beta = 1$. Then, we save the stability index of each node associated to the optimal partition of each graph.

For our set of graphs of interest, the average k^{Louv} is 17 (with standard deviation ± 3), while the average k^{fuz} is 5 (with standard deviation ± 1). This means that the diffusion kernel employed in the fuzzy spectral method, which captures long-range relationships between nodes, allows us to achieve more tractable clusterings. These also encapsulate broader structures within the networks, and will be preferred against Louvain communities. For the sake of completeness, we compute the Adjusted Rank Index (ARI) among the clusterings suggested by the two methodologies, for each week. The ARI is a similarity measure between two clusterings, whose specifications we now briefly introduce. Let $C^{(1)} = \{C_1^{(1)}, ..., C_k^{(1)}\}$ and $C^{(2)} = \{C_1^{(2)}, ..., C_k^{(2)}\}$ be two partitions in k clusters. The ARI reads

$$ARI = \frac{\sum_{k_i,k_j} \binom{\phi_{k_i,k_j}}{2} - \left[\sum_{k_i} \binom{\phi_{k_i}^{(1)}}{2} \sum_{k_j} \binom{\phi_{k_j}^{(2)}}{2}\right] / \binom{|V|}{2}}{\left[\sum_{k_i} \binom{\phi_{k_i}^{(1)}}{2} + \sum_{k_j} \binom{\phi_{k_j}^{(2)}}{2}\right] / 2 - \left[\sum_{k_i} \binom{\phi_{k_i}^{(1)}}{2} \sum_{k_j} \binom{\phi_{k_j}^{(2)}}{2}\right] / \binom{|V|}{2}}, \tag{3.24}$$

and takes a maximum value of 1 if $C^{(1)} = C^{(2)}$. In the above, $|V|$ is the number of nodes in the graph to cluster, and $\phi_{k_i,k_j} = |C_{k_i}^{(1)} \cap C_{k_j}^{(2)}|$ with $k_i, k_j \in \{1, ..., k\}$. Then, $\phi_{k_i}^{(1)} = |C_{k_i}^{(1)}|$ and $\phi_{k_i}^{(2)} = |C_{k_i}^{(2)}|$. The ARI between partitions generated by our Louvain and fuzzy methodologies is 0.37 ± 0.11 on average, meaning that there is a good level of agreement between the results, and increasing our confidence in focusing on the fuzzy spectral methodology from this point onward.

Fuzziness. Importantly, we have been considering the strict partition generated by the introduced spectral method so far, but have not yet focused on its fuzziness

component and stability of nodes. This is indeed leveraged upon now, since our hypothesis is that the communities highlighted by such method can be directly mapped to the main topics discussed within news. Thus, dropping the most unstable nodes will allow us to achieve a clearer and cleaner view. For each week, we proceed as follows:

1. For each community identified, we extract the list of nodes belonging to it.

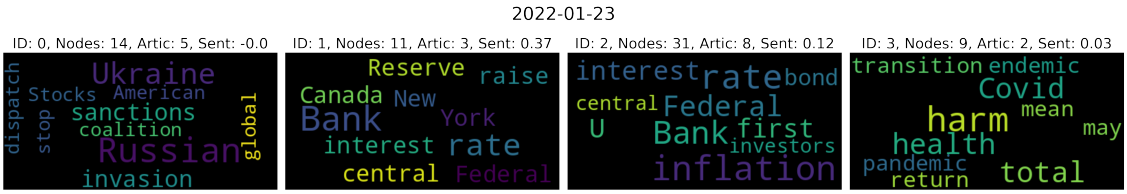
2. For each related node i, we check whether $S_i \leq 2$ (i.e. if the level of membership in the most likely community is less than twice the level of membership to the second one). If the relationship is satisfied, then we label the node as *unstable*. Importantly, such threshold is chosen by looking at the distribution of stability indices over nodes for a sample of plots.

3. Then, we take all the news published that week. For each one node, we consider all articles for which the node is one of its five main entities. We further check whether all the three main entities of the article are in the nodes of the community under investigation, but not in the associated set of unstable nodes. The articles that satisfy all conditions are kept.

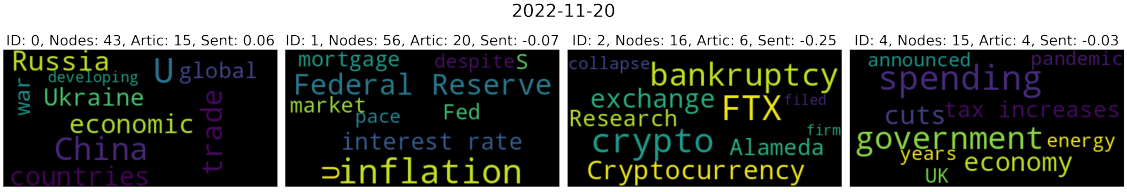
4. For each final set of articles representative of each community of the week, we save the related information (i.e. GPT-generated summary and GPT-generated abstract).

To summarise the above, we focus on constructing a set of highly representative articles for each community identified within the graph describing each week of data. We are very stringent with our stability requirements, and indeed drop on average 16 articles per community. However, this allows us to find the articles at the heart of each community, which enable the most effective downstream analyses. We then concatenate all related GPT summaries (and similarly abstracts) for each community, and visualise them via word-clouds[5] to see whether they produce coherent topics. In parallel, we also feed such joint data into GPT and ask it to produce a newly associated summary, which should then allow us to easily interpret each topic (and narrative) of the week. The examples proposed in Fig. 3.8 aim to provide initial evidence of the accomplishments achieved by our methodology, where we show word-clouds of topics found for the weeks ending on 2022-01-23, 2022-11-20, and 2023-08-20. This step also allows us to have an intermediate assessment of the coherence of articles

[5]A word-cloud is a collection of words depicted in different sizes and strengths to indicate the frequency, and so importance, of such word in the text.



(a) Word-clouds representative of the topics identified for the week ending on 2022-01-23.



(b) Word-clouds representative of the topics identified for the week ending on 2022-11-20.

(c) Word-clouds representative of the topics identified for the week ending on 2023-08-20.

Figure 3.8: Word-clouds for the topics within three sample weeks of news. If no articles are seen to belong to a topic-community after our stability analyses, then we drop the related empty word-cloud. This is the reason why some IDs are missing.

mapped to each group, and check that the results from GPT summarisation are not inaccurate or due to hallucination. The related summaries of joint summaries are reported in Tables 3.2, 3.3, and 3.4, respectively, and we remark that summaries on abstracts produce highly similar results. Importantly, we underline that if no articles belong to a community after imposing our stability requirements, then clearly neither a word-cloud is shown nor a summary reported.

By studying the generated topics via word-clouds and proposed summaries, we find interpretable results that accurately map to the broad major events described by the data for each related week. Clusters for week 2022-11-20 are a nice example. On the other hand, we also see ability to extract the precursor worries of a Russian invasion of Ukraine from the findings of week 2022-01-23. This could be of great use for a thorough and systematic study of early signs of crises with more historical

data. Moreover, one could compare the time of appearance of such signs across results computed from different corpora, e.g. collected by news from Journals of different countries. Finally, we show the clusters for week 2023-08-20, since this is one of our latest occurrences of a market dislocation event (refer to Fig. 3.1). Interestingly, we see that many clusters are identified depicting concerns across themes of discussion. These different topics still need to be part of the giant component of our graph, implying that we have an instance of market turbulence related to high entropy of discussions within news that must be somehow more broadly interconnected.

Table 3.2: GPT summaries on the joint summaries of articles belonging to each community for week ending on 2022-01-23, after our stability filtering

ID	GPT Summary of Summaries
0	China's stocks rose and Hong Kong's stocks fell after the government cut interest rates, while the US is preparing financial sanctions on pro-Russian agents in Ukraine to deter Russia from invading its neighbor.
1	US and Canada are considering raising interest rates, while China has cut its benchmark lending rates to support its economy, and the Federal Reserve Bank of New York has appointed a new official to oversee financial-market operations.
2	Tax revenue is up, inflation is rising, central banks have differing views on interest rates, and US companies may struggle to generate high profits in the coming year.
3	The transition of Covid-19 from pandemic to endemic could still have lasting effects on health and the economy, and policies should aim to minimize harm from both infection and prevention measures.

Table 3.3: GPT summaries on the joint summaries of articles belonging to each community for week ending on 2022-11-20, after our stability filtering

ID	GPT Summary of Summaries
0	The G-20 summit addressed global economic challenges, including the war in Ukraine and the weaponization of food production, while the US imposed sanctions on individuals and companies linked to Russia, and new intelligence showed that parts in Iranian drones downed in Ukraine were manufactured by companies in allied nations.
1	Investors are optimistic about decreasing inflation, but earnings are becoming a threat as Wall Street analysts slash profit forecasts, while the Federal Reserve warns of the need to raise interest rates to control inflation, and the housing market cools down due to a lack of buyers for mortgage bonds.
2	Cryptocurrency firm FTX and its investment arm Alameda Research have filed for bankruptcy, leaving customers facing potential losses in an unregulated sector, with details still scarce.
3	NA
4	The UK government has announced tax increases and spending cuts to reduce government debt relative to the economy, becoming the first major Western economy to limit spending growth after years of fiscal stimulus during the pandemic and recent energy subsidies.

Table 3.4: GPT summaries on the joint summaries of articles belonging to each community for week ending on 2023-08-20, after our stability filtering

ID	GPT Summary of Summaries
0	The US is in talks to increase alternative export routes for Ukrainian grain after Russia pulled out of an agreement, with a US-backed plan involving increasing capacity for Ukraine to export via the Danube River, while concerns have been raised over the safety and environmental impact of Turkey-based Beks Ship Management's transportation of Russian oil.
1	The Biden Administration is changing its analytical methods to make it easier to impose new rules while disguising their cost, resulting in increased regulatory costs on the economy.
2	Russia's ruble has fallen to its weakest level in over a year due to Western sanctions and the war in Ukraine, prompting an emergency meeting by the central bank, but analysts suggest that factors driving down the currency are largely out of their control.
3	China's interest rate cut to boost the economy is putting pressure on the currency and causing declines in major stock indexes.
4	Big American companies rooted in China are reporting weaker sales and turning to other countries for imports due to China's deepening economic slump.
5	NA
6	UK retail sales fell more than expected in July due to bad weather and economic issues, causing UK gilt yields to drop and potentially reducing the need for Bank of England interest-rate rises to combat inflation.
7	Inflation is decreasing in the US, but rising energy and food prices may cause turbulence, while Canada's inflation rose unexpectedly and may lead to a rate increase, and Latin American central banks are cutting interest rates as inflation eases.

Evolution of narratives. For each week of data, we found an optimised partition of its graph's nodes into communities with interpretable related economic themes. This is however a static representation of topics, which we now try to extend by linking and tracking such clusters over time. In a framework like ours, the most direct way to proceed is to gauge the similarity of our topics (i.e. sets of nodes) by computing the overlap between any two such sets of nodes, normalised by the union of considered nodes to increase the significance of the measure. This is indeed one of the most common approaches in the literature of network analysis for single communities' comparison [47], with related generalisations. Thus, we consider each pair of consecutive weeks, and compute the related Jaccard Index $J(C_{i,1}, C_{j,2})$,

$$J(C_{i,1}, C_{j,2}) = \frac{|C_{i,1} \cap C_{j,2}|}{|C_{i,1} \cup C_{j,2}|}, \tag{3.25}$$

for any pair of respective communities that identify topics. We call $C_{i,1}$ and $C_{j,2}$ the sets of nodes that belong to any one community identified within the first and

second week, respectively. The Jaccard Index allows us to compute a measure of similarity between communities from the overlap of their nodes, meaning that we can assess which clusters (and so topics) are the most similar over consecutive weeks and link them. We also restrict our results to have $|C_{i,1} \cap C_{j,2}| > 1$, to increase the associated significance. Figure 3.9 shows a sample of matrices that link topics from their Jaccard similarity, for a series of pairs of consecutive weeks. The simplest way to track a narrative is thus to start from its first acknowledged community, and "jump" to communities at consequent points in time by following the series of highest similarity scores. In our case, we resolve multiple mappings among communities by considering only the maximum entry per column and row in the proposed similarity matrices, but this approach can be of course improved.

As an illustrative case, community 0 in Fig. 3.9 at week 2022-01-16 is most similar to community 2 at week 2022-01-23, which is then most similar to community 0 at week 2022-01-30, and the latter to community 0 again at week 2022-02-06, and so on. Of course, if no mapping is at some point found, then the narrative "has broken". In Figures 3.10a and 3.10b, we respectively track both the shown topics 0 and 1 with the simple methodology proposed. We plot the nodes that are found to overlap for the most similar communities between each two consecutive weeks, until a mapping persists. It is direct to see that narrative 0 relates to inflation, interest rates, and the behaviour of Central Banks, while narrative 1 concerns Russia and the themes surrounding its war with Ukraine. In parallel, we also deploy a slightly different approach and just save the communities to which a chosen keyword belongs over time. In this way, we try to better investigate the surrounding evolving narrative. As an example, Fig. 3.11 shows the overlap of entities (generally plotted as dots) belonging

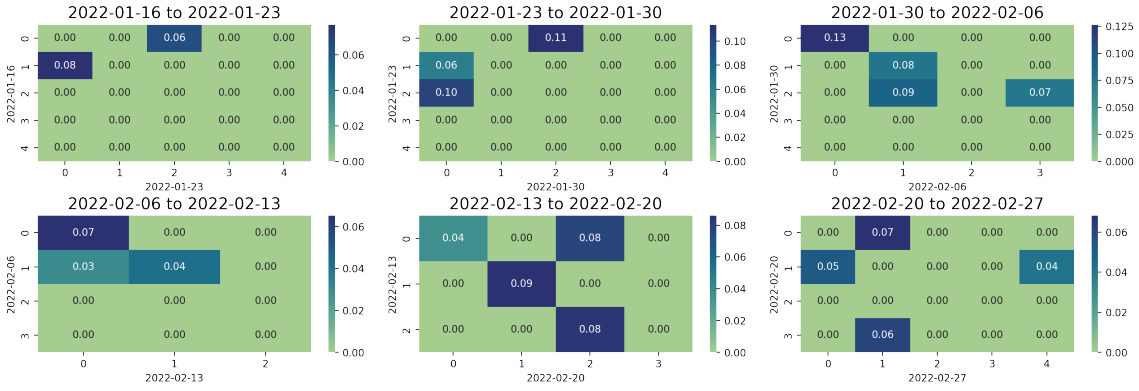
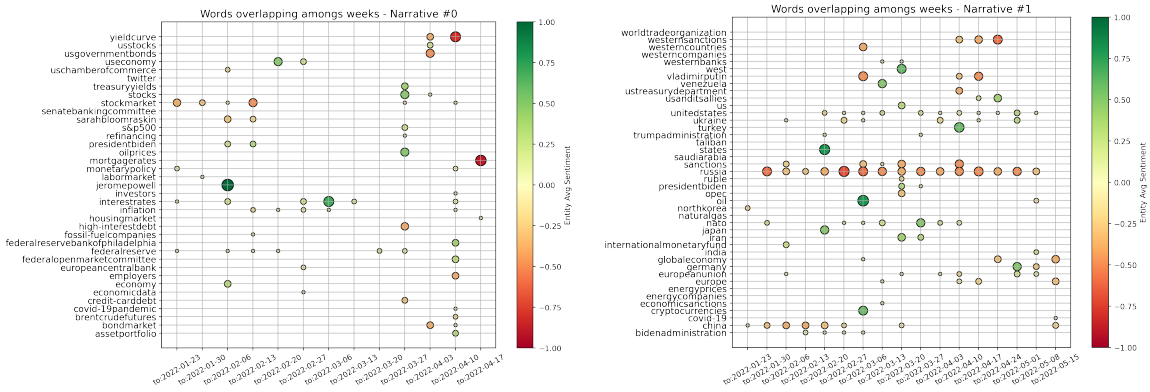


Figure 3.9: Matrices of Jaccard similarity between nodes in communities for consecutive weeks over time. In this way, we investigate potential links for topics (and narratives) over time.



(a) Tracking Narrative 0 from 2022-01-16. (b) Tracking Narrative 1 from 2022-01-16.

Figure 3.10: We test whether we can track interpretable narratives over time, via the similarity of communities of nodes in each two consecutive weeks as in Fig. 3.9. Narrative 0 is seen to relate to inflation and interest rates concerns, while Narrative 1 focuses on Russia.

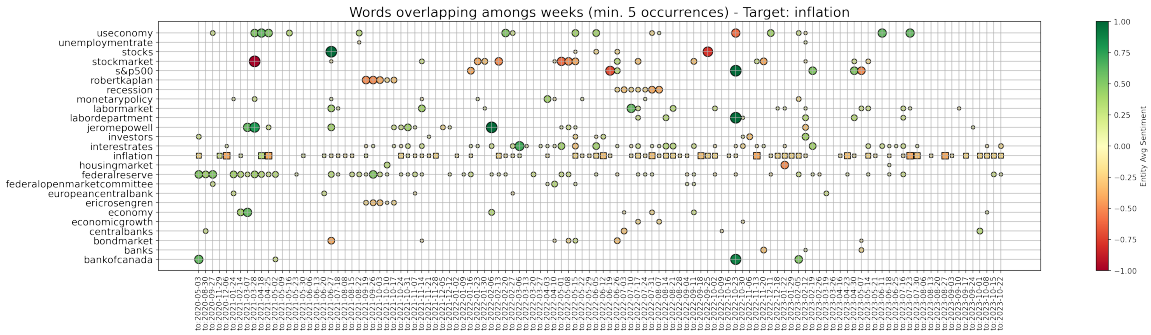


Figure 3.11: We define inflation as a keyword of interest, and track the recurrent entities belonging to its same community for consecutive weeks over time.

to the same communities as the word "inflation" (plotted as squares) over consecutive weeks. New characters are highlighted, such as Eric Rosengren and Rober Kaplan, and an important period of concern is signalled around July 2022 and the word "recession". Indeed, concerns about inflation, interest rates, and the fall of Gross Domestic Product (GDP) in the first two quarters of 2022, led to heightening recession fears during the month of July.

3.4.5 News and financial markets dislocations

Thanks to the analyses just completed, we have supporting evidence on the ability of our graph constructs to advance in the task of topic detection and narrative characterisation. However, we now desire to test whether information on such structures of news allows us to unravel novel insights on broad market dislocations. As introduced in Section

3.2.2, we describe the state of the market by the z-scores of our volatility indices. Here, we then define a market dislocation as a week when all four of our z-scores are strictly positive, and their average is above 0.5. Such weeks are identified with a +1 label, while all other periods with a 0 label, implying that we now have a binary variable to use as target of a logistic regression model (see Section 3.3.4). For the sake of clarity, Fig. 3.12 shows our points of interest.

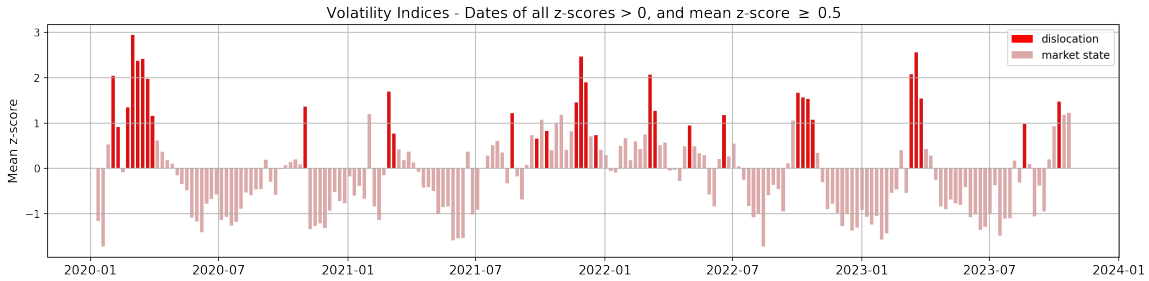


Figure 3.12: In bright red, we depict dates in which all our four z-scores are strictly positive, and their mean is also above 0.5. These are our dislocation dates of reference (to which we assign label +1). On the other hand, the simple average on the four z-scores is plotted in faded red. It is clear that dislocation dates are an under-sampled category, which we expand via the SMOTE technique.

To run the suggested logistic regression, we need to build a set of features that characterise each week. Then, we can either map such features to contemporaneous or next states of the market, to respectively investigate relationships between news and either unfolding or incoming dislocations. The features that we consider can be seen to belong to three different categories, namely:

1. *Market features.* Features on the current state of the market (used in the case of predictive logistic regression, and as a benchmark). These are the current average of our four z-scores of volatility indices, and the difference of such value from the mean of the previous week.

2. *News features.* Features based on the raw corpus of news available.

3. *Graph(+) features.* Features extracted from the set of graphs that we built to capture the structure and interconnectedness of news within weeks. We also leverage node2vec (n2v) embedding methodology, as soon fully described.

Due to the limited number of data points (i.e. weeks) available, we build and test only features that we believe to be the most significant and meaningful for each category. Indeed, it is a common proxy to allow at least ~ 20 outcomes for each independent

variable tested. Also, we check the correlation among each pair of variables, to drop features that would cause problems of multicollinearity and invalidate results of the regression. Table 3.5 summarises our final set of 10 tested features and chosen related tags, while Fig. 3.13 shows their correlation matrix from Pearson's test. As desired, all the retrieved features show either small or negligible correlation among each other.

Table 3.5: Final set of features evaluated within our logistic regression models. Features are tested to unravel relationships between news and financial market dislocations

Class	Name	Description
Market	z-vols	Average of our four volatility indices' z-scores (i.e. VIX, MOVE, VIX FX, and MRI).
Market	z-volsD	Difference of average z-scores between the current week and the previous one.
News	N-avgSent	Average sentiment of news in a week, where each article's sentiment is computed as the weighted mean of entities' sentiment.
News	N-stdSent	Standard deviation of the above-mentioned sentiment of news in a week.
Graph	giantRatio	Ratio between the size of the giant component and the total number of possible nodes.
Graph	clustCoeff	Average clustering coefficient of the graph.
Graph	eigFirst	Highest value of eigenvector centrality of nodes in the graph.
Graph	eigRatio	Ratio between the first and second highest eigenvector centralities.
Graph	comm	Optimal number of communities as identified from fuzzy community detection.
Graph+	n2v-entropy	Proxy for the total entropy of node2vec embeddings lying in their multi-dimensional space, by leveraging the Kullback–Leibler divergence measure.

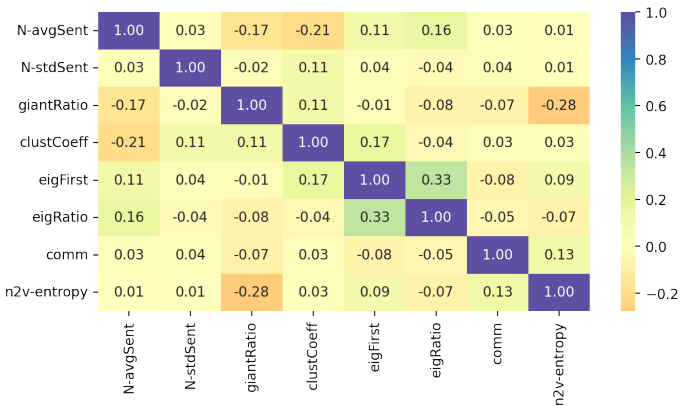


Figure 3.13: Correlation matrix of the features tested in the logistic regression, but without including market features.

Before proceeding to the logistic regression model itself, two points need to be now better discussed. These are the further features initially contrived but then dropped, and the methodology designed to construct the proposed "n2v-entropy" feature (i.e the last row in our table of features, with a "Graph+" class label). Our *news features* limit themselves to measures of the sentiment across articles, since we can safely assume that sentiment is indeed one of the most important market-related features that can be extracted from a plain corpus of economic news. We could have further included e.g. the specific number of news available for each week, but this would have been a noisy metric due to its strong dependence on the data collection step of our framework. Importantly, allowing for such simple but meaningful features of news gives us the opportunity to both test whether news have significant relationships with market dislocations, but also to assess whether our proposed graph representation of news is useful at all. Then, our *graph features* focus on characteristics that we can directly compute and extract from our graph representation of news in a week. For similar reasons as above, we do not include the absolute number of nodes and highest degree value of a graph (while we do compute the ratio between the size of the giant component and total number of nodes). Then, we do consider the average clustering coefficient of the network, but drop the average degree due to related high correlation. Due to a similar instance of high correlation, we unsurprisingly also drop the average and standard deviation of the sentiment of nodes in the graph. On the other hand, we retain the highest absolute value of nodes' eigenvector centrality, and the ratio between such first and second highest values. The latter features allow us to better account for the structure of the network, and the potential presence of highly influential hubs. Finally, we also add to our set of features the optimal number of communities to partition the graph into, as identified by the fuzzy community detection methodology.

Regarding our *graph+ feature*, this is based on encoding nodes of each graph into an embedding space via the node2vec methodology (see Section 3.3.3). Node2vec has been shown to perform significantly well across a variety of practical tasks (such as mapping accuracy, greedy routing, and link prediction) on real-world graphs having from dozens to thousands of nodes [134]. It also suits us, since in this way we can design a metric that accounts for an holistic view of the topology of each graph. Importantly, node2vec embedding spaces cannot be compared across graph, meaning that we need to construct an ad-hoc downstream measure to characterise each point in time.

First of all, we begin by testing embeddings for multiple combinations of node2vec hyperparameters, in order to gauge the associated sensitivity of results. Similarly, we

experiment with different combinations of the return parameter p_{emb} and the in-out one q_{emb}, in order to investigate the sampling strategy to adopt. Embeddings are thus computed for all (i.e. 24) the possible combinations of:

- output vectors dimensions $\in \{8, 16\}$,
- length of simulated walks $\in \{10, 20\}$,
- number of walks per node $\in \{10, 20\}$,
- $(p_{emb}, q_{emb}) \in \{(1, 1), (4, 2), (4, 0.5)\}$.

After extensive exploratory data analysis, we choose the following final combination of (hyper)parameters for our embeddings: output vectors dimension 8, length of simulated walks 20, number of walks per node 20, and $(p_{emb}, q_{emb}) = (1, 1)$. Overall, we saw higher stability of outputs for such values of length of walks and number of walks per node, while favouring larger dimensions does not significantly affect results. Finally, we choose $(p_{emb}, q_{emb}) = (1, 1)$, to allow the weight of edges to directly dominate the sampling strategy, i.e. the probability of transition to a neighbouring node. Thanks to the final set of embeddings achieved, we can then proceed to design related features that alternatively describe each week's structure of news. One in particular is here proposed.

For each week, we have a related graph, and a consequent embedding for the nodes of such graph in a 8-dimensional space. We then design the following proxy for the overall entropy of nodes in the identified embedding space:

1. For each (orthogonal) dimension, we approximate the distribution of data on such axis by computing the probability of occurrence of points in bins of width 0.1.

2. We then calculate the Kullback–Leibler divergence D_{KL} of such distribution P_{data} from the uniform probability distribution P_{unif}. This is

$$D_{KL}(P_{data}||P_{unif}) = \sum_{b \in B} P_{data}(b) \times \ln\Big(\frac{P_{data}(b)}{P_{unif}(b)}\Big), \tag{3.26}$$

   where B is the set of possible outcomes (i.e. bins).

3. Finally, we average these divergences computed on our eight dimensions to achieve a proxy of global entropy. This is indeed named "n2v-entropy".

Contemporaneous and predictive analyses. Now that we have introduced our target variable and motivated the features to test, we proceed to the implementation of our logistic regression models. Importantly, we consider data only from June 1st, 2020 to avoid dislocations related to the Covid-19 crisis. Such extreme exogenous shock would indeed prevent us to focus on more subtle and isolated moments of dislocations, which we believe to be the ones in need of better understanding. We also complete only in-sample tests, due to both the limited number of data points available and our specific interest on assessing whether useful information lies within the modelled *structure and interconnectedness* of news, on which one can then build upon if successfully proven. Since our dislocation dates are strongly under-sampled, we leverage the SMOTE technique introduced in Section 3.3.4 to equally re-balance our target classes. We also standardise all features, and then train a logistic regression model for the following two cases:

1. We allow all features but *market* ones, and try to unravel connections between news and a contemporaneous state of market dislocation.

2. We allow all features, and try to predict an incoming state of market dislocation.

Despite prediction of incoming market dislocations is the most desirable end goal, we are indeed strongly interested in unravelling the features of news' structure that are connected to such critical weeks. These would increase our understanding of such events with statistical confidence, and could suggest novel research ideas.

For each one of the two above cases, we do Recursive Feature Elimination (RFE) to complete feature selection, and require each chosen attribute to be significant at least at the p-value < 0.05 level. The full specifications of our final models are reported in Tables 3.6 and 3.8, for our contemporaneous and predictive cases, respectively. Similarly, Tables 3.7 (left) and 3.9 (left) report the associated confusion matrices, which arise from testing back our resultant models on the (not over-sampled) initial data. Tables 3.7 (right) and 3.9 (right) report instead the precision, recall, and F1-score of each model. The precision of a classifier is its ability to not label a sample as positive if it is negative (i.e. it is the ratio between recognised true positives divided by the sum of true positives and false positives). The recall assesses instead its capability of finding all the positive samples (and is computed as the ratio between true positives divided by the sum of true positives and false negatives). And finally, the F1-score can be interpreted as a weighted harmonic mean of the precision and recall, and reaches its best value at 1 and worst score at 0.

Table 3.6: Assessment of the *contemporaneous* relationships between news and moments of market dislocations

Model:	Logit	Pseudo R-squared:	0.212
Dependent Variable:	y	AIC:	342.0211
Date:	2023-11-04 13:33	BIC:	360.6062
No. Observations:	304	Log-Likelihood:	-166.01
Df Model:	4	LL-Null:	-210.72
Df Residuals:	299	LLR p-value:	1.7552e-18
Converged:	1.0000	Scale:	1.0000
No. Iterations:	19.0000		

	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
N-avgSent	-0.4409	0.1463	-3.0130	0.0026	-0.7277	-0.1541
N-stdSent	0.6774	0.1516	4.4672	0.0000	0.3802	0.9746
giantRatio	0.4679	0.1452	3.2223	0.0013	0.1833	0.7525
eigFirst	0.4502	0.1556	2.8932	0.0038	0.1452	0.7551
comm	0.4539	0.1378	3.2934	0.0010	0.1838	0.7240

Table 3.7: Confusion matrix from our *contemporaneous* model of news and dislocations, and further statistics

		Predicted $R(t)$		
		0	1	Total
True $R(t)$	0	148	4	152
	1	20	3	23
	Total	168	7	175

	Precision	Recall	F1-score
0	0.88	0.97	0.93
1	0.43	0.13	0.20
Accuracy			0.86

Our model for the contemporaneous relationship between news and market dislocations proposes five significant features, as highlighted in Table 3.6. Both the average and standard deviation of simple news' sentiment (i.e. "N-avgSent" and "N-stdSent") are among them, with the former having unsurprisingly negative coefficient. Thus, a more positive sentiment of news clearly lowers the probability of being in a moment of market dislocation. We also believe that a higher standard deviation generally implies a subset of articles with stronger negative sentiment rather than positive, and thus a positive coefficient is seen. The "giantRatio", "eigFirst", and "comm" features are instead related to our proposed graph characterisation of weekly news, and have all positive coefficient. Having a larger proportion of nodes kept within each graph's giant component implies that articles are more interconnected within each other, but a higher number of communities suggests that there are also more themes of discussion. Then, a larger first eigenvector centrality value suggests very influential

Table 3.8: Model for the *prediction* of market dislocations, from the current state of the market and features of news

Model:	Logit	Pseudo R-squared:	0.342
Dependent Variable:	y	AIC:	283.3737
Date:	2023-11-04 13:43	BIC:	294.5248
No. Observations:	304	Log-Likelihood:	-138.69
Df Model:	2	LL-Null:	-210.72
Df Residuals:	301	LLR p-value:	5.2217e-32
Converged:	1.0000	Scale:	1.0000
No. Iterations:	17.0000		

	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
z-vols	1.3160	0.1595	8.2490	0.0000	1.0033	1.6287
eigRatio	-0.3887	0.1610	-2.4144	0.0158	-0.7043	-0.0732
n2v-entropy	-0.4155	0.1593	-2.6077	0.0091	-0.7277	-0.1032

Table 3.9: Confusion matrix from our *predictive* model of market dislocations, and further statistics

		Predicted $R(t)$		
		0	1	Total
True $R(t)$	0	150	2	152
	1	13	10	23
	Total	163	12	175

	Precision	Recall	F1-score
0	0.92	0.99	0.95
1	0.83	0.43	0.57
Accuracy			0.91

hubs (i.e. concepts) within the construct. Merging all such information, we can see that the model hints to a relationship between market dislocations and high entropy of discussion among news. The latter need anyways to be in some way interconnected by construction, and thus point to contagion effects among themes of concern. The further statistics in Table 3.7 provide a benchmark for future improvements, and show how there are non-trivial news-based patterns indeed characteristic of dislocation moments (i.e. by the non-negligible values of precision, recall and F1-score in our target class with label 1).

On the other hand, our model to predict incoming market dislocations proposes only three significant features, as seen in Table 3.8. Clearly, the current state of the market is the strongest predictor for possible dislocations, but both "eigRatio" and "n2v-entropy" are still found significant and with not-negligible participation in the definition of the outcome probability. Such attributes have coefficients with negative sign, for which we now propose a motivation. A lower ratio between the first and second eigenvector centralities implies that the two related nodes have importance

values more similar in magnitude. Thus, this refers to a situation in which there are competing hubs of importance, and likely competing points of concerns. Similarly, a lower Kullback-Leibler divergence actually implies higher entropy within the generated node2vec nodes' embeddings. Such higher entropy points to less uniform structure within narratives, which seems to indeed encode an early alert of incoming market dislocations. For the sake of completeness, we also propose both the confusion matrix and mentioned further statistics for this model too, in Table 3.9. However, the reader must be aware that the current state of the market is indeed the main driver of predictions, lowering the information encompassed in such statistics with respect to assessment of news-based features.

Despite being very simple initial analyses, the considerations just outlined provide a baseline of evidence for a connection between news structure (i.e. beyond their mere sentiment) and market dislocations. Therefore, these first results further motivate the analyses proposed at the beginning of this project, and should prompt more studies that leverage our graph construct to investigate broad corpora of news.

3.5 Conclusions

Starting from a curated selection of economic articles sourced from The Wall Street Journal, our research introduces an innovative and dynamic approach to dissecting news content. We leverage GPT3.5 to sift out the most salient entities within each article, which become the building blocks of a proposed series of graphs. The graphs track indeed the co-occurrence of such entities among news on a weekly basis, and allow investigations on the inter-relations of topics discussed over time. Network analysis techniques and fuzzy community detection are then used to design a comprehensive framework, which systematically unveils interpretable topics and surrounding narratives within news.

The importance of the proposed investigations is highlighted by the results of the logistic regression models. Indeed, we test whether there is a statistically significant connection between features and structure of news, and moments of dislocation within financial markets. As expected (and desired), lower sentiment within news is more likely to be associated with weeks of market dislocation. However, multiple features computed from our graph construct are found to be also significant, especially from the entropy of discussions and consequent likelihood of contagion of sentiment, both in the contemporaneous and predictive scenarios. This suggests that the interconnectedness of news' topics and structure therein are meaningful aspect to further analyse within

financial research, for which our proposed study desires to serve as a first baseline. Improving entity recognition, extending the corpus of news, and designing generalisation studies are examples of possible advances to pursue in this research branch.

As a final remark, we desire to point to the problem of *network alignment*, which is especially important in network biology [102]. Many related studies try to find a measure of protein similarity between proteins in different species, since similar protein structures often imply the same biological results. With a parallel approach, one could investigate more deeply whether equivalent structures among news (but that do not account for the actual "label" of the topic) result indeed in similar market reactions.

Chapter 4

Evolving Correlation of Asset Classes and Market Regimes

By focusing on a financial data set of indices belonging to *multiple* asset classes, we now deliver novel insights on the evolution of long-term market patterns with an original approach. We first identify data-driven macroeconomic regimes by clustering the relative performance in time of the above-mentioned indices. Then, lead-lag relationships within the regimes distinguished are investigated, and a framework for informed portfolio construction is introduced. To achieve such results, we apply state-of-the-art clustering techniques for both signed and directed graphs on the evolving network of either correlations or Granger causalities between indices. And by leveraging those constructs, we are also able to introduce a related stability measure for the transition between regimes, which adds to the existing warning mechanisms that alert for changing market structure and possible need of portfolio rebalancing among asset classes. Overall, our study unravels market features characteristic of different windows in time and leverages this knowledge to highlight market trends or risks that can be informative with respect to recurrent market developments.

4.1 Introduction

The broader dynamics of financial markets can vary significantly across time. These can define periods of different macroeconomic regimes, which are clustered moments of persistent market conditions that can be characterised by external macroeconomic trends. The importance of regime identification mainly lies in its implications and impact on asset allocation and portfolio construction, as highlighted in [6]. This is also a developing area in academic research, with two main branches of efforts to highlight.

The first relates to Hidden Markov Models (see e.g. [50], [128], [104]), but these models require a fixed definition of desired number of regimes that does not allow flexibility of evolution over time. Otherwise, we can find several studies that concentrate on network analysis of correlation and causality matrices of multi-asset returns for the purpose. A review of the related research is available in [88], while [115] is an extensive survey on community detection in dynamic networks. The relevance of correlation studies is motivated by the higher interconnection of stock returns characteristic of periods of market distress [108]. Indeed, [24] and [135] are examples of clustering studies on equities via community detection on the minimum spanning trees built from returns' correlation matrices. The former paper studies data from different stock markets at a frequency from minutes to daily scale, while the latter constructs risk-diversified portfolios from centrality measures on the correlation-based evolving network. However, the literature lacks an extensive study of return correlations and causalities for securities belonging to *multiple* asset classes. Thus, we pursue a detailed investigation of the topic in this Chapter, fueled by its potential to shed further light on macro regimes identification and characterisation.

Main contributions. By looking at the evolution of correlations among a large data set of indices belonging to multiple asset classes, we identify data-driven market regimes that also suggest a clear macroeconomic interpretation. We then propose a stability measure for the transition between such regimes, from the clustering of a signed graph based on the evolving correlations of our instruments of interest. Finally, lead-lag relationships across asset classes are also investigated within such regimes, and are shown to suggest a possible associated investment framework.

Structure of the Chapter. Section 4.2 describes the two data sets considered in the analyses. Then, Section 4.3 details the methodology adopted for correlation analyses and lead-lag clusters identification. Sections 4.4 and 4.5 discuss the results for the two considered tasks. Finally, Section 4.6 concludes this work with final remarks.

4.2 Data

4.2.1 Long Term monthly indices (Data set 1 - LT)

The first data set considered is provided by Fidelity Investments Inc. This is an internal set of 33 indices belonging to different asset classes, whose majority of levels have been reconstructed as at the last day of each month, starting from January 1921.

Due to some remaining missing entries, only the 23 time series that show full data from February 1926 to April 2022 are specifically studied. The related names are reported in Appendix B.1.1, where an identifier from 0 to 22 is assigned to each index for ease of reference. As for the asset classes, indices 0 to 9 relate to equity, 10 is commodity, 11 to 21 are fixed income and 22 acts as a proxy for cash. For each time series, returns as percentage changes of the related levels are computed.

4.2.2 Bloomberg daily indices (Data set 2 - BBG)

To have a more resilient and holistic view of the full financial market, daily levels of a broad set of indices belonging to different asset classes are downloaded from Bloomberg (BBG) and pre-processed. The aim is to strike a balance between a too agglomerated or too granular view, paying attention to the final number of items in each class. The data set consists of 231 daily time series, 43 of which belong to the class of commodities, 29 to currencies, 67 to equities, 29 to bond spreads, 12 to volatilities and 76 to interest rates. The full list of indices for each class is reported in Appendix B.1.2. The data start on 30 September 2005, and end on 1 July 2022, where only trading days are kept. Each time series is converted into returns by calculating percentage differences, or simple differences for interest rates.

4.3 Methodology

Our first aim is to identify a reliable division of time into financial regimes from a data-driven point of view, partly extending [105] and [99]. Each regime is then characterised by summary statistics and relationships between the returns of different asset classes. While both the LT and BBG data sets are considered for this part of the analysis, our main focus will overall be on the latter, due to its broader range of asset classes available and daily frequency.

The second step of our analyses investigates lead-lag relationships between clusters of assets, for each uncovered regime. The results are then tested for profitability of an informed investment on the lagging assets.

4.3.1 Empirical identification of regimes

Correlation of asset returns. For each sliding window of time ΔT, we compute the Pearson's linear correlation ρ_P, Spearman monotonic correlation ρ_S and Kendall

rank correlation τ_K between the time series of indices' returns. Precisely, these are

$$\rho_P = \frac{\sum_{t\in\Delta T}(x_t-\overline{x})(y_t-\overline{y})}{\sqrt{\sum_{t\in\Delta T}(x_t-\overline{x})^2\sum_{t\in\Delta T}(y_t-\overline{y})^2}}, \tag{4.1}$$

$$\rho_S = 1 - \frac{6\sum_{t\in\Delta T} d_t^2}{l(l^2-1)}, \tag{4.2}$$

$$\tau_K = \frac{(\text{NO. CONCORDANT PAIRS}) - (\text{NO. DISCORDANT PAIRS})}{(\text{NO. PAIRS})}, \tag{4.3}$$

where $(x_t, y_t), t \in \Delta T$ are the joint observations for each pair of time series X and Y, $\overline{x}, \overline{y}$ are the related averages, $d_t = \text{RANK}(x_t) - \text{RANK}(y_t)$ and l is the length of ΔT. Concordant or discordant observations are counted from the number of pairs of observations (x_{t_1}, y_{t_1}) and (x_{t_2}, y_{t_2}) whose sort order agrees or not. We aim for a ratio $\frac{l}{N} \sim 2$, where N is the number of assets considered, to avoid too short time-windows that would introduce statistical noise and lower the reliability of correlations [87].

We decide to lower the importance of points further away in history by applying hyperbolic weighting to the Kendall correlation. Indeed, τ_K measures the ordinal association between each pair of time series and one can intuitively add weight $\frac{1}{r+1} + \frac{1}{s+1}$ for each exchange between elements with rank $r, s \geq 0$. Some of the benefits of weighted correlations are mentioned in [107], while unfortunately there is no known distribution-based test that allows to infer a $p - value$ of the constructed weighted Kendall correlation τ_K^w.

Market similarity in time. After obtaining a set of correlation matrices for different points in time, we compute the related pairwise similarities to build a resultant matrix of similarities $\mathbf{S}$. Two relevant measures are considered:

1. Cophenetic correlation [117]. Each correlation matrix $\mathbf{C}$ is transformed into a distance matrix $\mathbf{D}$ by computing $\mathbf{D} = \sqrt{2 \cdot (1 - |\mathbf{C}|)}$. Then, hierarchical clustering is completed via the linkage algorithm with the average method, i.e. applying the UPGMA algorithm. This implies that we follow an agglomerative strategy, by which each single node is added to the cluster to which there is the smallest average distance, in an iterative fashion. Finally, the cophenetic correlation is a measure of the correlation between the distances of points in the feature (Euclidean) space and on the dendrogram generated by the clustering. Therefore, one can employ the distance matrix and dendrogram of different points in time to measure the related similarity. This idea is implemented and the cophenetic correlation computed in both directions (then averaged) for each pair of points in time, thus constructing our matrix $\mathbf{S}$.

2. Metacorrelation. The matrices of assets' correlations are then flattened and, for each pair, Pearson's ρ_P is computed [98], similarly building $\mathbf{S}$.

All-time identification of regimes. Next, we perform Principal Component Analysis (PCA) on the resultant matrix $\mathbf{S}$ of similarity between points in time. PCA is a technique for reducing the dimensionality of a data set, increasing interpretability but at the same time minimising information loss. It does so by mapping the input vectors to a set of orthogonal latent factors (i.e. principal components) of lower dimension, which successively maximise the variance in data explained. PCA takes as input either a covariance or correlation matrix, and computes its n_e top eigenvectors $\mathbf{V}_{n_e}$ (with eigenvalues $\mathbf{\Lambda}_{n_e}$) as

$$\mathbf{S} \approx \mathbf{V}_{n_e} \mathbf{\Lambda}_{n_e} \mathbf{V}_{n_e}^T, \tag{4.4}$$

to propose a dimensionality reduction on indeed n_e dimensions. This can be efficiently computed from the Singular Value Decomposition (SVD) of the centered data matrix, and by taking the resultant right singular vectors as projections of its points.

Leveraging on PCA, we are thus able to project our points in time onto a new space, and cluster them there via KMeans++ [10] in order to identify different financial regimes. The plain KMeans method [81] seeks to minimise the average squared distance between points in the same cluster, but it is dangerously sensitive to the initialisation of its centroids. The algorithm begins by choosing the initial centroids uniformly at random from the data points, and assigns each data point to the nearest center. Then, such centers are recomputed as the center of mass of all points assigned to it, and the process is repeated until convergence. The KMeans++ augmentation adopts a randomised seeding technique to improve both the speed and accuracy of the vanilla algorithm, and forces initial centroids to be points lying far from each other. The optimal number of groups (i.e. regimes) is chosen following the *elbow method*, which implies looking for the rough optimal point with lowest inertia $\iota(k)$ and lowest number of clusters k on the related plot. Inertia measures the distance between each embedded data point $\mathbf{e}_t$ and its centroid $\mathbf{cr}(\mathbf{e}_t, k)$ for a clustering in k groups, and aggregates them as

$$\iota(k) = \sum_{\mathbf{e}_t} ||\mathbf{e}_t - \mathbf{cr}(\mathbf{e}_t, k)||_2^2. \tag{4.5}$$

Finally, the stability of the results needs to be checked while perturbing the chosen parameters.

Network study of the evolution of correlations. To increase the confidence on the identified regimes and build a metric that signals related transitions in real-time (to use in addition to existing regime-monitoring tools), we propose the following original approach. We first consider the set of correlation matrices between assets returns at different points in time, and retain only entries with magnitude above a threshold. From each matrix, we then build the related signed network $G = (I, E)$. The set of nodes I are indices and edges $(i, j) \in E$ with $i, j \in I$ have weights w_{ij} from the non-zero correlations. The giant component is saved, and we proceed to cluster nodes in each final graph in k groups via the symmetric SPONGE algorithm [48].

The SPONGE algorithm is a spectral algorithm that aims at having the maximum number of positive edges within clusters while having highest number of negative edges between clusters. To give a better intuition about its foundations, let us first consider an unsigned graph H where weights w_{ij} are strictly non-negative. For any cluster $C \subset V$, we then define the $\text{cut}_H(C, \overline{C}) := \sum_{i \in C, j \in \overline{C}} w_{ij}$ as the total weight of edges crossing from C to $\bar{C}$. We also define the volume of C, $\text{vol}_H(C) := \sum_{i \in C} \sum_j w_{ij}$ as the sum of degrees of nodes in C. Then, SPONGE focuses on contemporaneously minimising the following two measures of *badness*:

1. the proportion of positive edges out-going a cluster with respect to the positive volume of such cluster, expressed as
$$\frac{\text{cut}_{G^+}(C, \overline{C})}{\text{vol}_{G^+}(C)}, \tag{4.6}$$
2. the inverse of the the proportion of negative edges out-going a cluster with respect to the negative volume of such cluster, reading
$$\Big(\frac{\text{cut}_{G^-}(C, \overline{C})}{\text{vol}_{G^-}(C)}\Big)^{-1} = \frac{\text{vol}_{G^-}(C)}{\text{cut}_{G^-}(C, \overline{C})}, \tag{4.7}$$

where $G^{\pm}$ are the subgraphs generated by the subset of positive or negative ($\pm$) edges in G. The above objectives are combined into one optimisation problem, which is then reformulated in [48] as a suitable eigenvalue problem, and solved. The resulting smallest eigenvectors can be consequently used to embed nodes of the graph and group them via a desired clustering algorithm.

When applying the SPONGE algorithm, we need to input the desired number of communities to look for. To choose the optimal value for such parameter, we compute the signed modularity [58]

$$Q_{signed} = Q_+ \cdot \frac{W_+}{W_+ + W_-} - Q_- \cdot \frac{W_-}{W_+ + W_-}, \tag{4.8}$$

where $W_{\pm}$ are the total weights of the subgraphs generated by the subset of positive or negative ($\pm$) edges. Similarly, $Q_{\pm}$ are the modularities of such subgraphs (see Eq. (3.6) in Chapter 3 for the usual mathematical formulation of modularity). Our overall motivation behind using Eq. (4.8) is to measure the trade-off between the tendency of positive weights to form communities, and of negative weights to separate them.

Once a clustering for each chosen point in time is found, we compare communities at consequent periods and compute a measure of the related stability. This is achieved by calculating the Adjusted Rand Index (ARI), which indicates the consistency across any two input partitions similarly as to how detailed in Eq. (3.24) from Chapter 3. Whenever there is a significant drop in stability of the communities identified, then we can consider a regime change to have happened.

Characterisation of the identified regimes. Once regimes are identified, their peculiar features can be studied. We compute the average correlation matrix for each regime by considering the points in time assigned to it. Then, we also calculate the related average mean returns ret, average standard deviations of returns σ_{ret} and annualised Sharpe Ratios $S = \frac{ret}{\sigma_{ret}} \cdot \sqrt{252}$ (assuming 252 trading days within a year) of the underlying asset classes.

Finally, we build the summary network of correlations for each regime and the related distance graph. On the former, the symmetric SPONGE algorithm can be again run to cluster instruments. This allows us to analyse specific relationships across clusters, along with their composition across asset classes. On the latter graph, we compute instead the betweenness centrality [53]. This metric is defined as

$$\Upsilon(l) = \sum_{i \neq l \neq j} \frac{\lambda_{ij}(l)}{\lambda_{ij}}, \tag{4.9}$$

where $i, j, l \in I$, λ_{ij} is the total number of (weighted) shortest paths from node i to node j, and $\lambda_{ij}(l)$ is the number of those paths that pass through l. Betweenness centrality allows us to understand which indices have the highest influence over the flow of information or shock propagation in the graph of each regime.

4.3.2 Lead-lag clusters specific to regimes

Causality relations in regimes. After having identified macro regimes, we investigate whether any significant latent lead-lag relationships can be extracted within them. The Granger causality test [59] is a statistical hypothesis test for determining whether one time series is useful in forecasting another, with the requirement of both

series being stationary. Intuitively, one considers two time series X and Y, and models them within a bivariate autoregressive system. To test if X Granger-causes Y (or vice versa respectively), we gauge whether any lags of X are statistically significant in such model.

Thus, we proceed to fragment the time series of indices' returns into the related regimes. For each regime, the Granger causality is then computed between each pair of indices for lags $g \in [\pm 1, \pm 5, \pm 10, \pm 16, \pm 21, \pm 42]$ trading days and results at the 0.05 significance level are retained. In the rare scenario that both indices in a pair lead or lag each other, we subtract the weakest relationship from the strongest and keep the deflated result, which essentially leads to a skew-symmetric lead-lag matrix. The optimal lag g^R for each regime R is chosen as the one leading to the highest number of significant Granger causalities, since our work aims at unravelling an agglomerated informative view.

Each resultant causality matrix is used to build a directed network $G^{R,dir}$ for the related regime, from which it is possible to identify leading and lagging groups of indices via the Hermitian clustering algorithm of [49]. This is a spectral algorithm that aims at clustering nodes according to the similarity in their outgoing and incoming edge patterns. Specifically, the algorithm starts by considering a weighted directed graph G^{dir}, where each directed edge $j \rightsquigarrow l$ has weight w_{jl}. Then, the Hermitian adjacency matrix of G^{dir} is defined by computing entries $A_{jl} = (w_{jl} - w_{lj})i$, where i is the imaginary unit. If no directed edge is present between nodes j and l, then $A_{jl} = 0$. Importantly, $A_{jl} = \overline{A_{lj}}$ implies that our adjacency matrix is indeed Hermitian, and therefore has real-valued eigenvalues. Eigenvectors corresponding to the largest eigenvalues of the Hermitian adjacency matrix are thus computed, and can consequently be used to project and cluster nodes onto such new space. Such Hermitian algorithm was shown to be highly effective at exposing pairs of clusters (A, B) such that there is a strong *imbalance* in the directional flow of edges between A and B. This means that the majority of the edges flow cluster A to cluster B, and very few in the opposite direction. In light of our analyses, an imbalance of such type between A and B would translate into a lead-lag relationship between the times series in A and those in B.

To define the number of communities to look for, we leverage the idea that there is structural similarity between the different asset classes. These classes might lead or lag each other, and thus a clustering that generates groups with highest inner coherence in composition is preferred. As a measure of goodness, we choose the $v - measure$ [114]

$$v = (1 + \beta) \cdot \frac{\text{HOMOGENEITY} \cdot \text{COMPLETENESS}}{(\beta \cdot \text{HOMOGENEITY} + \text{COMPLETENESS})}. \tag{4.10}$$

Here, β weights our interest between focusing on homogeneity or completeness of the extracted clusters with respect to the related composition across asset classes.

Investment implications. As a final step, we test the performance of an investment based on lead-lag clusters specific to each regime against a vanilla benchmark that bets uniformly on all assets. For the former, we consider the most leading cluster in each regime R, compute the average return of the related subset of indices over the past g^R days, and use its sign as a signal to buy or sell assets uniformly in the most lagging cluster. The positions are closed after g^R trading days. The average returns between the two strategies are then compared, despite being generated by in-sample analyses. Indeed, the aim of this example is simply to show further features characteristic of regimes, and lay the foundations for an investment framework that can be further developed by practitioners if of interest.

4.4 Results: Identification of regimes

4.4.1 LT data set

Similarity matrix between points in time. Following the methodology described in Section 4.3.1, we compute the Pearson's, Spearman and (weighted) Kendall correlations on the time series of LT assets' returns using a time window of length four years, which slides every month. Then, regimes are identified by clustering the matrices of cophenetic correlations or metacorrelations computed on the above sets of correlations. Figure 4.1 shows these matrices of time-similarity from weighted Kendall correlations, which is the measure that produces the sharpest patterns. Indeed, the weighting dampens spill-overs of high correlation of furthest points in time in the overall description of each period under consideration. The results from other initial correlation options are not shown for the sake of brevity, but general agreement on the clear block structure of each matrix is witnessed. Each mean cophenetic correlation between a dendrogram and its own feature matrix is $> 93\%$, meaning that the hierarchical clustering is indeed able to preserve the relevant information of the data. However, the metacorrelation matrix provides a slightly stronger diversification in the level of similarity between different points in time, and will be the method of choice in the following analyses.

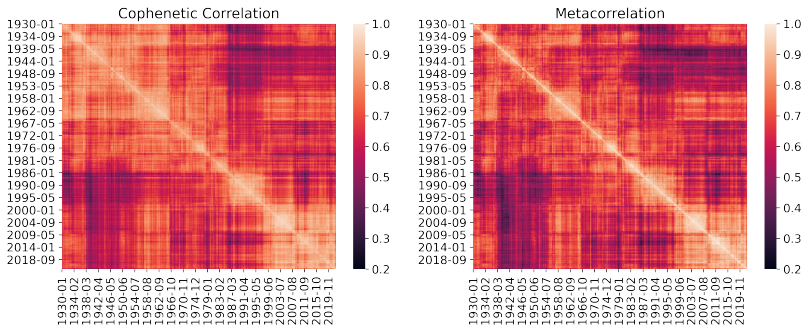


Figure 4.1: Time similarity measured via the cophenetic correlation or metacorrelation on correlations of returns. Windows long four years are used.

Clustering of the similarity matrices. The time-similarity matrices of Fig. 4.1 show a clear block structure that already hints to a related division of periods into regimes. We perform PCA on these matrices and find that three dimensions explain 92% of the variance within the data for the cophenetic case, while four dimensions account for 91% of it for metacorrelations. Next, we cluster the points projected onto each new space via KMeans++. We choose the best number of regimes following the elbow method on an inertia plot, and check the stability of results against perturbations of the amount of desired clusters. The overall stability of the results is also confirmed by varying initial random seed, time window length and number of dimensions kept in PCA. The results suggest the existence of **six** regimes, shown in Fig. 4.2 from the metacorrelation matrix. We also plot inflation via the Consumer Price Index Year-Over-Year in percentages (CPI YOY %), to have a macroeconomic indicator for comparison. Finally, the average correlation matrix between assets is calculated for each regime. The results are shown in Fig. 4.3, while Table 4.1 reports statistics on the related correlations.

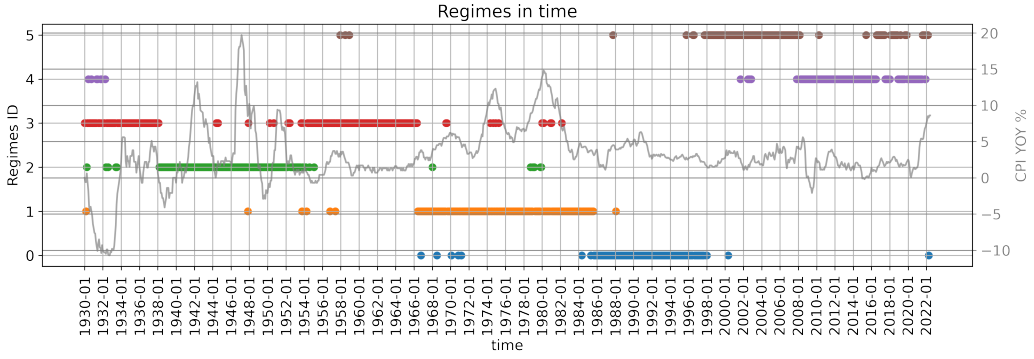


Figure 4.2: Division of time into six regimes from the correlation of returns of the LT indices. Inflation (CPI YOY %) is shown as a macroeconomic reference.

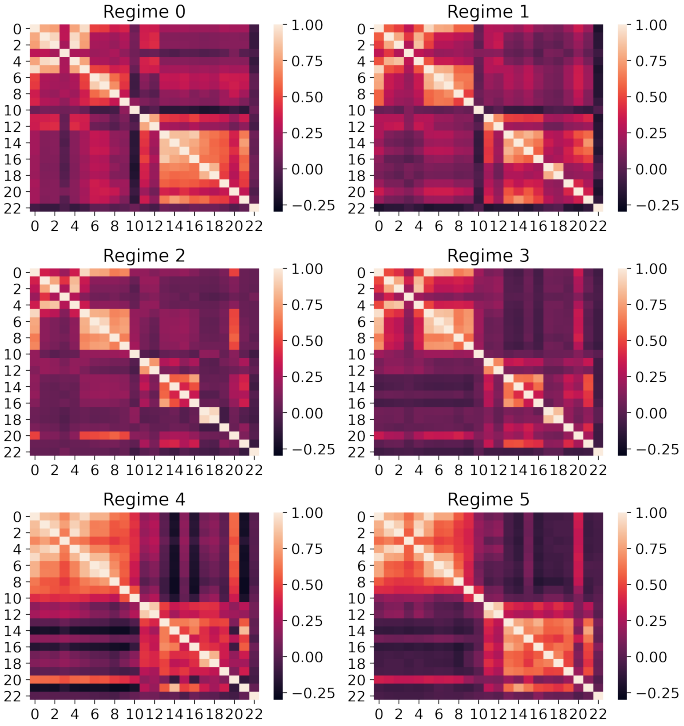


Figure 4.3: For each regime identified from the LT data set, its matrix of average correlations between indices is shown.

Table 4.1: Average (absolute) correlation between indices during the six different regimes identified from the LT data set

Regimes	**0**	**1**	**2**	**3**	**4**	**5**
Avg. corr	0.38	0.34	0.27	0.28	0.36	0.32
Avg. |corr|	0.43	0.41	0.32	0.35	0.47	0.42

Discussion. Table 4.1 highlights how regimes R2 & R3 have much lower average correlations between indices' returns than the others. These are quieter regimes that span years from the ′30s to mid ′60s, as depicted in Fig. 4.2. While they encapsulate World War II, R2 & R3 can be seen as representatives of the Golden Age associated with the post-War economic boom. Regime 3 ends in the mid ′60s, when indeed economists mark the beginning of the Great Inflation (from 1965 to 1982 [1]). The Great Inflation can be recovered in R1 and is followed by a moment of recession. The latter relates to R0, which indeed has higher average correlations. However, the highest average correlations happen in R4 when both the Great Financial Crisis (GFC) and

[1] `https://www.federalreservehistory.org/`

COVID-19 hit.

To delve deeper into the identified regimes, we now focus on their characteristic heatmaps of Fig. 4.3 that show the average correlations between indices' returns. As expected, R4 has the strongest block structure that underlines broadly high correlations within equity indices (IDs 0-9) and within fixed income ones (IDs 11-21). Furthermore, there are also significant off-diagonal correlations especially on the negative range. Regime 5 shows a similar structure but less enhanced, completing the period of the 2000s. One could propose to agglomerate the six regimes into three major ones (3&2, 1&0 and 5&4) by looking at the heatmaps' structure. However, the importance of the full discretisation becomes clear once the set of regimes are compared with the CPI YOY % index. Indeed, the levels of inflation and inflation's volatility vary substantially over time and can be associated to the regimes identified. Moving to specific indices, commodity (ID 10) has neutral or slightly negative correlations to equity in R1 & R0 but the two categories become strongly positively correlated especially in R4. Then, IDs 14&16 (risk-free, US Treasury investments) and ID 20 (higher risk, below investment grade rated investments) strongly react to market distress as intuitively expected. They show negative correlation in R4 but a positive relation during the Great Inflation. Overall, our methodology extracts data-driven regimes that are both intuitive and significant from a macroeconomic perspective. Thus, our approach is now confidently deployed also on the BBG data set.

4.4.2 BBG data set

Identification of regimes. We compute correlations between BBG indices' returns using a window two years long that slides one week at every iteration. Motivated by similar considerations to the ones for the LT data set, we focus on the extraction of regimes from metacorrelations between flattened matrices of weighted Kendall correlations. The matrix of periods' similarities shows again a clear block structure but is not reported here for the sake of brevity, as other following intermediate steps. We do PCA and see that three dimensions describe $\sim 90\%$ of the variance of the data. Points in time are thus projected onto this new space and clustered via KMeans++, where the optimal number of groups is again chosen by looking at the inertia loss function. The optimal discretisation of time is found to be into **seven** regimes, which are shown in Fig. 4.4. We then report average (absolute) correlations in Table 4.2, while Fig. 4.5 highlights the average mean return, average standard deviation and Sharpe Ratios among asset classes during each regime. R0 and R4 are periods of crisis

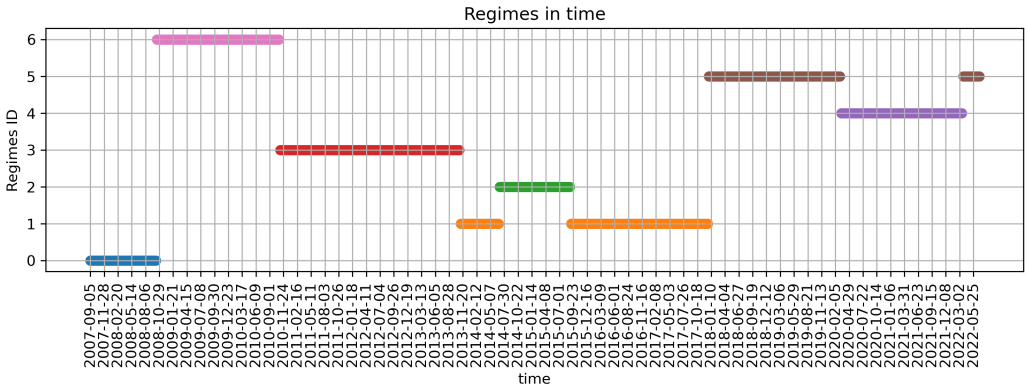


Figure 4.4: Division of time into seven different regimes identified from the correlation of returns of the BBG indices.

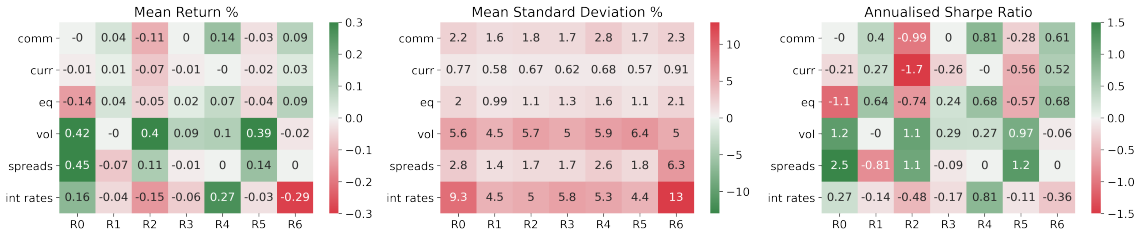


Figure 4.5: Mean return, standard deviation and annualised Sharpe Ratio of the indices belonging to each asset class divided by regime for the BBG data set. Note that this akin to a long-only portfolio containing indices from a specific asset class.

Table 4.2: Average (absolute) correlation between indices during the seven regimes identified from the BBG data set

Regimes	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Avg. corr	0.06	0.05	0.04	0.08	0.06	0.05	0.08
Avg. |corr|	0.15	0.14	0.13	0.18	0.15	0.14	0.16

and indeed, interest rates have some positive returns which could be explained by investors flocking to lower-risk securities during this type of market conditions.

Interestingly, R3 is the regime with highest absolute correlation and relates to the period between the end of 2010 and 2013. In 2010, there was the first drop in oil prices otherwise rallying in the related commodity super-cycle. However, this is also one of the moments of strong rounds of Fed's Quantitative Easing. By looking at the variation of standard deviation in interest rates returns, we notice how this measure dropped significantly in regimes after the end of 2010. The intuition is that the inflow of liquidity might have destabilised correlation levels when considering the broader

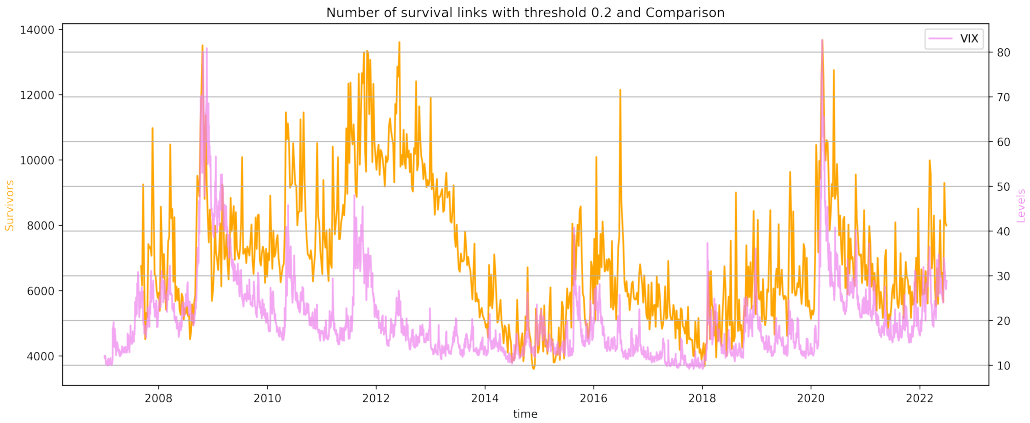


Figure 4.6: Comparison of the evolution of the number of survival links and peaks in the VIX Volatility Index.

system of all asset classes.

Network-based identification of regimes. To increase the confidence in the regimes detected, we also separately study the structural evolution over time of the network of returns' correlations between indices. One network is built each week, following the adopted sliding window. Nodes are the BBG indices and weighted edges are added from the related correlations if their magnitude is above the threshold $|\tau_K^w| \geq 0.2$. Then, the giant component of each network is kept. A higher number of survival links is an indicator of increasing correlations between assets and points towards periods of market distress. For the interested reader, it is worth mentioning that this is indeed verified for the evolution of the network when the introduced indicator is compared to peaks in the VIX Volatility Index. However, the decay of volatility levels is faster than the dissipation of links, as shown in Fig. 4.6.

Then, we cluster nodes of each signed network via the symmetric SPONGE algorithm. The similarity between communities extracted at consequent points in time is computed by averaging the ARI value between the current clustering and each clustering for the past four weeks. The result is plotted in Fig. 4.7, where the regimes previously found from the metacorrelation matrix are also shown in different colors. To choose the number of groups to cluster for at each iteration, our algorithm automatically picks the value for which there is the highest increase in Q_{signed} when clustering for $k \in [2, ..., 10]$, which suffices to build a summary measure to validate the regimes identified. Indeed, there is general consensus between strong drops in stability of the ARI average and a transition among the previously identified regimes.

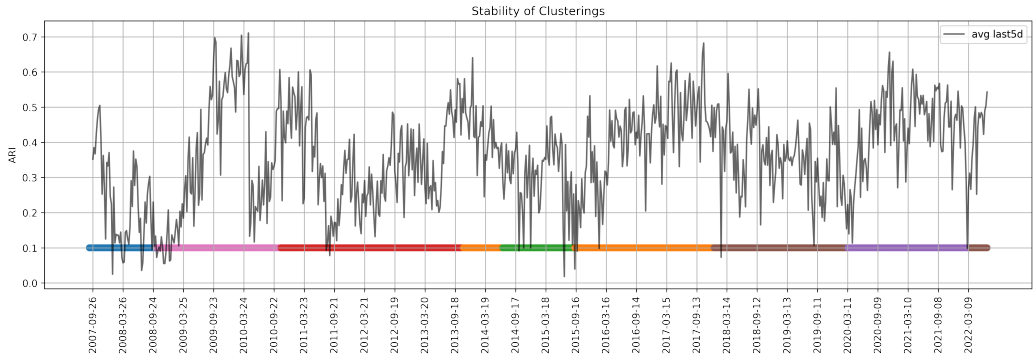


Figure 4.7: Stability of communities detected in time. The average ARI between clusterings from the latest five dates is shown, which agrees with drops in the measure and regime changes as from the metacorrelation matrix.

Network-based characterisation of regimes. Seven distance matrices $\mathbf{D}_R = \sqrt{2 \cdot (1 - |\tau_{K,R}^{w,avg}|)}$ are calculated from the average correlation matrices of regimes, and the related graphs are build. We then investigate which nodes have the highest betweenness centrality for each regime, and observe that the distribution of such centralities varies substantially with peak magnitudes in regime R2. Table 4.3 reports the five nodes with the highest centrality for each regime, allowing us to understand the most influential indices for the propagation of trends during different periods. As an example, it is interesting to see the Singapore exchange rate (SGDUSD) being the main bridge for the diffusion of the COVID shock due to its highest centrality in R4.

Then, the average correlation matrix within each regime is also directly considered, and the related signed graph built. Consequently, we cluster such nodes via the symmetric SPONGE algorithm. The optimal number of communities k^R for each regime is identified by looking at the signed modularities for $k \in [2, ..., 15]$. The results are depicted in Fig. 4.8, with the addition of the composition among asset classes of

Table 4.3: Nodes with highest betweenness centrality for each BBG regime

	Reg: 0	**Reg: 1**	**Reg: 2**	**Reg: 3**	**Reg: 4**	**Reg: 5**	**Reg: 6**
1st	MSCI US	MSCI EAFE	US Corp 10-25Yr	MSCI Europe	SGDUSD	MSCI Europe	US Corp 5-10Yr
2nd	US Corp 5-10Yr	DE 10Y	DE 10Y	AUDUSD	MSCI Europe	AUDUSD	MSCI Pacific
3rd	BBG COMM	MSCI Italy	MSCI Europe	BBG COMM	MSCI EM Asia	AU 10Y	BBG COMM
4th	MSCI Europe	SE 10Y	MSCI EAFE	US Corp 10-25Yr	MSCI US	US Infor. Tech.	KR 2Y
5th	EAFE SmallCap	SZ 10Y	MSCI EM	MSCI US	JP 10Y	DE 10Y	MSCI World

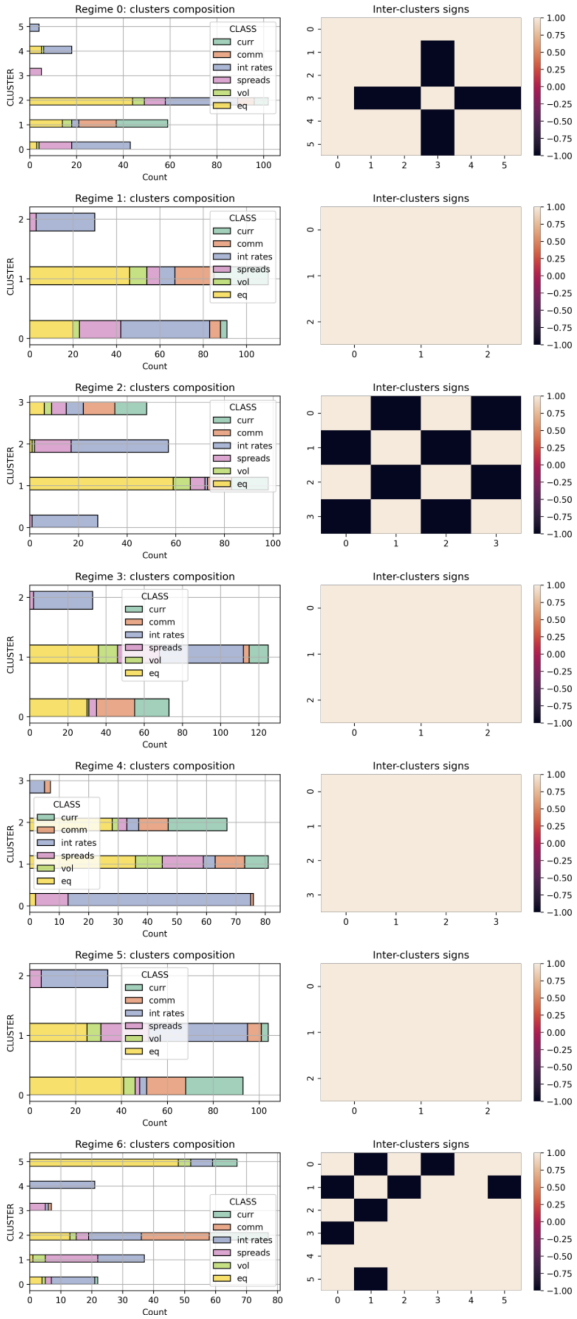


Figure 4.8: For each BBG regime, this plot shows the composition of communities identified by clustering the related signed network (left column). The sign of average correlations between the clusters (right column) is also reported.

each identified community for the different regimes. In parallel, we also report the sign of average correlations between each pair of communities, which highlights that only three regimes show some dominant negative correlations between clusters. Focusing on interest rates, these can be divided into risk-free ones (e.g. Germany, Switzerland)

or riskier options (e.g. Chile, Indonesia). Looking at the histograms for the latest R4 & R5 and the full composition of communities, it is interesting to notice that this division is violated during the COVID-19 crisis. Almost all indices related to interest rates are clustered together in R4, meaning that even the safest investments acquired some risk.

As intended, the framework proposed sheds further light on the evolution of broad market dynamics, whose understanding allows better investment decision and risk-on/off attitude switching.

4.5 Results: Lead-lag clusters

For each regime depicted in Fig. 4.4, we extract the related partial time series of daily returns for all assets. Then, we compute Granger causalities between indices for different lags in each regime, following the methodology highlighted in Section 4.3.2. The lag generating the highest number of significant relationships (generally over 50%) at the 0.05 level is kept as characteristic of the regime, and the resultant relationships are encoded in a skew-symmetric matrix. These optimal lags g^R's are reported in Table 4.4, where we can notice that R5 has a much longer-than-average g^5, of 42 trading days, and relates to turning points of monetary tightening.

A directed network is built from the Granger causality matrix of each regime R. The Hermitian clustering algorithm is then deployed to cluster its nodes, and understand evolving communities of major leaders and laggers. To define the number of clusters k^R to aim for, we recall that the indices belong to six asset classes with likely intrinsic structural similarities. Therefore, we decide to investigate the forecasting power between asset classes allowing for inner divisions. Values $k \in [2, ..., 15]$ are tested, and for each clustering the related $v - measure$ with $\beta = 10$ is computed. While other values of β are checked, this choice assures to have both homogeneity and completeness of the extracted communities, but favouring the latter. A too granular

Table 4.4: Average return of an investment based on lead-lag clusters or on the full universe of indices for each regime. The optimal lag g^R of each regime is also reported

Regimes	**0**	**1**	**2**	**3**	**4**	**5**	**6**
$\mathbf{g^R}$	**10**	**1**	**1**	**1**	**5**	**42**	**5**
Lead-Lag avg. ret %	5	1	0.1	0.8	0.2	0.3	0.5
Benchmark avg. ret %	1	−0	−0	−0	0.6	0.2	−0.3

view would be suggested with lower β's, hindering the interpretability of the results. A first rough jump is generally witnessed for $k = 5$, and this value is chosen as the fixed number of communities to look for in all regimes. The clustering is then completed, and both the strength of links between clusters and the clusters' composition within each regime compared. While the full set of plots is not shown for the sake of brevity, they reveal that the composition of the most leading and most lagging clusters varies significantly between regimes. These are usually equities vs. interest rates, but also currencies become strong leading assets in R3 & R6. As a final intuitive example, the analyses show indeed that the most leading cluster in R0 (GFC) is composed of US equities and US volatilities, while European interest rates lag.

To conclude, the average return of a portfolio constructed by considering the most leading and lagging clusters is compared versus a plain investment on all indices. Table 4.4 shows such results. In each regime, the introduced approach provides positive return (reported in percentage) and also outperforms the benchmark, apart for R4. While this is not a real investment strategy, the framework shows the importance of recognising both regimes and their inner evolving leaders and laggers, thus providing insights towards better portfolio construction and risk management approaches.

4.6 Conclusions

Our work sheds further light on the co-evolution of relationships between *multiple* asset classes and financial market conditions. Novel network-based approaches are proposed, which allow to both create an additional new tool to track possible transitions between macro regimes, and characterise the associated distinctive relationships among assets. By virtue of state-of-the-art network clustering techniques, we further introduce a new investment framework that leverages regime-specific groups of securities that lead and lag each other, in order to optimise asset allocation. Thanks to these investigations, we push towards a deeper understanding of the evolution of the global state of financial markets, which is of high relevance for improved risk assessment and portfolio allocation.

Chapter 5

Blockchain Transactions and Species of Traders

Uniswap is a Constant Product Market Maker built around liquidity pools, where pairs of tokens are exchanged subject to a fee that is proportional to the size of transactions. Each such trading event is registered on the Ethereum blockchain, which indeed provides a complete view of all liquidity provision and consumption actions that get finalised on Uniswap. Thanks to this unprecedented level of transparency in available market data, we decide to pursue an investigation of classes of behaviour of related traders. We first propose a systematic workflow to extract a tractable sub-universe of liquidity pools, where the interconnection among such pools is maximised to capture broader trading dynamics within the ecosystem. The resultant set of 34 pools is then used to cluster market participants according to their liquidity consumption behaviour over such environments, for the time window January-June 2022. Introducing a novel approach, we represent each liquidity taker (i.e. trader) by a suitably constructed *transaction graph*. Such graph is a fully connected network where nodes are the liquidity taker's executed transactions on the 34 pools of reference, and edges contain weights encoding the time elapsed between any two transactions. We then extend the NLP-inspired graph2vec algorithm to the weighted undirected setting, and employ it to obtain an embedding of the set of graphs representing market participants. This embedding allows us to extract seven clusters of liquidity takers, with inner equivalent behavioural patterns that can be interpreted in terms of trading attributes (e.g. preference for exotic assets over stablecoins, frequency of activity, tolerance for higher trading fees).

5.1 Introduction

A *blockchain* is a type of Distributed Ledger Technology (DLT) that stores users' transactions on an increasingly long sequence of blocks of data. The ledger is replicated across a network of servers to allow the validation of new transactions into new blocks by a peer-to-peer (P2P) computer network, thus increasing trust, security, transparency, and traceability of data. Bitcoin was the first blockchain to acquire worldwide notoriety. It was designed by the person(s) known via the pseudonym Satoshi Nakamoto during 2007 and 2008, and described in its whitepaper [100] in 2009. The project was released as an open source software in 2009, at which point Bitcoin started slowly acquiring increasing value and seeing higher trading volumes. However, there are important limitations preventing Bitcoin from hosting complex applications and general smart contracts, such as lack of Turing-completeness, value-blindness, lack of state, and blockchain-blindness [30]. These limitations fueled the rise of the Ethereum blockchain, which was first described in the 2013 whitepaper [30].

Ethereum supports smart contract functionality and, due to this, it is able to offer financial services that do not rely on intermediaries such as brokerages, exchanges or banks. Thus, Ethereum is commonly considered as the protocol that first allowed the formulation of foundations for Decentralised Finance (DeFi). Within DeFi, individuals can lend, trade, and borrow using software that automatically broadcasts their intentions for P2P verification. Valid financial actions are then recorded on the blockchain. Decentralised Exchanges (DEXs) are a direct result of this setup, and started being designed and implemented mainly from 2017. They differ from the usual centralised exchanges, since they are non-custodial and leverage the self-execution of smart contracts for P2P trading, allowing users to retain control of their private keys and funds. One of the first and most established DEXs at the time of writing is Uniswap, which was launched in November 2018. Uniswap is an Automated Market Maker (AMM) running on Ethereum, meaning that it allows the automatic exchange of pairs of tokens following a mathematical formula (where each pair is traded within a different "liquidity pool"). There exist three versions of Uniswap (namely v1, v2, v3, see the whitepapers [2], [3], [4] respectively) that update its design and evolve its functionalities.

Many further interesting new and old finance concepts live within DeFi and beyond DEXs. One such concept worth mentioning is that of *stablecoins*. Stablecoins are digital assets that are pegged to the value of a fiat currency, and can be useful to exit risky positions while remaining inside the crypto ecosystem. Some stablecoins are

fiat-backed (e.g. USDC, Tether), while others are backed by an over-collateralised pool of cryptocurrencies (e.g. DAI). There also exist algorithmic coins (e.g. UST), which closely resemble traditional pegged exchange rates and are indeed also vulnerable to speculative attacks, i.e. as it happened with the Terra-Luna crash in May 2022. Apart from stablecoins, DeFi provides several lending protocols (e.g. Aave, Compound, Instadapp, Maker), protocols for derivatives trading (e.g. dYdX, Futureswap, Nexus), and DEX aggregators (e.g. 1inch) that optimise routing to take advantage of the best exchange rates across multiple other exchanges. In [74], we find an interesting study of the interactions between different blockchain protocols.

While DeFi is fascinating, it is also the stage of many scams, speculative high-risk investments, direct blockchain attacks, and money laundering events. On top of that, its complexity and atomicity might disadvantage small users, whose transactions can e.g. be re-ordered before execution by the validators for their own profit, known as *miner extractable value* (MEV). Despite the current effort of regulators to penetrate the crypto world and establish some equilibrium between centralisation and decentralisation of power, the current situation and possible upcoming developments are still highly confusing, especially for outsiders or newcomers. Interesting overviews and critical thoughts are presented in [118] and [84], where the latter work especially discusses enforcing tax compliance, anti-money laundering laws, and how to possibly prevent financial malfeasance. The different layers of DeFi are studied in [70], where the related specific risks, i.e. at the blockchain, protocol, pool, and token level, are also analysed and a risk parity approach for portfolio construction in Uniswap is proposed.

The current academic research has also become significantly active in its efforts to understand the inner dynamics of DEXs and external relationships with the well-known traditional stock market, especially from an empirical and data-driven point of view. Among interesting recent studies there is [52], where the authors investigate how promoting a greater diversity of price-space partitions in Uniswap v3 can simultaneously benefit both liquidity providers and takers. Then, [54] studies whether AMM protocols, such as Uniswap, can sustainably retain a portion of their trading fees for the protocol and the expected outflow of traders to competitor venues. Inefficiencies between Uniswap and SushiSwap are investigated in [17], where sub-optimal trade routing is addressed. However, [8] shows that constant product markets should closely track the reference market price. Flows of liquidity between Uniswap v2 pools are studied in [65], while [64] shows the difficulty of earning significant returns by providing liquidity in Uniswap v3. Interestingly, [36] fully characterises the wealth of agents active on Uniswap v3, and shows that liquidity providers suffer predictable losses.

Related literature. We nevertheless believe that there does not exist any study yet, which is able to clusters DEX traders in interpretable classes as we will do, by simply considering the temporal component of executed transactions and the generic related pools of reference. Furthermore, only few research efforts have investigated similar objectives on Limit Order Book (LOB) data, since the confidential nature of such centralised data oftentimes prevents accessibility. A relevant study is the recent work in [45], in which the authors extract and analyse four distinct clusters of client order flow from a large broker in US equity markets. This is achieved by assessing the similarity of agents' trading behaviour from e.g. the side usually taken (buy/sell orders), number of orders, time of submission, and size. The uncovered heterogeneity of order flow is modelled by partitioning clients into different clusters, for which representative prototypes are also identified. Pursuing a complementary analysis, the authors of [34] build statistical models to describe how individual market participants choose the direction, price, and volume of orders. Such models are endorsed with features representative of both the recent and current state of the LOB, and also the agent's past actions, cash levels, and inventory levels. Then, the coefficients from such fitted models are used to cluster trading behaviour, suggesting the existence of three separate groups of agents, namely directional traders, opportunistic traders, and market makers. Relatedly, the authors of [86] analyse audit trail data at transaction level from the S&P 500 eMini Future during four days, including the flash crash in May 2010. They parse through the footprints of all traders with a proposed machine-learning method, and identify five related categories of agents (i.e. high frequency traders, market makers, opportunistic traders, fundamental buyers and sellers, and small traders). These classes mainly differ from the dynamics of intraday trading volume and end of day holdings, as similarly considered in [73]. The seminal work in [73] further argues for the importance to study the heterogeneity of order flow, as for its implications for both risk managers and market regulators. And this is indeed one of the reasons why we study such topic also in the context of decentralised markets.
The work in [131] also introduces machine learning methods to cluster active traders, by leveraging a custom discrimination metric that analyses times, volumes, and prices at which traders are willing to transact. In particular, the authors compare traders' individual order book participation over defined windows of time.

Main contributions. This study advances our understanding of the Uniswap ecosystem. First, we systemically investigate how to subset the extremely large set of Uniswap v3 pools, most of which are greatly illiquid, to still retain an highly accurate

view of Uniswap v3 dynamics. We assess both the pools' inner features and their interconnectedness, and propose a filtration mechanism to achieve a tractable subset of such pools. This allows us to then analyse the behaviour of multiple liquidity takers across different pools. We encode each trader's activity into a related graph, and cluster the resultant set of graphs to identify structural similarities in traders' behaviour. This clusterisation highlights the main typologies of market participants, which are found to have interpretable external market features, and it enables further downstream tasks customised for each cluster individually if desired.

Structure of the Chapter. Section 5.2 explains how the Uniswap DEX operates. In Section 5.3, we then identify the most important and interconnected liquidity pools for different time windows within 2022. Next, we cluster traders according to their behaviour on the relevant sub-universes in Section 5.4. Finally, we summarise our thoughts and discuss further possible research directions in Section 5.5.

5.2 Constant Product Market Makers

The most common type of blockchain-based AMMs is the *Constant Function Market Maker* (CFMM), which was also the first class of AMMs to successfully function as a real-world financial market. CFMMs are based on a function that establishes a pre-defined set of prices for the assets available on the exchange, based on the available reserves of these mentioned resources. If the function that characterises a CFMM involves the product between reserves of assets, then we call the DEX a *Constant Product Market Maker* (CPMM). Uniswap is indeed a CPMM.

5.2.1 Liquidity pools

All versions of Uniswap (and any general CPMM) are built around *liquidity pools*, which are venues where a pre-defined subset of the set of digital assets $A = \{$USDC, DAI, WETH, WBTC...$\}$ are stored and traded following a common pricing formula. For precision of notation, we note that we refer to the wrapped versions of some tokens, e.g. WBTC instead of Bitcoin BTC and similarly WETH for Ether ETH, since wrapped tokens are standard ERC-20 tokens created to allow a coin to be traded and used on a non-native blockchain or decentralised application. In Uniswap, only a pair of assets $(X, Y) \in A^2$ can be traded per liquidity pool.

Three market events can happen on a liquidity pool that deals assets (X, Y):

1. **Swap** operation - the exchange of some amount of asset X for asset Y (respectively Y for X) between a Liquidity Taker (LT) and the pool.

2. **Mint** operation - the addition of liquidity in the form of assets X and Y, executed by a Liquidity Provider (LP).

3. **Burn** operation - the removal of liquidity belonging to a LP, who previously performed a related mint operation.

For each one of the above events, there are conditions that need to be satisfied for the successful execution of the operation. These follow from the deterministic trading function under which a CPMM operates, which is of the form

$$f(x, y) = x \cdot y, \tag{5.1}$$

where x and y are units of assets X and Y that are locked inside the pool. These reserves x and y are provided by the LPs and define the current level of liquidity κ of the pool, i.e.

$$f(x, y) = x \cdot y = \kappa^2. \tag{5.2}$$

The trading activity of LTs cannot impact the level of liquidity of a pool, i.e. $\kappa = \text{const}$ after swap operations. This implies that

$$f(x, y) = f(x - \Delta x, y + \Delta y) = \kappa^2, \tag{5.3}$$

for the example of a LT selling Δy of asset Y in exchange of an amount Δx of asset X. As conceivable, we choose the signs in Eq. (5.3) to reflect variations in the reserves from the point of view of the pool, and this trading pressure gets incorporated in a variation of the exchange rate Z for the swap of the two assets. On the other hand, κ does change from the addition or removal of liquidity from LPs, while these latter operations are constrained to maintain Z constant.

To derive the marginal *instantaneous exchange rate* Z for an infinitesimal trade, we can compute

$$Z = \lim_{\Delta y \to 0} \frac{\Delta x}{\Delta y} = -\frac{\partial}{\partial y}\left(\frac{\kappa^2}{y}\right) = \frac{\kappa^2}{y^2} = \frac{x}{y}. \tag{5.4}$$

And the *execution rate* $\tilde{Z}(\Delta y)$ that a LT receives when selling a quantity $\Delta y > 0$ is

$$\tilde{Z}(\Delta y) = -\frac{\frac{\kappa^2}{y+\Delta y} - \frac{\kappa^2}{y}}{\Delta y}, \tag{5.5}$$

which suffers from the market impact of the trade.

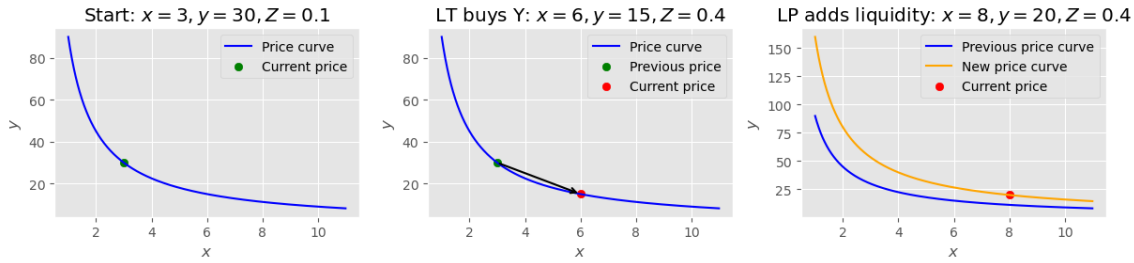


Figure 5.1: From left to right, example of the possible evolution of state of a liquidity pool. The price curve is plotted as the trading function at the current level of liquidity. First, we show the initial state of the pool. Then, we signal the variation of exchange rate after a LT executes a swap operation that buys asset Y. Liquidity is not affected at this point. Finally, we depict the increase in liquidity after a mint operation is completed by a LP. The latter has no impact on the exchange rate but shifts upwards the price curve.

To clarify the characteristic mechanisms that determine the evolution of state of a liquidity pool, we propose here an example, in which the market impact of trades is not taken into account for simplicity. Consider a liquidity pool that starts with 3 units of asset X and 30 units of asset Y, thus implying that the liquidity is $\kappa^2 = 90$ and the exchange rate is $Z = 0.1$. Then, a LT has 3 units of asset X that he wants to sell to buy asset Y. Using Eq. (5.3), we deduce that he will receive 15 units of asset Y. The new reserves thus become $x = 6$ and $y = 15$, making the price of asset Y increase to $Z = \frac{6}{15} = 0.4$, while the liquidity is fixed at $\kappa^2 = 90$. A LP then mints liquidity. She needs to maintain the exchange rate between assets constant and thus, she decides to provide 2 units of asset X and 5 units of asset Y. The new liquidity is now $\kappa^2 = 160$, meaning that there will be less slippage per unit of asset on a following trade. A representation of this example is shown in Fig. 5.1 by leveraging diagrams of trading functions.

5.2.2 Pool fees, impermanent loss, and gas fees

We have just described the essential mechanisms that govern both liquidity provision and consumption on a liquidity pool. However, there are two further components that need to be introduced, namely *pools fees* and *gas fees*. We now begin by providing information about the former, and this will also lead us to a brief digression on the important notion of *impermanent loss* (IL) of LPs.

In our derivations above, we have assumed that the whole quantity Δy sold by a LT is converted into some amount Δx according to the relative proportion of reserves of the pool. In reality, a LT needs to pay some fee γ proportional to the total quantity

Δy^{gross} that he wants to trade. This means that the net amount input to the swap operation is

$$\Delta y = \Delta y^{net} = (1 - \gamma) \cdot \Delta y^{gross}, \tag{5.6}$$

where the fee γ is characteristic of the pool. As an example, Uniswap v3 allows only $\gamma = \frac{\gamma_{perc}}{100}$ basis points, with $\gamma_{perc} \in \{0.01, 0.05, 0.3, 1\}$. Throughout our work, we refer to $\gamma \times 10^6$ as the *feeTier* characteristic of a pool (i.e. labels are $\{100, 500, 3000, 10000\}$).

The fees collected by the protocol, i.e. $\gamma \cdot \Delta y^{gross}$ from Eq. (5.6), are then distributed to LPs proportionally to the amount of liquidity they are providing. Thus, if a LP i provides liquidity for a weight $\frac{\kappa_i}{\kappa} = 0.1$ of the total amount of liquidity in the pool, then she will receive 10% of the total amount of fees accrued from the trading of LTs. This redistribution of pool fees is extremely important not only because it serves as an incentive for LPs to keep on providing their service, but it also aims to compensate them for the loss in value surely incurred when staking assets into a liquidity pool, namely their *impermanent loss*. LPs can always withdraw their funds from the reserves of the pool, but the amounts of assets retrieved depends on the fraction of liquidity provided by the LP, and not the initial exchange rate. By comparing the evolving value of assets either staked in the pool or plainly held, one can show that a LP suffers a loss for any price deviations. However, this "impermanent" loss can indeed be overcome by the profit from the redistribution of pool fees.

Finally, we mention one last fixed cost that both LTs and LPs need to sustain, namely *gas fees*. Every operation (i.e. swap, mint or burn) must be validated on the blockchain and this implies paying a fee that is collected by the miners. This fee varies[1] according to the complexity of the operation requested by the user, the urgency to execution, and the current congestion of the network. Fees are paid in gas, which is paid in giga-wei ($1gwei = 10^{-9}ETH$). Thus, the price of gas varies with variations of the ETH price but it is shared across the blockchain. LPs and LTs have also the option to pay a further subjective "gas tip" to miners and increase the probability of their desired transaction to be included in the first new block mined on the blockchain, and not left pending for an indeterminate amount of time.

5.2.3 Concentrated Liquidity

Uniswap v3 claims to provide increased capital efficiency via the implementation of a new mechanism, named *concentrated liquidity* (CL). LPs are given the possibility to concentrate their liquidity by "bounding" it within an arbitrary price range. While in

[1] `https://crypto.com/defi/dashboard/gas-fees`

the general CPMM design liquidity is spread across the entire price range $(0, \infty)$, LPs can now approximate any desired distribution of liquidity on the price space. The only requirement is to use available *ticks* as limits of the range, where there is a tick at every price that is an integer power of $\sqrt{1.0001}$. Not all ticks can be initialised on every pool, since there is a mandatory spacing of $1, 10, 60, 200$ basis points (bps) for pools with fee $\gamma_{perc} \in \{0.01, 0.05, 0.3, 1\}$ respectively.

Liquidity concentrated on a finite price range $(Z_l, Z_u]$, with integers l and u for the lower and upper bounds respectively, is commonly referred to as a *position*. A position only needs to maintain enough reserves to support trading within its range, and therefore can act like a constant product pool with larger reserves within that range. The related reserves' curve indeed becomes

$$\left(x + \kappa\sqrt{Z_l}\right)\left(y + \frac{\kappa}{\sqrt{Z_u}}\right) = \kappa^2_{(l,u]}, \tag{5.7}$$

where $\kappa_{(l,u]}$ is the liquidity minted across the price range under consideration. When the price exits the defined range, the reserves of one of the assets must have been entirely depleted. The position's liquidity is no longer active and no more fees from LTs are earned by the LP. However, the liquidity can become active again if the price re-enters the range. CL improves the liquidity available in the proximity of the current exchange rate, since it leads to a more active management of liquidity by LPs. LPs are pushed to concentrate the majority of their liquidity at the current price to try to increase their relative share of the pool and consequently earn more fees from LTs. However, this rebalancing of liquidity is very costly due to gas fees, and forces LPs to also consider wider positions that require less frequent management.

5.3 Systematic selection of Uniswap v3 pools of interest

5.3.1 Empirical introduction to the ecosystem

Our first goal is to extract a tractable sub-universe of Uniswap v3 liquidity pools, in which the interconnectedness among such pools is maximised to capture broader dynamics within the ecosystem. This sub-universe will then be used to investigate the broader trading behaviour of market participants, and experiment in the objective of clustering these agents into interpretable classes.

At the time of writing, Uniswap v3 is the latest implementation of the Uniswap DEX. Uniswap v3 launched in May 2021, introduced the concept of concentrated

liquidity and allowed multiple feeTiers (i.e. $\{100, 500, 3000, 10000\}$). For each Uniswap version $N = 1, 2, 3$, we access the related liquidity pool smart contracts via Etherscan[2], and quantify the daily creation of new pools in Fig. 5.2. The dates of transition from Uniswap v1 to v2, and v2 to v3, are also depicted. It is interesting to notice that the previous protocols remain active after the transitions, but their liquidity can be easily moved to the new Uniswap versions via "Migrator" contracts. Although there is a total of $4,889$ pools directly created with UniswapVNFactory contracts, the majority of them are the result of the 2020-21 cryptomania and inflated creation of new tokens (see Fig. 5.3 for a proxy of Total Value Locked, TVL, in the protocol over time). This translates to many pools not containing any relevant amount of liquidity locked, but which do not disappear due to the immutability of the blockchain. On the other hand, we also expect wrapped calls to the Factory contracts and thus refer to the above as a lower bound to the number of pools created.

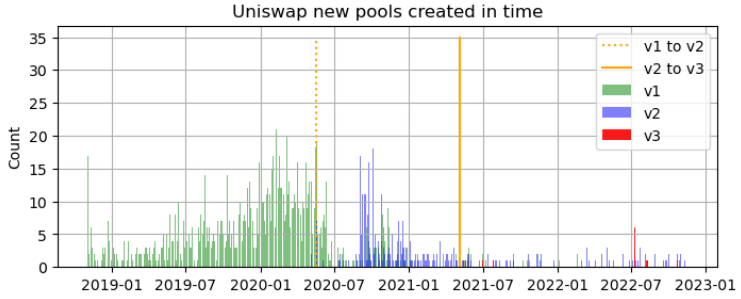


Figure 5.2: Daily count of new pools created via UniswapV1Factory, UniswapV2Factory and UniswapV3Factory smart contracts. The two vertical orange lines depict the dates of official transition from Uniswap v1 to v2, and from v2 to v3.

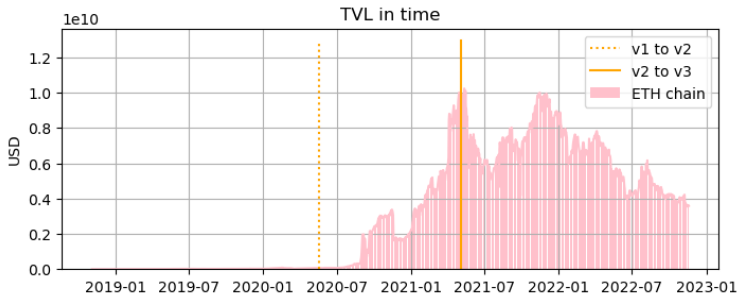


Figure 5.3: Evolution in time of the TVL in USD on Uniswap main Ethereum chain, where the data are downloaded from Defi Llama API. The higher the TVL, the more liquid the ecosystem is considered to be. In orange, we show the dates of transition from Uniswap v1 to v2, and from v2 to v3.

[2]`https://etherscan.io/`

5.3.2 Data download and coarse refinement of pools

"The Graph"[3] is a decentralised protocol for indexing and querying blockchain data. It allows to query data that are otherwise difficult to access, such as information from smart contract computations that is stored on the Ethereum blockchain. Thus, we construct Python scripts that connect to The Graph Uniswap v3 pointer[4] and download data on both the latest states of pools, and the full history of related swap, mint, and burn events. This is achieved via GraphQL queries that follow the Uniswap v3 own subgraph schema[5], meaning that we directly access "Entity" GraphQL objects with full features for the operations of interest. Our first aim is to identify the pools most representative of the Uniswap ecosystem, which we interpret as having significant liquidity consumption and provision events, but also showing high interconnectedness. To this end, we extract the latest summary data of all possible pools (from "Pool" GraphQL entities), full historical record of liquidity consumption operations (from "Swap" GraphQL entities), and full record of liquidity provision actions (from "Mint" and "Burn" GraphQL entities). Next, we develop a systematic approach that aims at increasingly discarding layers of pools with weakest features and dynamics. The filtered data set represents the starting point of our subsequent analyses, but aims at being useful to a wider group of researchers that desire to empirically investigate Uniswap v3.

We consider the data from Uniswap v3 liquidity pools as of 15 November 2022. We begin by applying a threshold of minimum $1,000$ transactions already happened on each pool, and recover an initial number of $1,344$ pools to consider. Then, we also restrict ourselves to pools where both exchanged tokens are traded in at least three pools (e.g. token T is traded against a stablecoin, against ETH and against ETH with different feeTier), in order to focus on interesting dynamics of the full ecosystem. The result is a subset of 696 Uniswap liquidity pools to further consider in our study. Uniswap v3 data start on 6 May 2021, when the transition from the previous version of the protocol successfully completed. While for the first months only the 500, 3000 and 10000 feeTiers were implemented, in November 2021 a fourth feeTier 100 was activated. This generated structural flows, noise and adjustments that we want to exclude from our analyses. Furthermore, we recognise that the transition of Uniswap's foundation blockchain (i.e. Ethereum) from Proof-of-Work to Proof-of-Stake in September 2022 could have triggered turbulences on the ecosystem too. Thus, we decide to focus our

[3] `https://thegraph.com/docs/en/about/`

[4] `https://api.thegraph.com/subgraphs/name/uniswap/uniswap-v3`

[5] `https://github.com/Uniswap/v3-subgraph/blob/main/schema.graphql#L1`

analyses on the six-months period from January 2022 to the end of June 2022, which we consider as the most representative of the actual DEX dynamics.

LP data filters. We extract liquidity provision data for these 696 pools, of which only 629 have non-empty entries due to issues on Uniswap end. Liquidity provision provides a historical record of all liquidity mint and burn operations on each pool, with related USD value. By computing the total cumulative sum of LPs activity, we proxy the TVL in USD that each pool contains at each point in time and denote it as "proxyTVL". Unfortunately, we cannot simply rely on the "PoolDayData" values provided by the subgraph due to incoherences found when cross-checking with Ethereum blockchain data on Etherscan. We then proceed to filter pools by requiring our proxyTVL to be larger than $1,000,000$ USD (one million dollars) at any point before the end of June 2022. This is motivated by the aim to find pools that were liquid enough at some point in our time window to capture interesting behaviour of LTs and LPs. We find 282 pools that satisfy this further requirement, as highlighted in the filtration summary diagram of Fig. 5.4.

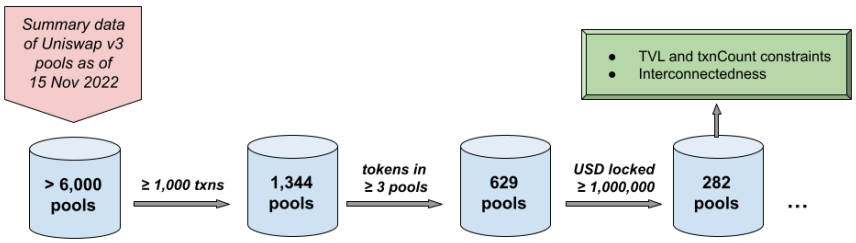
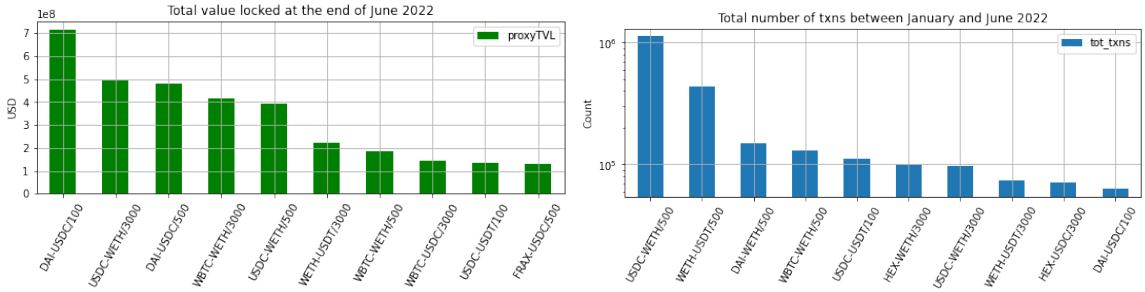


Figure 5.4: Summary diagram of the filtration steps pursued during our coarse refinement of pools. Stronger constraints on the TVL and txnCount of pools will follow, such as an attention to maximise the interconnectedness of the final sub-universe of pools.

LT data filters. To further lower the noise-to-signal ration in the data, we can subset our universe of interest to the pools with enhanced activity of LTs. As already motivated, we wish to focus on the six-months window $[January, July)$ 2022, which we denote as our case A. We also consider five sub-ranges, namely the two three-months windows $[January, April)$, $[April, July)$ that we denote as cases $B1/B2$, and the three two-months windows $[January, March)$, $[March, May)$, $[May, July)$, that we name cases $C1/C2/C3$. For each case and related time window $[start, end)$, we extract the pools with at least $1,000$ transactions before $start$ (where the number of transactions in time is calculated via the cumulative sum of both swap events and

mint or burn operations) and that also had at least $1,000,000$ USD in proxyTVL both at the *start* and *end* of the interval. Considering sub-ranges allows us to further account for the appearance of new pools that became significantly liquid or active after January 2022, or pools that lost the majority of their liquidity before July 2022, thus lowering survivorship-loser bias. For cases $A/B1/B2/C1/C2/C3$ in order, we find respectively 113/126/148/131/146/155 pools that satisfy the above requirements, for which we save the related blockchain addresses and information. Taking the union of these sets of pools, we notice that we are considering 177 different pools overall. Of these, five pools belong to the 100 feeTier, 28 pools to the 500 feeTier, 84 pools to the 3000 feeTier, and 60 pools to the 10000 feeTier.

To gain a brief insight into the most liquid and active venues, we consider the pools extracted for case A and plot in Fig. 5.5a the 10 pools with highest proxyTVL at the end of June 2022, and in Fig. 5.5b the 10 pools with highest total number of transactions over the six months of relevance. As a convention, we refer to pools with the format "SYMBOL1-SYMBOL2/feeTier", where we use the trading symbols of the two tokens exchanged by the pool. Stablecoins, wrapped Ether (WETH) and wrapped Bitcoin (WBTC) dominate the landscape of tokens swapped in the most liquid and active venues, which is expected since they are the oldest, most established, or safest cryptocurrencies that agents can trade and develop strategies onto.



(a) Pools with highest liquidity at the end of June 2022.

(b) Pools with highest total number of transactions during case A.

Figure 5.5: The 10 most liquid and active pools for case A, i.e. over the time window between January and June 2022.

5.3.3 Network-driven filters

Above, we provided methodical steps needed to complete an initial filtration of the extremely large set of Uniswap v3 pools. Now, we propose a series of network-based measures that aim to be used to further subset liquidity pools into a universe with

maximised interconnectedness, as desired before proceeding to the clustering of market participants.

Filtering of pools by basic interconnectedness measures. For each one of our cases $A/B1/B2/C1/C2/C3$, we build a weighted graph $G = (P, E)$. The set of nodes P denotes relevant pools, and edges $(p, q) \in E$ with $p, q \in P$ have weights w_{pq} that encode a measure of similarity, as possibly defined below. We start by considering two possible different measures of such similarity between pools:

1. Number of common LTs (*or* LPs) active on both pools, which are identified by the entry "origin" in the Uniswap data.

2. Number of common smart contracts, i.e. "senders" in the Uniswap data, called by origins to execute swap transactions (*or* to execute liquidity provision operations).

We separate between a focus on liquidity consumption or provision in the above measures, since the two dynamics differ substantially. Despite we aim to focus on LTs, we also provide related results for a reader with main interest on LPs. Of course, the intersection or union of the results can be then also used to pursue broader analyses. To clarify the Uniswap terminology adopted above, we remind that every "swap action" is initiated by an *origin* O, it then calls a smart contract referred to as *sender* S, and ends to the *recipient* R. In liquidity provision, only the origin and sender of operations are relevant. Figure 5.6 shows the distribution of number of origins, senders and recipients in each pool for both LT and LP data over the time window of case A. We also show the distribution of the intersection between origins, senders and recipients' addresses.

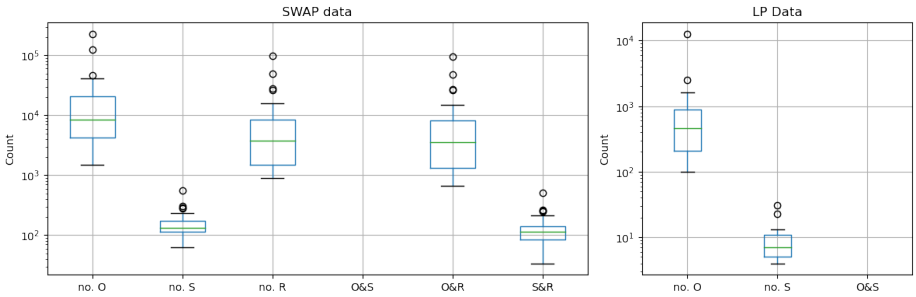
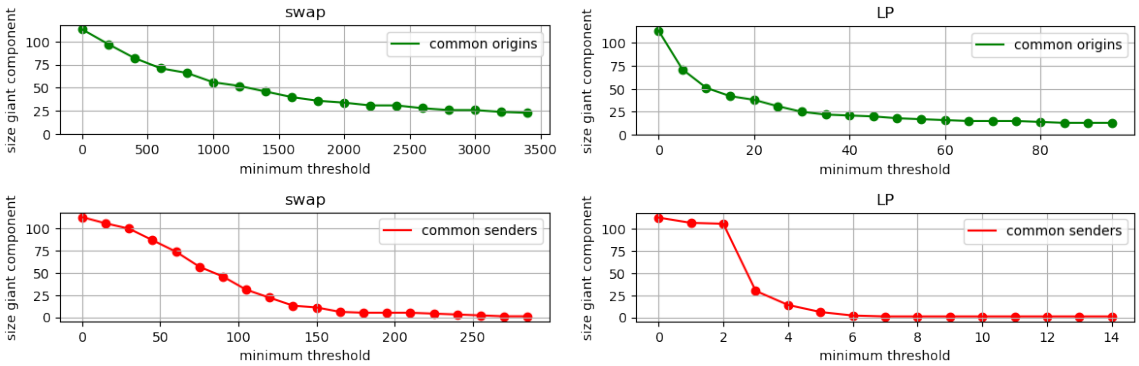


Figure 5.6: The intersection of origins and senders is always zero since the former are wallets of users and the latter smart contracts. Recipients can instead be both, hinting to more complex patterns in the execution of transactions. To be precise, here we are specifically considering the sub-universe of pools relevant for case A at the end of all our refinements.



(a) LT data. (b) LP data.

Figure 5.7: Evolution of the size of the giant component for graphs of pools in case A, when varying the threshold of common origins and senders for (a) swap transactions, (b) liquidity provision operations.

We now proceed to studying the relationship between the size of each graph's giant component and a minimum threshold on the value of the measure used to create the link between each pair of pools. After fixing a threshold, we consider the pools in the related giant component as our relevant interconnected sub-universe. Figure 5.7 shows the variation in size of the giant component for case A, when modifying the minimum number of common origins or senders for LT and LP data. We aim at considering the tails of the distributions for each case (i.e. time interval), which amounts to $\sim 20-30$ pools in each instance, to retain the most significant connections and possible dynamics of the ecosystem. For case A, this results in the choice of thresholds $2,000$ and 100 for minimum common origins and common senders respectively, on the LT data. Similarly, we choose thresholds 30 and 3 for minimum common origins and common senders respectively, on the LP data. Finally, we consider the intersection of survival pools for the two graphs generated by LT data, and find 27 common pools (out of the 34 and 36 pools, respectively in each graph). For LP data, we find a number of 19 final relevant pools (from the intersection of 25 and 30 pools). The full pipeline is repeated for cases $B1/B2/C1/C2/C3$.

Enhancement of pools for liquidity consumption analysis. Ideally, we should also consider the flow of funds across pools and find the related most interconnected graph. However, this is intractable if using only Uniswap data and not the full list of Ethereum blockchain transactions. Indeed, LTs are active across different DeFi protocols and can easily move liquidity from one venue to another and back. We propose an approximation to the problem by taking advantage of the fact that each

trader's transaction can include more actions, which happen "instantaneously but in order" when the full transaction is validated. Thus, if a LT executes two swaps of the form $X \to Y, Y \to Z$ for tokens X, Y, Z in one same transaction, then we interpret Y as a *bridge* between the action of selling X to buy Z. We view this as an indication of the flow of (smart) money between pools and of possible arbitrage opportunities, relevant to the LT sub-universe. In summary, we complete the following steps:

1. Merge all LT data before the interconnectedness analyses, e.g. data for the 113 pools of case A.

2. Keep all the transactions for which there are at least two inner actions, i.e. same "transaction id" but different "logIndex" in Uniswap terminology.

3. For each resulting transaction:

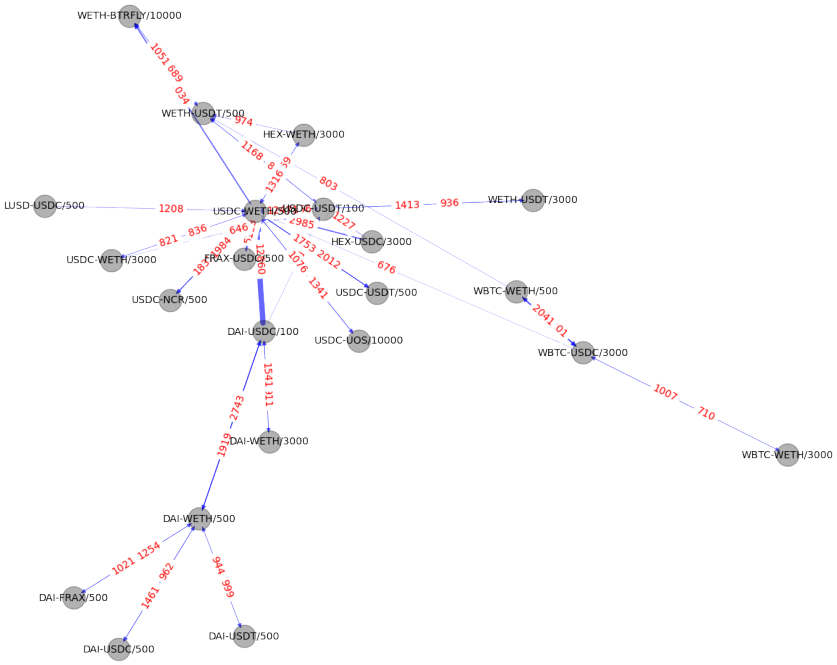
   (a) For each token that appears in the transaction actions, keep a *flow list* of related buying (-1) or selling $(+1)$ trades in all the related pools by looking at the sign of the amount swapped by the pool.

   (b) For each token, consider its flow list and find all the occasions when a -1 is immediately followed by a $+1$ (i.e. the token was first bought in a pool and then sold in another pool, acting as one of our bridges).

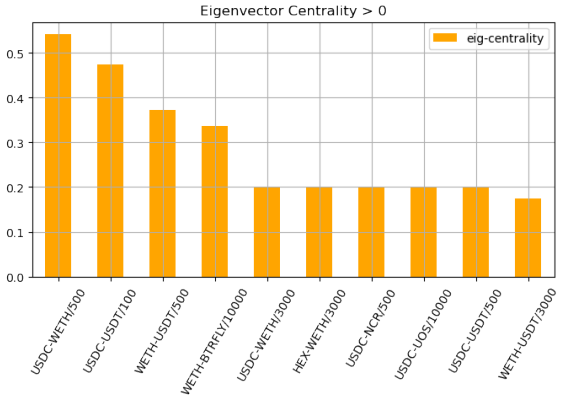
   (c) Save this occurrence of a flow between pools as a *bridge transaction*,

where we are approximating only jumps of length one. As an example, for a flow list of the form $[-1, +1, +1]$, we only consider the flow as one from the first pool to the second one. For more specific analyses, one could consider the specific amounts traded and check the relative proportions exchanged from the first pool to the second and third ones, but this is outside the scope of our current investigation.

We extract all bridge transactions between pools and create a directed graph for each one of our temporal cases. Nodes are pools as usual, and edges are built for each pair of pools that have at least some number of bridge transactions between them. Of course, each pair of pools can have up to two edges between them according to the direction of related bridge transactions. Then, we keep the giant component from the undirected version of the graph and add the resultant set of nodes to the LTs pools saved from the previous interconnectedness analyses. For case A, we require at least 800 bridge transactions between two pools to create the related edges. The resultant giant component (see Fig. 5.8a) has 22 nodes, seven of which were not included in our LT set of pools from the previous analyses and are thus added. Figure 5.8b highlights



(a) The resulting giant component, with edge weights for both directions.



(b) Pools with highest eigenvector centralities.

Figure 5.8: Results from our bridges investigation for case A, which covers the six-months window from January to June 2022. If a LT executes two swaps $X \to Y, Y \to Z$ one after the other (for tokens X, Y, Z), then we interpret Y as a bridge between the action of selling X to buy Z. We save all pairs of pools for which there is a common token that acts as a bridge, with the related number of occurrences of bridge transactions. Then, we create a directed graph where nodes are pools and edges are built for each pair of pools that have at least 800 bridge transactions between them.

the nodes with highest eigenvector centrality (see Eq. (3.5) in Chapter 4) in the graph, where we can especially notice how several pools of WETH against a stablecoin are proposed. This is intuitively sensible, since LTs can take advantage of routing to complete specific re-balancing of tokens via more liquid and favourable pools, which tend to have stablecoins, WETH and WBTC as their tokens, as shown in the earlier analyses.

Thus, our framework finally proposes a set of 34 pools to consider for LTs analyses for case A, and a related set of 19 pools for LPs analyses. The full lists of pools are reported in Appendix C, where we also propose our filtering results for cases $B1/B2/C1/C2/C3$.

5.4 Structural investigation of Uniswap v3 ecosystem

5.4.1 Clustering of Liquidity Takers

The DeFi ecosystem has grown increasingly complex in recent years. The first step to shed more light on its intrinsic features and dynamics is to better understand its own components, which is what motivates the following empirical investigation of LTs trading behaviour on Uniswap v3. This is a non-trivial task, for a number of reasons. First of all, agents can easily generate numerous crypto wallets, and hence in some sense, "multiply" their identities to hide or obfuscate their full behaviour. Their actions are then generally spread over a broad set of possible pools, vary significantly in size both within and across different types of pools, and also happen with evolving frequencies over time. Due to the high interconnectedness of the ecosystem and noticeable volatility across cryptocurrencies (which is of interest to speculators), we believe that it is essential to consider the broad trading behaviour of agents across pools, which we can easily do thanks to the transparency of our Uniswap data. The filtering of pools just completed thus becomes of immediate use, since it points to the most important venues that characterise agents' trading behaviour. However, this choice forces us to diverge from the literature on clustering order flow discussed in Section 5.1, because such studies do not account for the contemporaneous exposure to multiple different assets. So, we will soon propose a novel method to express and cluster the structural trading equivalence of agents active on multiple environments, by leveraging on both network analysis and NLP techniques. Nevertheless, we will also gauge whether the unravelled clusters can be interpreted with proxies of some of the mentioned main discerning factors of LOB order flow (i.e. volume and time of submission, similarly to [45], [34] and [131]). We will also build features from the

decentralised ecosystem of reference, to better characterise the groups identified and extract insights on the main types of agents present. Importantly, we are not able to compute the inventory of agents as done in LOB models, since traders tend to be active on multiple DeFi protocols for the same assets, but we only see transactions specific to Uniswap.

5.4.1.1 Overview and pre-processing

We focus on the LT data for our three longest periods $A/B1/B2$, for the related pools identified in the previous section. For each case, we first look at the distribution of the total number of transactions performed by the different LTs over each full time window. As an example, the distribution for case A is represented by the blue bars in Fig. 5.9. We then require a minimum number of transactions completed by each LT, since considering only a very small sample of trades per agent does not provide meaningful structural information on their behaviour. Thus, we impose a lower bound of a minimum average of 10 transactions per month that each LT must have completed. On the other hand, we manually define maximum thresholds to remove only extreme singular outliers from each distribution for computational purposes. The initial total distribution for case A is shown in Fig. 5.9, where we also highlight how it changes when requiring a minimum number of transactions equal to 25 (orange bars), and when we require our final minimum and maximum thresholds of 60 and $15,000$ total number of transactions, respectively (green bars). For cases $B1/B2$, we require the range 30 to $5,000$ transactions for the former, and 30 to $11,000$ transactions for the

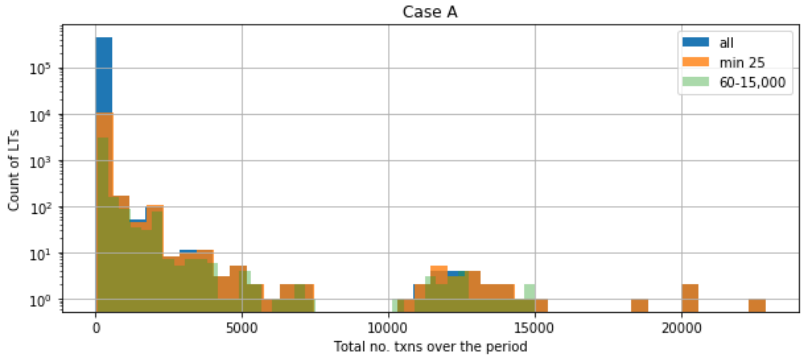


Figure 5.9: Distribution of total number of transactions (txns) performed by LTs during case A. We show the full distribution, the result after requiring a minimum of at least 25 transactions, and the distribution after applying thresholds of minimum 60 and maximum $15,000$ transactions. The latter scenario results in our final set for case A, which comprises $3,415$ LTs. A small cluster of LTs much more active than others is already discernible.

latter. Overall, we find a number of LTs approximately between $3,500$ and $5,000$ for all our periods $A/B1/B2$. This altogether defines the final sets of LTs along with their transactions. Next, we proceed to define and compute their embeddings, which are subsequently used for the final clustering stage.

5.4.1.2 Methodology

Graph2vec. The field of NLP studies the development of algorithms for processing, analysing, and extracting meaningful insights from large amounts of natural language data. We already extensively discussed it in Chapter 3, but further examples of its myriad of applications include question answering, language translation, and speech recognition. One of the already mentioned turning points in NLP was the development of the word2vec word embedding technique [90], which considers sentences as directed subgraphs with nodes as words, and uses a shallow two-layer neural network to map each word to a unique vector. Taking inspiration from this idea of preserving knowledge of the context window of a word in its embedding, the node2vec algorithm [61] (fully introduced in Section 3.3.3 of Chapter 3) learns a mapping of nodes in a graph to a low-dimensional space of features by maximising the likelihood of preserving network neighbourhoods of nodes.

By taking a further step towards general language representations, [77] proposes the unsupervised algorithm *Paragraph Vector* (also known as doc2vec), which learns continuous fixed-length vector embeddings from variable-length pieces of text, i.e. sentences, paragraphs and documents. The vector representation is trained to predict the next word of a paragraph from a sample of the previous couple of sentences. Both word vectors and paragraph vectors models need to be trained, which is again performed via SGD and backpropagation.

As doc2vec extends word2vec, graph2vec [101] is a neural embedding framework that aims to learn data-driven distributed representations of an ensemble of arbitrary sized graphs. The authors propose to view an entire graph as a document, and to consider rooted subgraphs around every node in the graph as words that compose the document, in order to consequently apply doc2vec. This approach is able to consider non-linear substructures and has thus the advantage to preserve and capture structural equivalences. One necessary requirement to pursue this analogy is for nodes to have labels, since differently labelled nodes can be then considered as different words. These labels can be decided by the user, or can be simply initiated with the degree of each node.

Thus, graph2vec considers a set of graphs $\mathcal{G} = \{G_1, G_2...\}$, where the nodes S of each graph $G = (S, T, \lambda)$ can be labelled via the mapping function $\lambda : S \to \mathcal{L}$ to the alphabet $\mathcal{L}$. The algorithm begins by randomly initialising the embeddings for all graphs in the set $\mathcal{G}$, then proceeds to extract rooted subgraphs around every node in each one of the graphs, and finally iteratively refines the corresponding graph embedding in several epochs via SGD and backpropagation, in the spirit of doc2vec. The rooted subgraphs act as context words, which are used to train the paragraph (i.e. graph) vector representations. Subgraphs are extracted following the Weisfeiler-Lehman (WL) relabeling process [119]. The intuition is that, for each node in a graph, all its (breadth-first) neighbours are extracted up to some depth d. Labels are then propagated from the furthest nodes to the root one, and concatenated at each step. In this way, a unique identifier for each node is identified from its "context" and the full set can be used to train an embedding for the graph. The optimisation problem thus becomes

$$\max_{f'} \sum_{G \in \mathcal{G}} \sum_{s \in S} \log Pr(g^d_{WL}(s) | f'(G)), \tag{5.8}$$

where the aim is to maximise the probability of the WL subgraphs given the current vector representation of the graph. Here, f' is a mapping function of graphs to n-dimensional representations, and g^d_{WL} are WL subgraphs with depth d.

A modification of graph2vec for LTs embedding. For each one of our cases $A/B1/B2$, we consider all the related LTs and their full set of transactions on the sub-universe of LTs' pools of relevance. We then introduce the concept of a *transaction graph* G_{txn}, which we use to represent the behaviour of each active agent.

Definition 5.4.1 (Transaction graph)**.** A *transaction graph* $G_{txn} = (S, T, W)$ is the complete weighted graph where nodes S are the swap actions that the LT under consideration has executed, and edges $(s, r) \in T$ with $s, r \in S$ are built between every pair of nodes. Each edge has a weight $w_{sr} \in W$, which encodes the amount of time Δt (in seconds) elapsed between the two transactions s, r. Each node $s \in S$ has a label l_s from the alphabet $\mathcal{L}$, which uniquely identifies the pool that the swap was executed into. Importantly, $\mathcal{L}$ is shared among the full set of LTs and related transaction graphs.

Labels in the alphabet $\mathcal{L}$ differentiate between swaps executed on different pools, i.e. pools with unique combination of tokens exchanged and feeTier implemented. This implies that the algorithm receives as input only general identifiers of pools. Thus,

we can consider intuitive differences (e.g. expected volatility of the exchange rate on pools of stablecoins versus on pools of more exotic tokens) only afterwards, when assessing and investigating the meaningfulness and interpretability of the extracted clusters.

We now have a set of graphs representing LTs, and our aim is to find a n-dimensional vector representation of each one of its elements. We cannot plainly apply the graph2vec algorithm, since the concept of neighbours of a node is irrelevant in a complete graph. Thus, we modify the graph2vec mechanism to take advantage of the weight that the different links between nodes have, while maintaining the overall intuition. For each node $s \in S$ of a graph G_{txn}, we sample a set of neighbours $N_{txn}(s)$ by generating random numbers from a uniform distribution between $[0, 1]$, and comparing them to a *sampling-value* for each edge outgoing from the node.

Definition 5.4.2 (Sampling-value)**.** The *sampling-value* $M(w_{sr})$ of an edge $(s, r) \in T$ with weight $w_{sr} \in W$ in graph $G_{txn} = (S, T, W)$ is computed as

$$\begin{aligned} M(w_{sr}) &= \frac{H(f^{scal}(w_{sr}))}{H(f^{scal}(\min W))}, \\ \text{with} \quad H(w_{sr}) &= \sqrt{\frac{2}{\pi}} \exp \frac{-w_{sr}^2}{2}, w_{sr} \geq 0, \\ f^{scal}(w_{sr}) &= \frac{w_{sr} - \min W}{(\max W)/|S|}, \end{aligned} \tag{5.9}$$

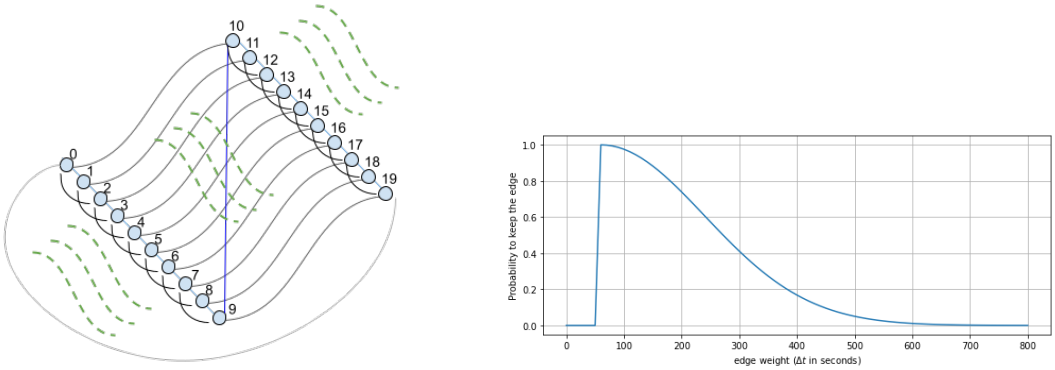
where we are using a half-norm that is shifted and scaled to adapt to each LT's extreme features, i.e. $\min W$ and $\max W$. The final sampling-value is also normalised to impose a value of $M(\min W) = 1$, meaning that the shortest link(s) in the graph is chosen with probability 1 (of course, only if it is involved in the current node under consideration).

If the random number is below the sampling-value, then the link is kept and the associated node added to $N_{txn}(s)$. In this way, the probability of an edge to be chosen is inversely proportional to its weight Δt, and the sub-structures kept represent clustered activity in time.

After having generated the set of $N_{txn}(s), \forall s \in S$, we perform WL relabeling and proceed as in the vanilla version of the graph2vec algorithm. We set all the hyperparameters to their default values, i.e. number of workers = 4, number of epochs = 10, minimal structural feature count = 5, initial learning rate = 0.025, and down sampling rate of features = 0.0001. The only exception is the number of WL iterations, which in our case must be set to 1 instead of 2. The result is an embedding for each

graph in our set of transaction graphs, which becomes a set that we can subsequently cluster via the K-Means++ methodology. Importantly, we want to underline that our embeddings and clusters do not depend on the real magnitudes of weights Δt, since the sampling is adjusted on that. In addition, they also have no notion of the amount of USD traded, thus being agnostic to the transaction value. As a final note, we refer the reader to [39] for a version of graph2vec that uses edge labels. However, the algorithm creates the dual version of the graph and would not be effective in our case, thus providing ground for our proposed extension.

An illustrative example. To clarify our approach, we describe a simple example. Consider an agent that executes 20 transactions. She executes the first 10 transactions shortly clustered in time, waiting only 60 seconds one after the other. Then, she waits 42 minutes to action on the final 10 transactions with, again, a frequency of one minute. Her behaviour is plotted in the transaction graph G_{txn} of Fig. 5.10a, where we assume for simplicity that each transaction is performed on the same pool and thus colour-code nodes all the same. We also number nodes to show the order in which the related transactions are executed. We do not draw all the edges of this complete graph for clarity and ease of visualisation, but hint with the green dashed lines that indeed there are more connections to be remembered. In this example, the minimum time between transactions is 60 seconds (light blue edges) and the maximum one is one hour, i.e. $3,600$ seconds (light grey edge). Some intermediate times are depicted as edges with the same colour for the same weight. The resultant sampling-value function $M(w)$, which defines our sampling probabilities to choose edges, is shown



(a) Transaction graph G_{txn}.

(b) Sampling-value $M(w)$.

Figure 5.10: For our illustrative example, we show in (a) a simplified representation of the LT's transaction graph, and in (b) the sampling-value that defines probabilities of keeping edges as neighbours.

in Fig. 5.10b. As intuitively desired, we aim at always keeping the shortest edges in the graph and these have indeed probability 1. Then, we also aim to keep the most clustered "communities", and indeed, we observe from the plot that transactions five minutes away are still chosen with 40% probability, but longer times are very easily dropped.

5.4.1.3 Results

For each case $A/B1/B2$, we study the structural equivalence of LTs' trading activity by clustering the representations generated via our modified graph2vec algorithm. Focusing first on case A, we compute embeddings for dimensions $n \in \{8, 16, 32, 64\}$, and confirm with PCA that the proportions of data's variance captured by different dimensions are well-distributed in the most conservative cases. For each n-dimensional set of vectors, we then group LTs by performing a series of K-Means++ clusterings with different number of desired groups. We compute the inertia of each partition found, and choose the optimal clustering via the usual *elbow method*. The similarity between optimal clusterings for different dimensions is then computed, in order to investigate the stability of results across representations of increasing dimensionality. We achieve this by computing the related ARIs [68] as introduced in Eq. (3.24). We find ARIs for clusterings on 8-vs-$\{16, 32, 64\}$ dimensional data around 0.75, while clusterings on 16-vs-32, 16-vs-64 and 32-vs-64 dimensional data reach approximately the value of 0.90. Therefore, we conclude that there is a high stability of results when our data are embedded at least in 16 dimensions, and use the related 16-dimensional vector representations for our final analyses. The related optimal number of clusters of LTs for case A is seven. Similar results arise for cases $B1/B2$ too, and the related optimal numbers of clusters of LTs are six and seven, respectively.

Each extracted clustering is based on the structural similarity of LTs' trading behaviour. To judge the goodness of our modified algorithm and assess the results, we investigate whether there are specific features or trends that are highly representative of only some of the groups. Thus, we proceed to computing a set of summary statistics for each LT, and calculate the average of these results over the LTs belonging to each different group. The features that we consider are:

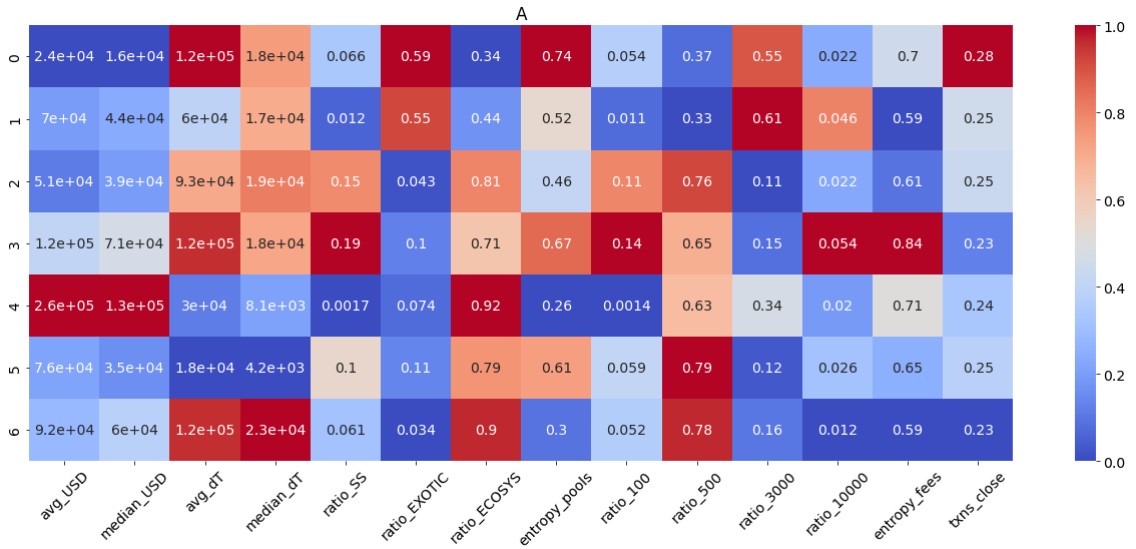
- average and median USD traded,
- average and median time Δt in seconds between transactions,
- ratio of transactions done in "SS", "EXOTIC" or "ECOSYS" pools, and entropy,

- ratio of transactions done in pools with a specific feeTier, and related entropy,
- ratio of trades on days when the SP LargeCap Crypto Index[6] increased/decreased in value, or when the market was closed, due to weekends and bank holidays.

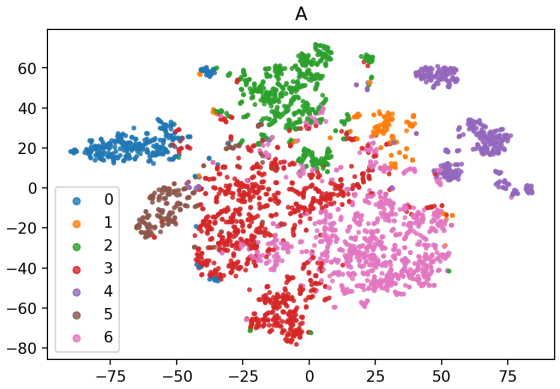
The first two points above are inspired from the literature on clustering of LOB order flow, i.e. [45], [34], and [131], which highlights the importance of volume and time of submissions to characterise types of agents active on centralised exchanges. The distinction between "SS", "EXOTIC" or "ECOSYS" pools is then inspired by the classification in [65], where the authors introduce a notion of normal pools, stable pools and exotic pools. For them, stable pools exchange tokens that are both stablecoins. Normal pools trade instead tokens that are both recognised in the crypto ecosystem, while exotic pools deal with at least one token that is extremely volatile in price (e.g. YAM, MOON and KIMCHI). We slightly divert from this classification and define "SS" pools as pools whose tokens are both stablecoins, "ECOSYS" pools as pools that exchange only tokens that are either stablecoins or pegged to the most established BTC and ETH coins, and "EXOTIC" pools as the remaining ones. ECOSYS pools can be seen as the venues carrying the "safest" opportunity for profit for a novice crypto investor, since they trade volatile tokens though directly related to the most established blockchains that are the true foundations of the whole DeFi environment.

The average magnitude of features computed over the LTs belonging to each different cluster for case A, i.e. over Jan-June 2022, is reported in Fig. 5.11a. We focus on the groups found specifically for this period because it is the longest one and thus, it provides us the most general results and insights. Cases $B1/B2$ will be later described too, in order to assess the overall stability of recovered *species* of LTs and highlight any specific variations due to different sub-periods in time and related pools of relevance considered. The seven clusters of LTs found have sizes of 304/142/512/978/379/186/914 agents respectively, which means that we are able to find a well-balanced distribution of cluster sizes without any extremely dominant group. Thanks to the heatmap in Fig. 5.11a, we also easily confirm that our methodology is able to extract different groups of LTs that have significant variation of behaviour with respect to the outer features defined. However, a few columns had to be dropped due to non-significance of their results. Importantly, we also recall that inner biases on ratios are present (e.g. when considering that our sub-universe does not have a uniform distribution of numbers of pools with specific feeTier), and thus we can expect

[6] `https://www.spglobal.com/spdji/en/indices/digital-assets/sp-cryptocurrency-largecap-index/#overview`



(a) Average features for LTs in case A.



(b) t-SNE embedding of case A.

Figure 5.11: Clustering of LTs for case A, i.e. over the six-months between January and June 2022. In (a), each row represents one of the recovered clusters and columns are the different features computed to characterise species of LTs. The color-code employed applies to each column separately to be able to quickly identify the related smallest and biggest values in magnitude, and judge the general distribution. It is essential to check the magnitudes of cells per se too, due to highly variable variance between columns. In (b), the t-SNE plot of embeddings of LTs is reported with perplexity = 15 and points are color-coded according to their cluster of membership.

more/less transactions of some type on average. For visualisation purposes, we also embed the 16-dimensional representations of LTs into a 2-dimensional view via t-SNE, and plot them with perplexity = 15 in Fig. 5.11b. LTs are colour-coded according to the cluster they belong to, and we indeed observe that different groups lie on different parts of the plane.

Focusing on Fig. 5.11a, one can draw the following high-level remarks:

- **Groups 0 and 1** have a strong focus on trading exotic cryptocurrencies. The former is a set of LTs that mainly uses feeTier 3000 for the purpose, and shows slightly higher than average tendency to trade when the market is closed. The latter group uses significantly both the 3000 and 10000 feeTiers, meaning that the related LTs are willing to accept also extremely high transaction costs. This behaviour could indicate that they have high confidence on their intentions and possibly urgency.

- On the other hand, **groups 2 and 3** trade stablecoins more than usual. The former cluster could point to an enhanced use of SS pools to take advantage of optimised routing, while the latter has a non-negligible proportion of trades in exotic pools with feeTier 10000. Likely, group 3 isolates a set of LTs that are interested in niche exotic tokens, which are only proposed in pools against stablecoins that do not overlap. Diverting funds between two of these exotic tokens requires an exchange between the two related stablecoins too, which motivates the recovered statistics. We also witness strong usage of the feeTier 100, which hints to traders trying to compensate the high costs suffered in pools with feeTier 10000 by paying the lowest possible fees on the SS pools.

- **Groups 4 and 6** are more active than average on ECOSYS pools. The two groups differ noticeably from their opposite relative strength of USD traded and time between operations. Overall, group 6 trades less money and waits longer, mainly using pools with low feeTier 500. These features can be interpreted as characteristics of cautious retail traders that invest in less risky and highly well-known crypto possibilities. And indeed, we also find that this group is one of the largest in size. Then, group 4 also relates to ECOSYS pools. However, these users tend to trade more USD with higher frequency, and this is also the cluster with much higher than average proportion of LTs that also act as LPs ($\sim 16\%$) . Therefore, we identify here a group of more professional investors.

- Finally, **group 5** shows a significant usage of all the three types of liquidity pools, but trades are concentrated in pools with cheap feeTier 500. These agents trade often, and indeed show the smallest median time between transactions. These eclectic, active and thrifty LTs are probably our group of smartest investors.

Our results confirm that the proposed algorithm is able to recognise variance in the data, and allow us to extract interesting insights into the behaviour of different types of species of LTs. In particular, we observe how the type of pools on which LTs are active plays a primary role in the definition of their trading behaviours. This is especially interesting since no full notion of tokens and feeTier is used in the generation of the embeddings. Indeed, only a unique label per pool is provided as input to our algorithm, e.g. USDC-WETH/500 could be pool "P1", USDC-WETH/3000 pool "P2" and FXS-WETH/10000 pool "P3". Thus, these pools would be considered equally different if no structural discernible pattern was recognised by the methodology, providing some further evidence of the strength of our proposed extension to *graph2vec*.

Stability analyses. As already motivated, we now pursue the same analyses described above but for cases $B1/B2$. We cluster the n-dimensional embeddings for $n \in \{8, 16, 32, 64\}$ and compute the ARIs between each pair of resultant sets of LT groups. We confirm that at least a 16-dimensional embedding is required in order to have a stability of clusters for case $B1$, while only eight dimensions suffice for the case $B2$. For simplicity, we use the 16-dimensional representations consistently in all cases. We recover six groups of LTs for case $B1$, and seven for case $B2$. In both cases, we find two clusters with same characteristics as groups 4 and 6 of case A, i.e. traders mainly active on ECOSYS pools. We also recover the eclectic traders of group 5. Therefore, we observe several stable and persistent types of LTs. Small perturbations happen instead on the groups trading on SS or EXOTIC pools, as one could expect from the mere evolution of time and external market conditions, and consequently generation of different behaviours. In particular, all case A species, except group 1, are also found in case $B1$. On the other hand, case $B2$ shows less intensity on group 3, probably due to investors diversifying more during the crypto turmoils of the second quarter of 2022. Overall, we observe general agreement on the groups and main features recovered during cases $A/B1/B2$, and we can thus rely on our species of LTs found for the longest duration case A as general descriptors of the ecosystem.

The above finding on the stability of species over time is of interest in itself. As an example, central banks started hiking interest rates in March 2022 and this consequently stopped a strong influx of liquidity into the crypto ecosystem. This also accentuated a period of significant underperformance, and could have weakened the stability of results from a stronger difference in trading activity. On top of that, the Terra-Luna crash happened in May 2022 and it could have in theory enhanced noise and instabilities especially in the structural clustering on case $B2$. As a very

last remark, we notice that only $\sim 20\%$ addresses are present in all cases $A/B1/B2$. Therefore, we are either recovering similar behaviour but for different people, or in some cases it could be the same person simply employing a new wallet to better hide their trading behaviour.

While we have clearly focused on the liquidity consumption component of the crypto ecosystem thus far, we shift our interest from LTs to pools in the next step of our investigations. We will indeed perform a clustering of pools based on features built from simple statistics that consider both liquidity consumption and liquidity provision. This will allow us to assess whether the SS, ECOSYS and EXOTIC classification is beneficial for describing the crypto ecosystem or is only useful for LTs characterisation.

5.4.2 Clustering of pools

The above analyses revealed a characterisation of the main types of LTs structural trading behaviour. While the importance of different types of pools in the ecosystem seems to be also clear, we stress that a full understanding of liquidity pools goes beyond the mere liquidity consumption mechanism (i.e. it needs to further account for both liquidity provision and price evolution). Thus, we now pursue an intuitive initial investigation on the similarity of pools themselves, in order to gain additional insights on the entire ecosystem.

5.4.2.1 Methodology

We focus on case A, as it covers the longest period of time. We consider the intersection of pools relevant for both LTs and LPs to properly account for both mechanisms, and find a resulting set of 16 pools. For each pool, we compute the following 13 features:

- average daily number of active LTs/LPs - "SdailyLT" and "LdailyLP" respectively,
- volatility of the execution price of the pool - "SstdP",
- average size of swap/mint/burn operations in dollars - "SavgUSD", "LavgUSD-mint" and "LavgUSDburn",
- average daily amount of dollars used in swap/mint/burn operations (i.e. volume) - "SdailyVol", "LdailyVolMint" and "LdailyVolBurn",
- average daily number of LTs/LPs transactions - "SdailyTxn" and "LdailyTxn",

- average daily number of different senders, i.e. smart contracts, called within swap transactions - “SdailyS”,

- number of agents with only one transaction normalised by the number of days considered - “Sdaily1txn”. This measure is computed to gauge the tendency of external smart investors to hide their behavior by creating several different wallets on the pool.

For the above features, we create related labels for ease of reference, which start with letter “S” if the quantity is computed from swap operations, or letter “L” if the quantity is computed from liquidity provision operations. In Fig. 5.12, we show the heatmap of Spearman correlations between the above attributes plus feeTier (“SfeeTier”) for our pools. There are significant positive correlations, especially among features developed from LT data and LP data, respectively. Thus, we standardise entries and employ linear PCA and kernel PCA (with both “rbf” and “cosine” kernel in the latter) to reduce the dimensionality of our data. The eigenvalue decay for all three mentioned cases is shown in Fig. 5.13, where only the first seven eigenvalues are depicted for clarity of visualisation. The cosine kernel PCA is seen to capture more variance in fewer dimensions, and thus we embed the data by projecting on its related first three components. The resulting 3D embedding is shown in Fig. 5.14 from three different angles, where we color-code pools according to their feeTier. In particular, green relates to feeTier 100, blue to 500, orange to 3000, and red to 10000.

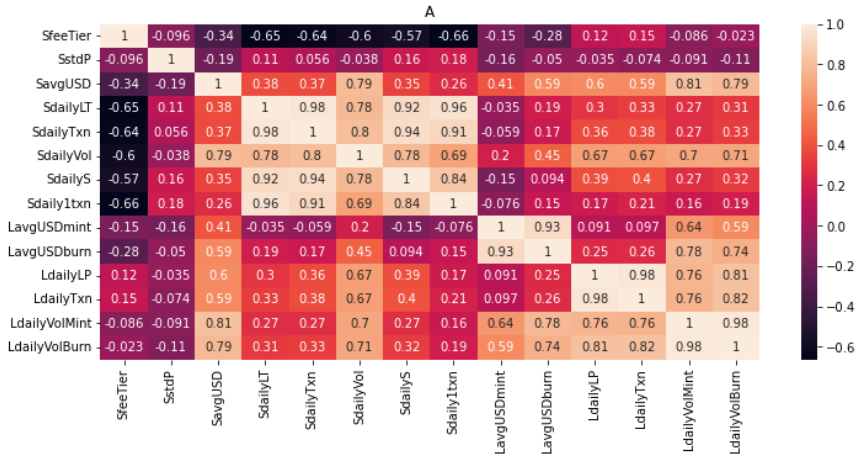
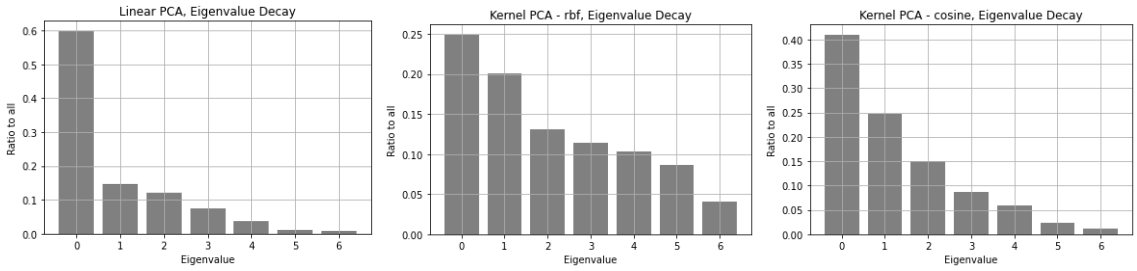
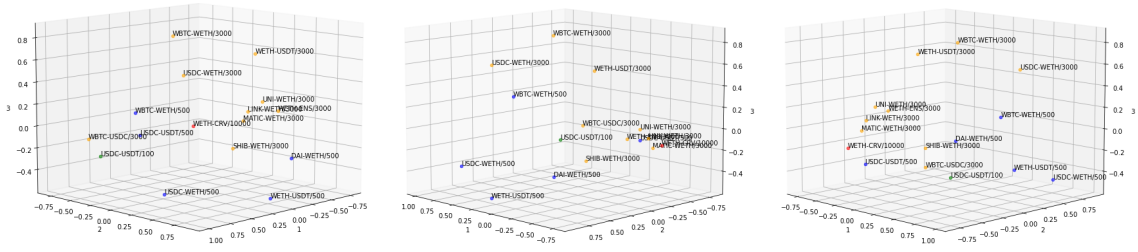


Figure 5.12: Spearman correlation between the computed features for pools, with the addition of feeTier, for our case A.



(a) Linear PCA. (b) rbf kernel PCA. (c) Cosine kernel PCA.

Figure 5.13: Eigenvalue decay for the first 7/13 eigenvalues for different PCA kernels.



(a) Angle = 45 degrees. (b) Angle = 135 degrees. (c) Angle = 315 degrees.

Figure 5.14: Projection on a 3D space of the vectors encoding our pools' features, after PCA with cosine kernel. Views from different angles are reported for a better judgment of the results, and pools are color-coded according to their feeTier (green relates to feeTier 100, blue to 500, orange to 3000, and red to 10000).

5.4.2.2 Discussion

From the projections shown in Fig. 5.14 and initial trials of clustering, it is clear that the division between SS, ECOSYS and EXOTIC pools does not hold when considering the full set of dynamics on pools (while it is indeed suitable in connection to LTs' behaviour specifically). Similarly, we do not witness strong proximity of pools with same feeTier. Liquidity consumption, provision and price evolution are all essential mechanisms to consider for a full description of the Uniswap ecosystem, and our intuition is that certain combinations of tokens and feeTiers are more similar and suitable for trading at different moments in time. LPs are more incentivised to enhance liquidity on pools with strong LTs activity, low volatility of the exchange rate to avoid predictable loss, and possibly high feeTier from which they indeed mainly profit. In parallel, LTs are more interested in pools with low fees but high volatility of the price of tokens in order to extract gains from trading opportunities, and high liquidity to diminish the market impact of their trades. Thus, different adjustments of these mechanisms can result in the proximity or not of our projections of pools.

5.5 Conclusions

Despite blockchains and Decentralised Finance are quite recent concepts, they have quickly established themselves among the great topics of interest of both practitioners and academics. Nevertheless, a real comprehension of the characteristic dynamics within the related protocols is still far away. Our investigations aim at being a stepping stone towards a deeper understanding of the crypto ecosystem, and we achieve this task by empirically studying and characterising the Uniswap v3 DEX. We build a workflow to define the most relevant liquidity pools over time by assessing the inner features of pools along with their interconnectedness, and provide related lists of liquidity pools significant for six different windows in time, i.e. cases $A/B1/B2/C1/C2/C3$, that can be directly used for future research studies. We then focus on LTs and show the existence of seven "species" of traders with interpretable features. These clusters are recovered by assessing the equivalence of LTs structural trading behaviour on a set of highly interconnected pools, via a novel graph embedding approach developed for the purpose. These results suggest a connection between patterns in the features of traders' swap transactions in the broad Uniswap ecosystem, and specific types of pools on which these operations are indeed executed.

Regarding related further directions of research, there are two threads of possible major interest. The first one is a detailed investigation into the behaviour of LPs, along with the corresponding clustering of species, also leveraging on a strongly data-driven approach. Then, the entire data on the Ethereum blockchain could be used to track the flow of funds over multiple DEXs and active protocols. This would enable an extended understanding of LTs, as one would then be able to approximate their profits and losses. Indeed, we showed that the crypto ecosystem is highly interconnected, and we further know that agents easily trade also between different exchanges on the same blockchain.

Chapter 6

Blockchain Activity and Incoming Trading Volume

To conclude this manuscript, we complete one further study that shows the usefulness of blockchain data transparency. We consider two major venues that exchange BTC-ETH on the Ethereum blockchain, namely the Uniswap v3 WBTC-WETH liquidity pools with proportional transaction costs of 5 and 30 basis points, respectively. Our intuition is that such market fragmentation, combined with the extensiveness of blockchain data on both liquidity takers' and providers' actions, can allow the prediction of incoming trading volume. Thus, we introduce a multivariate linear regression model with quadratic terms to predict the logarithm of incoming trading volume on either pool, where the possible prediction horizons are measured in number of blocks mined on the blockchain. We design and test features that relate to the latest activity on the mentioned pools, but we also allow spillover effects from a sample of one-hop liquidity pools in the network of venues that exchange a common asset. Furthermore, we augment the space of possible predictors with statistics computed from the BTC-ETH trading activity on the Binance crypto centralised exchange. Our results indicate that both the latest activity on our two main pools of interest and their neighbours, and the most recent evolution of Binance trading, are necessary information to explain a significant proportion of the variance in the upcoming trading volume. Thus, the drivers of liquidity takers' activity span from the broader decentralised ecosystem to centralised crypto exchanges, highlighting the importance of interconnectedness analyses and breadth of data needed to achieve stronger insights.

6.1 Introduction

As already highlighted in Chapter 5, the programmable nature of the Ethereum blockchain has facilitated the emergence of a novel type of finance, namely DeFi. This part of our study aims at investigating the drivers of incoming trading volume on liquidity pools, with the ultimate goal of forecasting the size of this volume. We focus on the most advanced and active trading venues, and specifically construct our model upon the Uniswap v3 mechanisms, which we introduced in Section 5.2.

The academic literature concerning DeFi proposes several investigations on the problems of optimal swap order execution [35, 37], and optimal liquidity provision to maximise the profitability of LPs' positions [15, 36, 43]. However, to the best of our knowledge, no prior study has focused on the task of prediction of trading volume in liquidity pools. This is what we indeed aim to pursue, with a special focus on leveraging the witnessed market fragmentation (i.e. persistent significant trading activity on multiple pools that exchange same tokens but with different execution fee). Interestingly, only few research [20, 28, 40, 125] considered such problem also in traditional finance, despite volume being a key variable in many financial and economic theories, as well as a practical indicator of liquidity, movements of prices, slippage, and overall market activities.

Main contributions. Our main contributions lie in the development of a framework for short-term prediction of trading volume on liquidity pools, and the identification of forecasting factors for such volume via an optimised multivariate log-linear regression model with quadratic terms. The framework measures time in number of blocks mined on the blockchain, and triggers a prediction event each time a mint operation is executed on a pool of interest. We show that a significant proportion of the variance in incoming trading volume can be explained by accounting for features that are both built from the latest activity on the pair of pools of interest, but also computed from spillover effects from other liquidity pools or Centralised Exchanges (CEXs) such as Binance. For clarity, a schematic diagram of this framework is provided in Fig. 6.1.

Structure of the Chapter. Section 6.2 introduces the data used throughout the study, while Section 6.3 describes the features built to forecast incoming trading volume. In Section 6.4, we optimise a multivariate regression model and discuss the results. Section 6.5 concludes our work and provides future research directions.

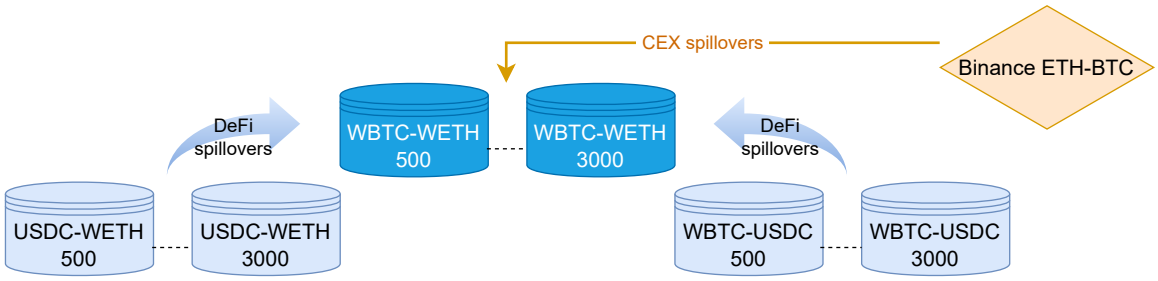


Figure 6.1: Illustrative representation of the spillover effects from other Uniswap v3 liquidity pools, and a CEX such as Binance, over our specific pools of interest (i.e. WBTC-WETH pools).

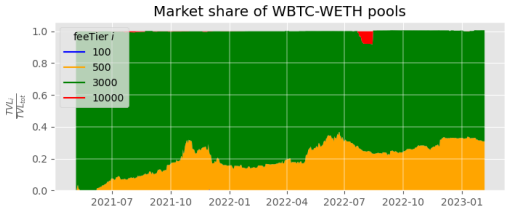
6.2 Data

6.2.1 Uniswap v3 DEX data

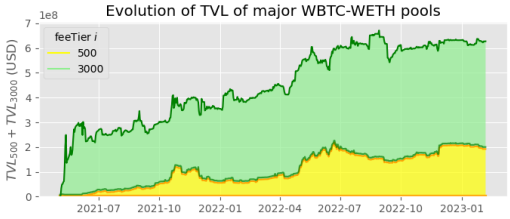
We focus on the trading activity of the most emblematic pair of cryptocurrencies that are both not stablecoins, namely the BTC-ETH pair. Interestingly, there are two liquidity pools on Uniswap v3 with intense operation activity and strong interconnectedness in the broader DEX ecosystem, which both exchange the wrapped versions of these currencies (see Appendix C, from Chapter 5 analyses). These pools are the Uniswap v3 WBTC-WETH liquidity pools with feeTier 500 and 3000. Figure 6.2 proposes evidence for such statement, by showing both the volume locked and traded for all four possible WBTC-WETH pools. Related data are downloaded from the Uniswap protocol subgraph[1], for both liquidity provision events (i.e. LPs mint and burn operations) and liquidity consumption swap operations (i.e. the exchange of one asset for the other one by LTs). Then, we approximate the Total Value Locked (TVL) in a pool at the end of each day by computing the cumulative sum of minting and burning operations, valued in USD, from May 2021 up to that moment (i.e. since the latest version Uniswap v3 was launched in May 2021). Also, we calculate the daily volume exchanged by summing the USD value of LTs' swap operations at each date.

By comparing transactions data with Etherscan[2] for our two main pools of interest, we then adjust times to the standard UTC timezone and restrict ourselves to the period between April and September 2022 to avoid misalignment errors due to daylight savings. Indeed, we need to require a standard timezone to link block mining times to the latest realised activity on the Binance CEX. The restricted six-month period includes a series of interesting events, such as the Terra-Luna collapse in May 2022

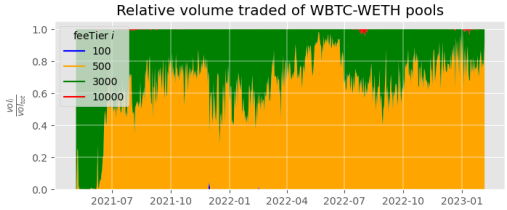
[1] `https://github.com/Uniswap/v3-subgraph/blob/main/schema.graphql#L1`
[2] `https://etherscan.io/`



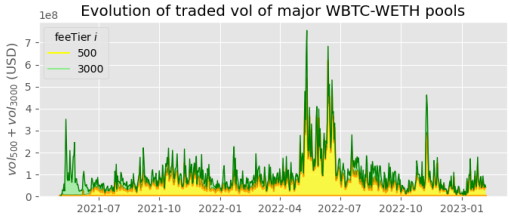
(a) Relative TVL among the four pools considered.



(b) Stacked plot of TVL for the two major pools.



(c) Relative volume among the four pools considered.



(d) Stacked plot of volume for the two major pools.

Figure 6.2: Fragmentation of liquidity locked in the pools and correspondent volume traded. The pools considered exchange the same pair of assets WBTC-WETH but adopting different feeTier. It is clear that the two pools with feeTiers 500 and 3000 share the majority of market activity, while the others are not significant.

and the Merge in September 2022. Next, we threshold operations to a minimum value of 10 USD and group LP mint events that happen on the same block. Instances of the latter are rare (i.e. $\sim 2\%$ of our entries) and are aggregated for later convenience into a single operation of value equal to the sum of money provided, and width (i.e. range of prices among which such liquidity is confined) equal of the average of the related entries.

Finally, we download and pre-process data also for LTs operations on the Uniswap v3 WBTC-USDC and USDC-WETH pools, for both feeTier 500 and 3000. These pools are among the most active one-hop neighbours of WBTC-WETH in the network of pools that exchange one common asset, and allow us to consider network spillover effects in the DeFi ecosystem.

6.2.2 Binance CEX data

We augment the DEX data set with associated data of trades on Binance, made available by the data provider Tardis.dev. These trades refer to the ETH-BTC pair and we subset them to the same period of interest chosen for the DEX data set. The data is at minutely frequency, in UTC time, and notional values are quoted in BTC.

To associate Binance data to block numbers, we first uniformly spread the volume and trade counts for Binance operations recorded at the end of each minute over the previous 60 seconds. At each second, we also include the latest mid-price registered by the exchange, since divergences with Uniswap pool prices can trigger volume from arbitrageurs. Next, we consider our DEX data set and extract all transaction block numbers with the related timestamps. Blocks are mined every $10 - 19$ seconds on Ethereum before its upgrade to the Proof of Stake consensus mechanism, and every 12 seconds afterwards. Thus, we approximate the link between Binance data and block numbers by mapping all transactions on the former venue to the closer block number in the future. Note that this does not introduce a forward-looking bias, since we define the clock of our prediction model via mint operations and we consider a strict lower bound whenever we compute features after the tick of the clock.

6.3 Feature engineering

6.3.1 Framework and target variables

From now on, we refer to our two WBTC-WETH main pools of interest as pools 500 and 3000, for the sake of conciseness. Any mint operation taking place on either pool is an event of relevance to us, and we aim at predicting the cumulative incoming volume in USD on both pools after each such new provision of liquidity (i.e. ticking of our trading clock). We test forecasting horizons every ten blocks up to the next mint operation on either pool. Thus, our target variables are the cumulative incoming trading volumes that occur either on the *same* pool of the mint operation, or on the *other* pool. To build features to subsequently use in our forecasting model, we consider blockchain data on DEX activity, as delimited by the three previous mint operations on pool 500 and the previous three mint operations on pool 3000, up to the block of the current mint operation of reference. For clarity, we provide an illustrative example of this framework in Fig. 6.3.

6.3.2 Features

We construct a set of 59 possible predictive features of incoming trading volume from both our DEX and CEX data sets. Importantly, data from the pool where the mint operation of reference occurred are referred to as *same* venue variables, while alternatively they are referred to as features from the *other* venue. We also denote with N_1N_2 the time ranges in blocks from the N_1 past mint operation to the previous

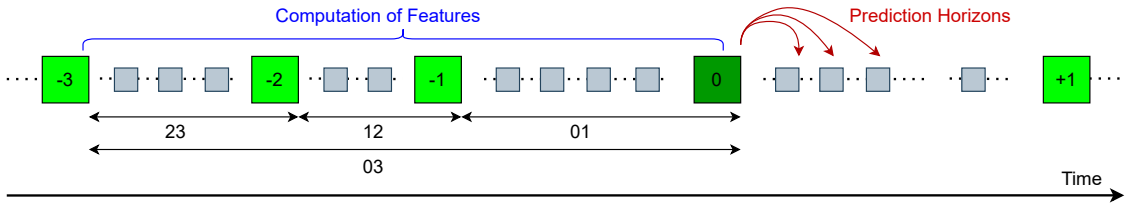


Figure 6.3: Illustrative diagram of our framework for the case of a single pool of reference, for simplicity. We depict the blockchain and highlight blocks that record mint operations occurring on one *same* pool in green. Features are computed from the current block of interest (i.e. the dark green block) up to three previous mint operations. We predict the cumulative incoming volume on horizons every ten blocks, up to the next mint operation.

past N_2 one. As an example, the range $\widetilde{03}$ shown in Fig. 6.3 refers to the blocks between the current mint operation up to the third latest mint operation on the pool. We now fully list our features, while further descriptions and related short labels are provided in Table 6.1 for ease of reference.

- Direct pool features (43 features): ♠ time in blocks from the previous three mint operations on the *same* pool and on the *other* pool (i.e. lengths of the $\widetilde{01}$, $\widetilde{02}$, $\widetilde{03}$ intervals on both pools, where the 0 point is always the mint operation of reference regardless of the pool). ♠ Size and ♠ width (i.e. due to the concentrated liquidity mechanism of Uniswap v3) of both the current mint operation of reference, and of the previous three mint operations on both pools. ♠ Volatility of the pool price on the *same* pool over the $\widetilde{01}$, $\widetilde{02}$, $\widetilde{03}$ intervals. ♠ Rate, as USD traded per block by LTs, on the $\widetilde{01}$, $\widetilde{12}$, $\widetilde{23}$ intervals on both pools. ♠ Rate, as count of trades per block of LTs, on the $\widetilde{01}$, $\widetilde{12}$, $\widetilde{23}$ intervals on both pools. ♠ Average trade size in USD on the $\widetilde{01}$, $\widetilde{12}$, $\widetilde{23}$ intervals on both pools. ♠ Latest value of the total value locked (TVL) in the pool 3000 minus the TVL in the pool 500, and ♠ ratio of the two (TVL is approximated by computing the cumulative sum in USD of the mint and burn operations for both pools from May 2021, which is when Uniswap v3 was deployed).

- DeFi network spillover effects (8 features): ♠ traded volume in USD on the Uniswap v3 WBTC-USDC and USDC-WETH pools, for both feeTier 500 and 3000. These volumes refer to the $\widetilde{03}$ block time range as defined by the mint operations on the *same* pool. For these pools and time range, we also compute the ♠ surplus buying pressure on either WETH or WBTC by computing the

difference in the volume of buying versus selling orders, often referred to as *trade flow imbalance*.

- CEX spillover effects (6 features): ♠ traded volume in BTC on Binance, and related ♠ count of trades, for the $\widetilde{01}$, $\widetilde{12}$, and $\widetilde{23}$ block time ranges with respect to the latest mint operation executed on the *same* pool.

- Price divergences (2 features): ♠ difference between the latest recorded price on the *same* pool versus the latest one on the *other* pool, and ♠ difference between the latest recorded price on the *same* pool versus the latest mid price on Binance.

Table 6.1: Features for the determination of LTs incoming trading volume. We provide these labels for ease of reference of the above-mentioned possible predictors

Label	**Feature**
bl_{same}, bl_{other}	Distance in blocks of the previous $l \in \{1, 2, 3\}$ mint operations on the *same* pool as the one where the mint of reference happened, or on the *other* pool.
sl_{same}, sl_{other}	Size in USD of the previous $l \in \{1, 2, 3\}$ mint operations on the *same* pool as the one where the mint of reference happened, or on the *other* pool.
wl_{same}, wl_{other}	Width in number of ticks of the previous $l \in \{1, 2, 3\}$ mint operations on the *same* pool as the one where the mint of reference happened, or on the *other* pool. This feature might become significant due to the concentrated liquidity mechanism and possible just-in-time liquidity provision.
s0, w0	Size in USD, and width in number of ticks, of the current mint operation that we take as reference.
vol0l	Volatility as the standard deviation of the pool price on the *same* pool as the mint of reference, during the expanding intervals of time $\widetilde{0l}$ with $l \in \{1, 2, 3\}$ that refer to the previous mint operations on the *same* pool.
rate-USD-i_{same}, rate-USD-i_{other}	Rate of traded volume in USD on our WBTC-WETH pools of principal interest, for the intervals $i \in \{\widetilde{01}, \widetilde{12}, \widetilde{23}\}$ block ranges with respect to either the latest mint operations and swaps executed on the *same* pool, or on the *other* pool.
rate-count-i_{same}, rate-count-i_{other}	Rate of count of trades on our WBTC-WETH pools of principal interest, for the intervals $i \in \{\widetilde{01}, \widetilde{12}, \widetilde{23}\}$ block ranges with respect to either the latest mint operations and swaps executed on the *same* pool, or on the *other* pool.
Continued on next page	

Table 6.1 – continued from previous page

Label	Feature
avg-USD-i_{same}, avg-USD-i_{other}	Average traded volume in USD on our WBTC-WETH pools of principal interest, for the intervals $i \in \{\widetilde{01}, \widetilde{12}, \widetilde{23}\}$ block ranges with respect to either the latest mint operations and swaps executed on the *same* pool, or on the *other* pool.
TVL3000-500, TVL3000/500	Latest value of the TVL in the pool 3000 minus the TVL in the pool 500, and ratio of the two.
eth500-USD-03, eth3000-USD-03	Rate of traded volume in USD on the one-hop USDC-WETH pools, over the $\widetilde{03}$ time interval with respect to the previous mint operations on the *same* pool as our mint of reference.
btc500-USD-03, btc3000-USD-03	Rate of traded volume in USD on the one-hop WBTC-USDC pools, over the $\widetilde{03}$ time interval with respect to the previous mint operations on the *same* pool as our mint of reference.
eth500-press-03, eth3000-press-03	Rate of surplus WETH buying volume in USD on the one-hop USDC-WETH pools, over the $\widetilde{03}$ time interval with respect to the previous mint operations on the *same* pool as our mint of reference.
btc500-press-03, btc3000-press-03	Rate of surplus WBTC buying volume in USD on the one-hop WBTC-USDC pools, over the $\widetilde{03}$ time interval with respect to the previous mint operations on the *same* pool as our mint of reference.
binance-btc-i	Rate of traded volume in BTC on Binance, for the intervals $i \in \{\widetilde{01}, \widetilde{12}, \widetilde{23}\}$ block time ranges with respect to the latest mint operations executed on the *same* pool as our mint of reference.
binance-count-i	Rate of count of trades on Binance, for the intervals $i \in \{\widetilde{01}, \widetilde{12}, \widetilde{23}\}$ block time ranges with respect to the latest mint operations executed on the *same* pool as our mint of reference.
$\Delta(Z_{same}, Z_{other})$, $\Delta(Z_{same}, Z_{Binance})$	Difference between the latest recorded price on the *same* pool versus the latest one on the *other* pool, and difference of the former again versus the latest mid price on Binance.

6.4 Prediction

6.4.1 Multivariate linear regression

We propose to investigate the predictive power of the above-mentioned features by optimising a multivariate linear regression model. To this end, we provide a brief review of such methodology, based on [71]. In general, we start with p distinct predictors and assume that there is approximately a linear relationship between the predictors and the response Y. Then, the multivariate linear regression model takes the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon, \tag{6.1}$$

where X_j represents the jth predictor variable, and the coefficient β_j quantifies its association with the response. The term ϵ is a mean-zero random error that is assumed to be independent of predictors X. If we have a number n of observations for our predictors and corresponding response values, we compute estimates $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ of the coefficients of the model via the least-squares approach. The latter minimises the sum of squared residuals $\text{RSS} = \sum(y_i - \hat{y_i})^2$, where $\hat{y}$ are predictions of the response variable computed as

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p. \tag{6.2}$$

However, we also need to test for the hypothesis that there is no relationship between the predictors and the response. This is done by computing the F-statistic

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)}, \tag{6.3}$$

where $\text{TSS} = \sum(y_i - \bar{y})^2$ is the total sum of squares. If the F-statistic is larger than 1 but its p-value is small, then we can conclude that at least one β_j is indeed non-zero.

Finally, we assess the lack of fit of the model via the root mean squared error $\text{RMSE} = \sqrt{\frac{\text{RSS}}{n}}$, and similarly gauge the extent to which the model fits the data by computing its R^2 coefficient

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}}, \tag{6.4}$$

which measures the fraction of variance in the response variable that is indeed explained by the model. Since R^2 always increases when more variables are added to the model, it is often useful to compare such models by considering the adjusted R^2

$$R^2_{adj} = 1 - \Big(\frac{n-1}{n-p}\Big)(1 - R^2), \tag{6.5}$$

which takes into account the number of predictors.

6.4.2 Collinearity analysis

We built a set of 59 features to possibly include in our model. However, two or more of these variables could be closely related to one another, i.e. be collinear. Collinearity reduces the accuracy of the estimates of the regression coefficients, since it lowers our ability to separate out the individual effects of the concerned variables in the model. Similarly, collinearity also causes inaccuracies in the t-statistic of the related variables, where this measure assesses the significance and level of contribution of each feature to the regression.

To this end, we compute the correlation matrix of our predictors in Fig. 6.4 to detect obvious cases of collinearity. The few critical instances of high correlation are dealt with by first dropping the b2_{same}, b3_{same}, b2_{other}, b3_{other}, vol01, vol02, TVL3000-500, and binance-count-$(01, 12, 23)$ variables. Then, we aggregate problematic rates to the longer $\widetilde{03}$ intervals, i.e. we compute rate-count-03_{same}, rate-count-03_{other} and binance-btc-03, and drop the features related to the respective sub-ranges. Finally, we sum eth500-USD-03 and eth3000-USD-03 into a new feature ethTot-USD-03, and similarly we calculate btcTot-USD-03. These modifications result in a final set of 41 features to consider in our model. For completeness, we further assess multicollinearity among three or more of our latest set of variables by computing the Variance Inflation Factor VIF

$$\text{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}, \tag{6.6}$$

where $R^2_{X_j|X_{-j}}$ is the coefficient of determination of the regression of X_j onto all other predictors. If VIF > 10, then there is still a problematic amount of collinearity. However, the majority of our factors have a VIF value close to 1, with only a few cases that reach VIF $= 7.5$ but that we allow.

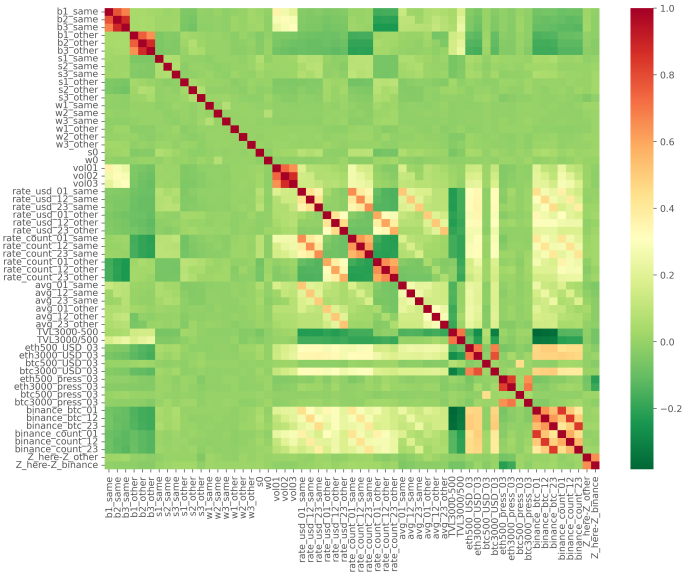


Figure 6.4: Correlation matrix of our features.

6.4.3 Model selection

We now move on to an analysis of the overall predictive power of our variables, and will then proceed to features selection in order to achieve optimised models for forecasting

incoming trading volume on our liquidity pools of interest. In particular, we build different models according to whether the mint operation of reference is executed on pool 500 or pool 3000. Then, our target variables can either relate to the cumulative incoming volume on the *same* pool as the one where the mint operation of reference is executed, on the *other* pool, or in aggregation on *both* pools. In addition to this, we also test for different horizons that widen ten blocks each time, and that can theoretically lengthen up to the maximum time between which two consecutive mint operations on either pool are recorded on the blockchain. In practice, it is often the case that these mint operations occur on a timescale of minutes. And indeed, this can be observed from the decay of the distribution of number of data points available per horizon, which is shown on the secondary axis of both plots of Fig. 6.5. The stacked histograms are restricted to a maximum horizon of 300 blocks, and they are coloured differently according to whether the mint operation of reference occurred on pool 500 or pool 3000.

For each horizon up to 300 blocks, we split the related data points into train and test sets with a $80:20$ ratio. We compute the standard deviation of the features in the train set, and we divide both the train and test sets by these values. We do not fully standardise our data sets, i.e. we do not first subtract the mean of features, since we wish to maintain the interpretability of the sign of coefficients arising from the regression model. Finally, we perform multivariate linear regressions that consider all our 41 features for the models, as described above. Figure 6.5 (left) shows the R^2s achieved on the train set when predicting volumes after a mint operation occurred on pool 3000. Figure 6.5 (right) then applies each related model on the test set and

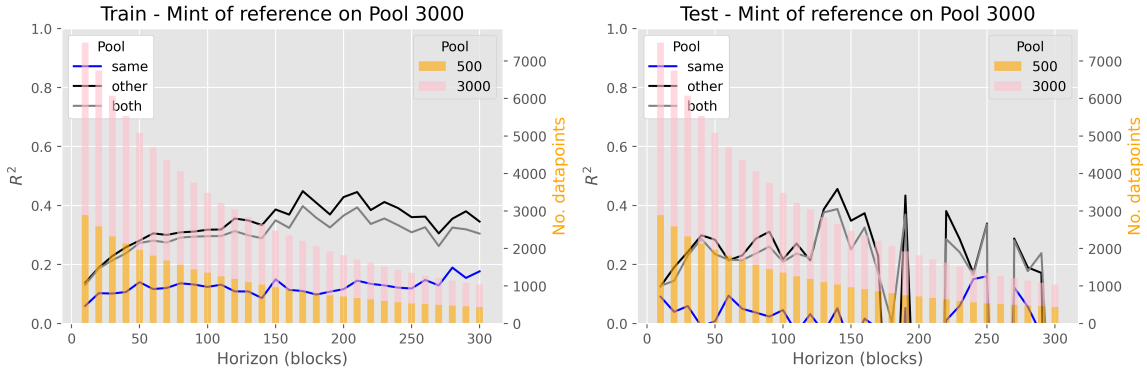


Figure 6.5: R^2 on the train (left) and test (right) sets. We predict incoming volume after a mint operation occurs on pool 3000, by allowing all the 41 features developed. For completeness, we also plot on both secondary axes the number of data points available for the regressions.

shows the resultant R^2s. Our features explain a significant percentage of the variance of responses in the train but also test sets, if we specifically consider the prediction of incoming trading volume on the *other* pool, i.e. pool 500. We also compute the F-statistic defined in Eq. (6.3) for each related horizon up to 150 blocks, and find indeed that we can always reject the null hypothesis that there is no relationship between our predictors and the responses. The model also exhibits predictability for the aggregated volume (*both*) with some significance, but this is not surprising since the volume traded on pool 500 is strongly higher due to the lower trading costs for LTs, and thus dominates the prediction task. We finally perform the above analyses for models focusing on a mint operation of reference occurring on pool 500. The results are shown in Fig. 6.6 but not further investigated, since the related R^2s on the test sets are generally negligible.

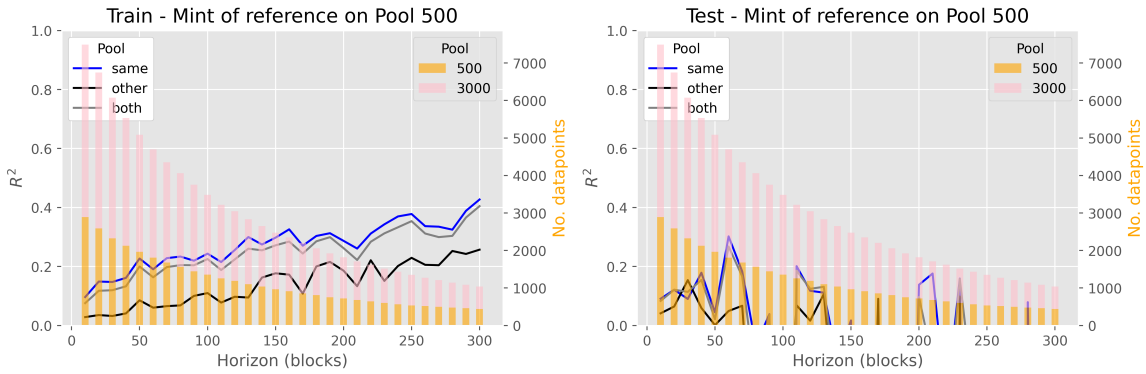


Figure 6.6: R^2 on the train (left) and test (right) sets. We predict incoming volume after a mint operation occurs on pool 500, by allowing all the 41 features developed. For completeness, we also plot on both secondary axes the number of data points available for the regressions.

We now proceed to a detailed investigation of our forecasting ability regarding the incoming trading volume on pool 500, after a mint operation occurred on pool 3000. This particular combination of actions is of main importance, since the majority of volume tends to be traded on the pool with lower trading fees for LTs, i.e. on our pool 500, while LPs are indeed more keen to stake their assets on pool 3000 and profit from such higher fees paid by LTs. In multivariate linear regression, it is often the case that the response variable is only associated with a subset of the predictors. Thus, we now pursue step-wise feature selection. We choose a target horizon of 120 blocks by examining the R^2 values on the test set in our chosen framework. This can be thought of as a time of ~ 24 minutes, if we use the common approximation of 12 seconds per new block mined on the Ethereum blockchain. We then start with a model with no

features and iteratively perform forward and backward steps, until our forecasting model does not incur changes and is indeed considered as optimised. Our forward steps consider all variables not included in the model yet, and add the one feature that would result in the highest increase in adjusted R^2 if it also has p-value < 0.05. The backward steps simply drop all features with p-value ≥ 0.05, where this check is necessary since the significance of predictors can change after the addition of new ones in the model.

The results of our step-wise feature selection indicate that ten features define our optimal model at the 120-blocks horizon. With reference to the labels provided in Table 6.1, these features are: rate-usd-01$_{other}$, rate-usd-12$_{other}$, ethTot-USD-03, TVL3000/500, b1$_{other}$, avg-23$_{other}$, w1$_{same}$, binance-btc-03, rate-usd-12$_{same}$, and btc3000-press-03. The model achieves high R^2 values of 0.35 and 0.22, on the train and test sets, respectively. However, we further check the distribution of error terms via the residual plot in Fig. 6.7 (left) and find a clear indication of heteroscedasticity, i.e. non-constant variances in the errors that violate our related assumption in the regression. Thus, we experiment by taking the logarithm in base 10 of our responses and fitting otherwise the same model. The resultant residuals are compared against our fitted values for the train and test sets, and they still exhibit a non-linear pattern that broadly resembles a quadratic curve. To this end, we experiment with feature transformations, and also test an augmented model in which we include the second power of the three types of past volumes appearing in our relevant features (i.e. of the rate-usd-01$_{other}$, ethTot-USD-03 and binance-btc-03 features, in order to enhance the well-known volume autocorrelation effect ubiquitous in financial applications that is also reported for our pools in Fig. 6.8). The related residual plot is shown in Fig.

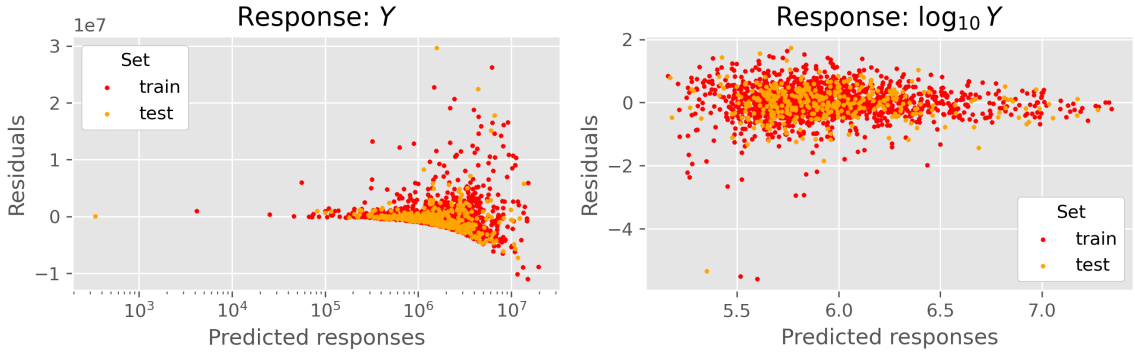


Figure 6.7: (Left) Residual plots for our plain multivariate linear regression. (Right) Residual plot after we allow for the second power of past volumes and take the logarithm of the response.

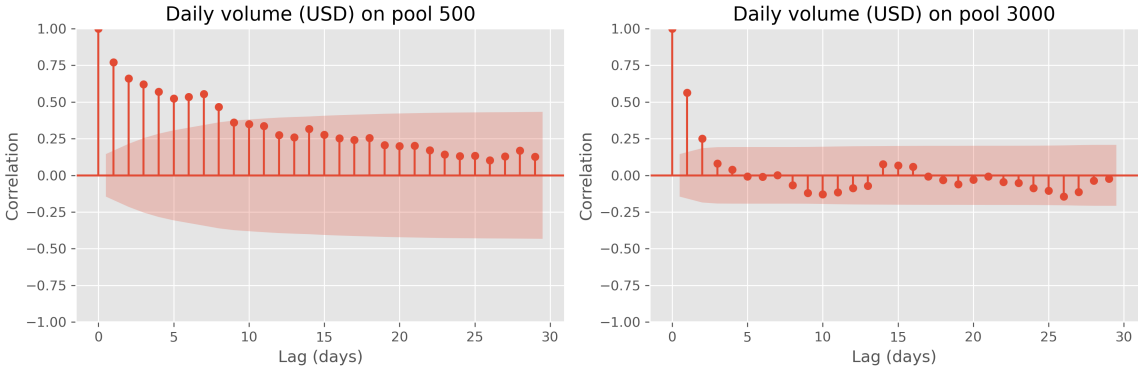


Figure 6.8: Coefficients of correlation between a time series and its lagged values, for the cases of daily traded volume on pool 500 (left) and on pool 3000 (right). Autocorrelation is significant up to a week on pool 500, while it decays more strongly on pool 3000. The shaded areas indicate the 95% confidence interval.

6.7 (right). We observe that the previous pattern is significantly less prominent after such a transformation, and indeed our exhibit approaches what is expected from a tested linear relationship. For completeness, we also perform the White's test for heteroscedasticity on the residuals from the train data. We observe a variation from a (statistics, p-value) of $(318,\ 5 \cdot 10^{-35})$ to $(64,\ 1 \cdot 10^{-3})$ before and after our transformation of the response and addition of quadratic terms, implying some progress towards homoscedasticity.

Before concluding that the above-mentioned sets of features are indeed determinants of incoming trading volume, we assess the strength and stability of this result. Since we are forecasting volume over time, we also expect our features to be significant also at horizons similar to the target one, i.e. our 120-blocks target. Thus, we consider the above ten linear features and three quadratic ones. We then build related log-linear regressions at horizons from 10 to 150 blocks, and retain the set of features that are found to be significant over at least 50% of such horizons. Interestingly, seven features (plus the constant of the model) are the parameters of overall significance for the span of horizons of interest. These are: $\mathbf{rate\text{-}usd\text{-}01_{other}}$**, ethTot-USD-03, TVL3000/500, binance-btc-03, and** $\mathbf{(rate\text{-}usd\text{-}01_{other})^2}$**,** $\mathbf{(ethTot\text{-}USD\text{-}03)^2}$**,** $\mathbf{(binance\text{-}btc\text{-}03)^2}$. We perform final log-linear regressions for the above horizons with only such features and plot the resultant coefficients in Fig. 6.9 (upper left). If a feature is not significant at a certain horizon, its line breaks at that point. The related t-statistics (that measure the number of standard deviations that the estimated coefficients are away from zero) are plotted in absolute value in Fig. 6.9 (upper right). Our coefficients have consistent and similar strengths overall, while it is the constant

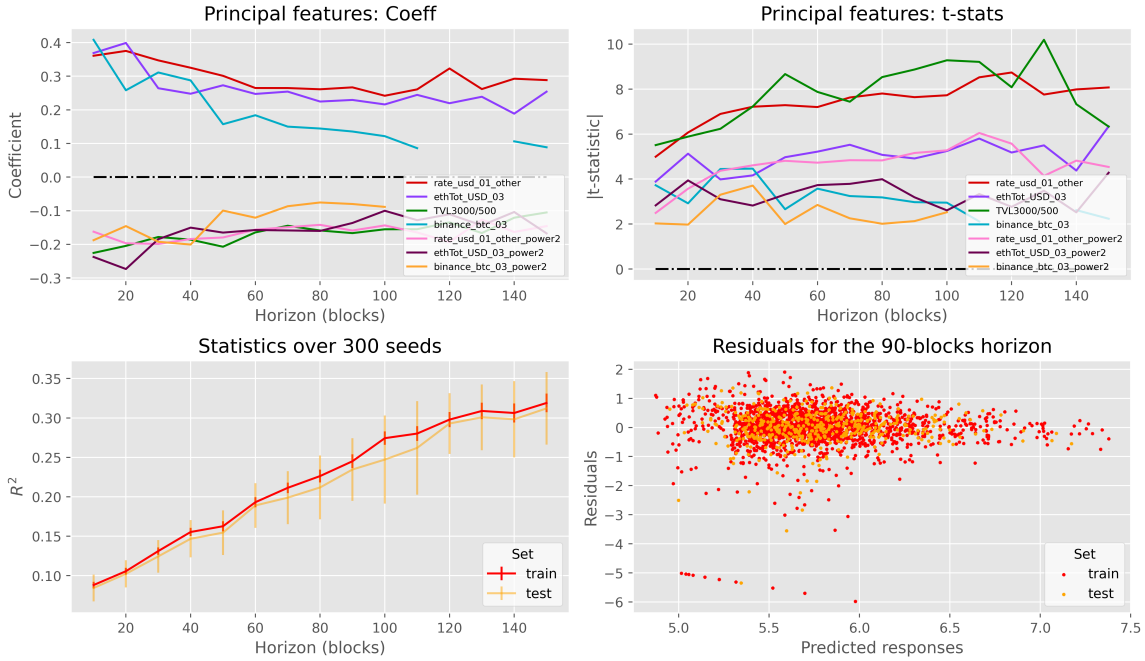


Figure 6.9: Results related to a log-linear multivariate regression that includes quadratic effects. (Upper left) Coefficients of features that are consistently significant for the prediction of incoming volume over horizons from 10 to 150 blocks. (Upper right) Magnitude of the t-statistics of such coefficients. (Lower left) Average train and test R^2s, with standard deviations as error bars, when we run the model over 300 different partitions of the data into train-test split at each horizon. (Lower left) Residual plot of errors at the 90-blocks horizon.

that defines an increasing trend of logarithmic shape in line with the task of predicting cumulative trading volume at lengthening horizons. Furthermore, we notice that the volume traded on Binance has a shorter-term effect and is not significant after the horizon of 110-blocks.

We then repeat these regressions at each horizon, but for 300 different partitions of our data into 80 : 20 train and test sets. The averages of train and test R^2s are computed at each horizon and plotted in Fig. 6.9 (lower left), where the error bars are the related ± 1 standard deviations. Such R^2s show that we are indeed able to capture a significant amount of the variance in the data with the seven features retrieved, and that the results are stable and consistent across the splits. We observe a knee in the curve at the 90-blocks horizon, and consequently plot the related residuals distribution against the predicted responses for one of the runs in Fig. 6.9 (lower right). Since no heavy trend is visible, we accept the related model as our definitive one.

As a final remark, we stress that the test R^2s can be witnessed to be higher than the train ones for single runs of models. This occurs due to the variability inherent in

our data, and the consequent presence of specific unusual events in either the train or test subset. One clear example regards the data points related to days during the Terra-Luna collapse in May 2022, which is a shock that indeed affected the broader DeFi ecosystem (see the high levels of trading volume on our pools of main interest in Fig. 6.10).

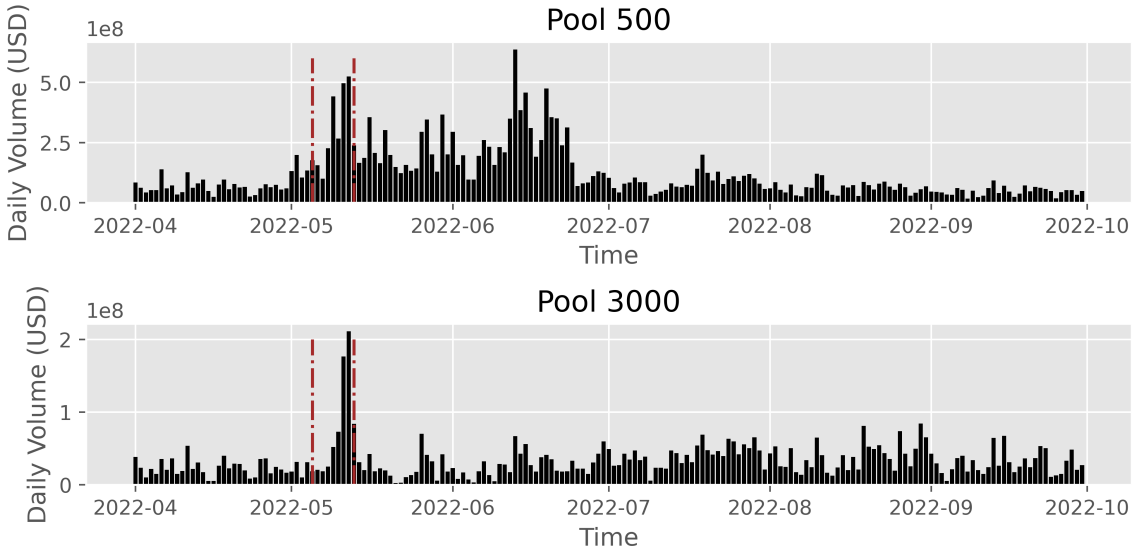


Figure 6.10: Daily volume traded in USD during our period of interest on both pools. The brown vertical lines indicate the begin (May 5th, 2022) and end (May 13th, 2022) of the Terra-Luna collapse event.

6.4.4 Discussion of results

Our final model predicts the incoming trading volume y on pool 500 at the 90-blocks horizon, after a mint operation occurs on pool 3000. Its forecast is given by

$$\widehat{\log_{10} y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \hat{\beta}_5 (x_1^2) + \hat{\beta}_6 (x_2^2) + \hat{\beta}_7 (x_3^2), \tag{6.7}$$

where the predictors and related coefficients follow from Sec. 6.4.3 and are summarised in Table 6.2.

The model achieves a train R^2 of 0.24 for a randomly chosen reference seed in the train-test split, and an F-statistic of 82 with p-value $8 \cdot 10^{-104}$. The intercept $\hat{\beta}_0 \sim 6$ suggests a base value of USD $1,000,000$ for the expected volume in the ~ 24 minutes after a mint operation is registered on pool 3000. This measure is then affected by the values taken by our predictors. In particular, coefficients $\{\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3\} \in \mathbb{R}^+$ imply that volumes $\{\text{rate-usd-01}_{other}$, ethTot-USD-03, binance-btc-03$\}$ have a positive

Component of the regression	**Description**
Ticking of the trading clock	Mint operation happening on pool 3000.
Response variable	Logarithm (with base 10) of the cumulative trading volume y in USD on pool 500.
Horizon	90 blocks.
Predictor variables	$x_1 :=$ rate-usd-01_{other}, $x_2 :=$ ethTot-USD-03, $x_3 :=$ binance-btc-03, and $x_4 :=$ TVL3000/500. Each variable is normalised by an associated measure of standard deviation computed from the train data.
Coefficients	$\hat{\beta}_0 = 6.02$, $\hat{\beta}_1 = 0.26$, $\hat{\beta}_2 = 0.22$, $\hat{\beta}_3 = 0.13$, $\hat{\beta}_4 = -0.17$, $\hat{\beta}_5 = -0.15$, $\hat{\beta}_6 = -0.12$, and $\hat{\beta}_7 = -0.09$.

Table 6.2: Model specification for the prediction of incoming trading volume, following Eq. (6.7)

relationship with the predicted value of incoming trading volume when in their linear form. This is partly expected, since rate-usd-01_{other} refers to the latest volume per block registered on pool 500, and thus provides evidence for the volume autocorrelation effect ubiquitous in financial applications. However, it is interesting to note how other types of volumes are also found to be strong determinants of incoming volume traded by LTs, i.e. our ethTot-USD-03 and binance-btc-03 variables. This means that spillover effects from both the volume traded on the neighbour Uniswap v3 liquidity pools that exchange the common WETH asset against USDC, and the trading activity on the BTC-ETH pair on Binance (the major crypto CEX), are also essential mechanisms to consider to better forecast the incoming volume of interest.

Next, we investigate the coefficients $\{\hat{\beta}_4, \hat{\beta}_5, \hat{\beta}_6, \hat{\beta}_7\} \in \mathbb{R}^-$ and discern two related aspects to discuss. The first one regards $\hat{\beta}_4$ and its associated variable TVL3000/500, meaning that there is a negative relationship between incoming volume on pool 500 and the relative proportion of liquidity locked in our two pools of main interest. Indeed, a higher ratio corresponds to an increase in the liquidity available on pool 3000 with respect to pool 500, and a possible associated more favourable market impact of trades. It is thus reasonable to witness a consequent negative effect on the incoming volume on pool 500 and a possible increase in LTs activity on pool 3000 instead, despite the heftier trading costs characteristic of the latter venue. The seconds aspect relates to coefficients $\{\hat{\beta}_5, \hat{\beta}_6, \hat{\beta}_7\}$, since they imply a negative impact of the square of historical trading volumes {rate-usd-01_{other}, ethTot-USD-03, binance-btc-03} on the incoming volume on pool 500. We interpret this result as a regularisation effect, which penalises exceptionally large latest volumes, and limits the size of our forecasts that would

otherwise arise from the positive influence of the associated linear terms. This is a reasonable result and, if we indeed consider the example of arbitrage trades, then we expect such opportunities to be quickly consumed and the related outlier volumes not to sustain but promptly decay.

For completeness, we now briefly linger on Figs. 6.9 (upper left) and (upper right). These plots confirm the stability of signs and strengths of our predictors over horizons of increasing length. The linear rate-usd-01_{other} and TVL3000/500 act as slightly dominant predictors from the absolute value of their t-statistics, while the traded volume on Binance is a significant feature in our model only at the shorter-medium horizons. Overall, we can conclude that our identified drivers of incoming volume can be indeed deployed for prediction *up to* our 90-blocks horizon of specific interest, with confidence in the strength of their associated coefficients.

Finally, we test the predicting accuracy of our model in Eq. (6.7) against the baseline

$$\hat{y}_{baseline} = x_1 \cdot h, \tag{6.8}$$

where $h = 90$ is the horizon of interest measured in number of blocks. In this case, we are considering the latest volume per block witnessed on pool 500 as plain estimate for the incoming rate of volume on the same pool. We compute the RMSE that both our model and the baseline achieve for 300 different train-test split of our data. On average, the RMSE of our model is a fraction 0.779 and 0.784 of the RMSE from the baseline predictions on the train and test sets, respectively. Thus, our model is a tangible improvement with respect to the baseline.

6.5 Conclusions

The present study investigates the extent to which we can forecast incoming trading volume on Uniswap v3 liquidity pools. We build and compute features from the latest behaviour of both liquidity takers and liquidity providers on both pools of interest. We then augment the space of possible predictors with statistics on the trading activity in neighbouring pools and on the Binance crypto CEX for the same BTC-ETH pair. We find that we are able to explain a significant proportion of the variance in the incoming trading volume if we indeed consider features that extend from the broader decentralised ecosystem to centralised crypto exchanges.

To the best of our knowledge, this work is the first one to build and test a model for the prediction of incoming trading volume on Uniswap v3 liquidity pools. Our framework aims at becoming a baseline in the area, and opens future avenues of

improvements and investigation. Allowing further non-linearities in the model is the first step towards increasing prediction accuracy, while a comparison of performance of the same predictors over different pairs of assets would also be an interesting examination of generalisation. Finally, we suggest to include multi-hop spillover effects from the network of liquidity pools that exchange a common asset, especially since the routing of orders for optimal execution is accentuated on blockchains.

Chapter 7

Conclusions

This thesis expands our understanding of both traditional and decentralised market ecosystems by exploring unconventional data sets via advanced data science techniques. Indeed, the current abundance of data from diverse sources poses new opportunities for broader financial investigations, which divert from the traditional academic focus on time series analysis of asset prices.

To conclude and summarise the work that was completed, we propose in Fig. 7.1 a schematic representation of the topics addressed. However, there are multiple possible extensions directly suggested by our work, which could be pursued within future research efforts. We mentioned many of them within their related chapters of this thesis, but we also desire to comment here on how our proposed methodologies and analyses within traditional (or decentralised) finance could be symmetrically tuned and tested against the decentralised (or traditional) financial ecosystem.

Understanding instances of crowding, due to the overlap of positions from different types of investors, is a topic that truly necessitates to be studied within the crypto universe. This would shed better light on both the health of such ecosystem and the behaviour of speculators, and on the creation of related asset bubbles. Similarly, a better understanding of the evolution of narratives related to decentralised finance could help in identifying the more solid drivers of such assets' behaviour, and separate the fraction of such assets' volatility fomented by sentimental retail traders. We also believe that it could be interesting to directly apply our regime detection studies to the crypto ecosystem, but are aware that results could be less informative due to decentralised finance being a relatively recent innovation in time.

With respect to our specific decentralised finance analyses, our methodology to identify species of traders could be directly applied to the traditional finance context. However, the level of details and transparency required in the data makes it harder to plan such investigations on centralised exchanges due to proprietary reasons. Finally,

we believe that the best use of our trading volume prediction framework is indeed within Uniswap-like exchanges, due to their inherent market fragmentation and related trading mechanisms. However, the framework could surely be also tested against a set of traditional centralised exchanges.

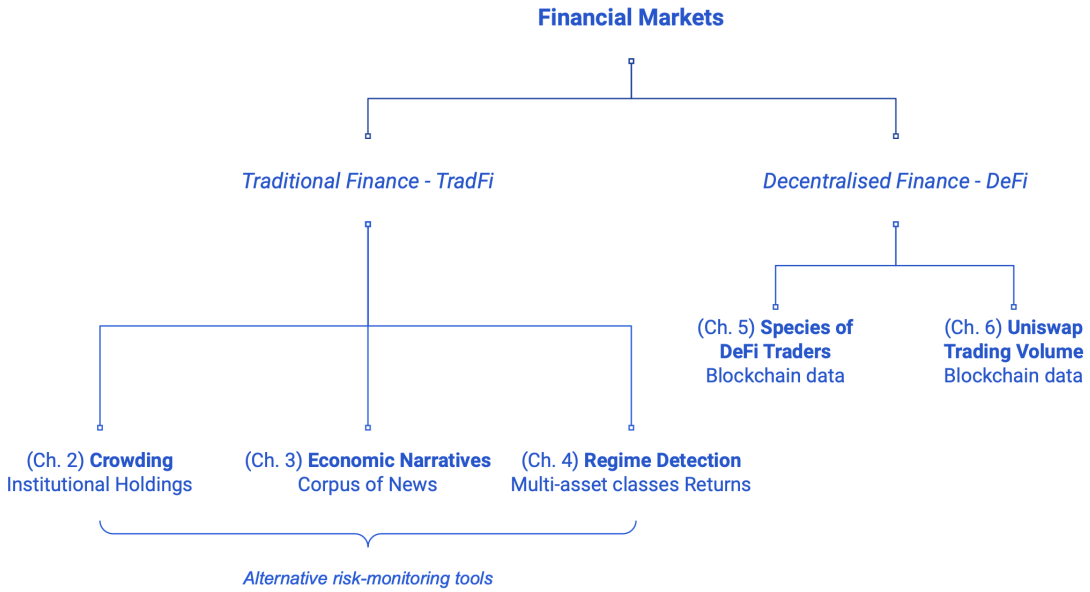


Figure 7.1: Research questions addressed within this thesis and related data.

Appendix A

Appendix of Chapter 2

A.1 Cumulative PnL for the vanilla strategy

We compute time series of the cumulative PnL gained while trading the plain imbalance signal, i.e. buying (selling) stocks if the related imbalance is positive (negative). Figures A.1, A.2 and A.3 show the results for minimum number of active funds $N \in \{50, 150, 500\}$ required on each security. In each case, we plot the performance for quantile ranks qr_i with $i \in \{1, 2, 3, 4, 5\}$ and horizons of $m \in \{5, 10, 21, 42, 63\}$ trading days. We also divide between trading I^{vol} versus I^{tr}. The results agree that a general opportunity for profit arises if the investor is willing to trade *contrary* to imbalances from 13F filings.

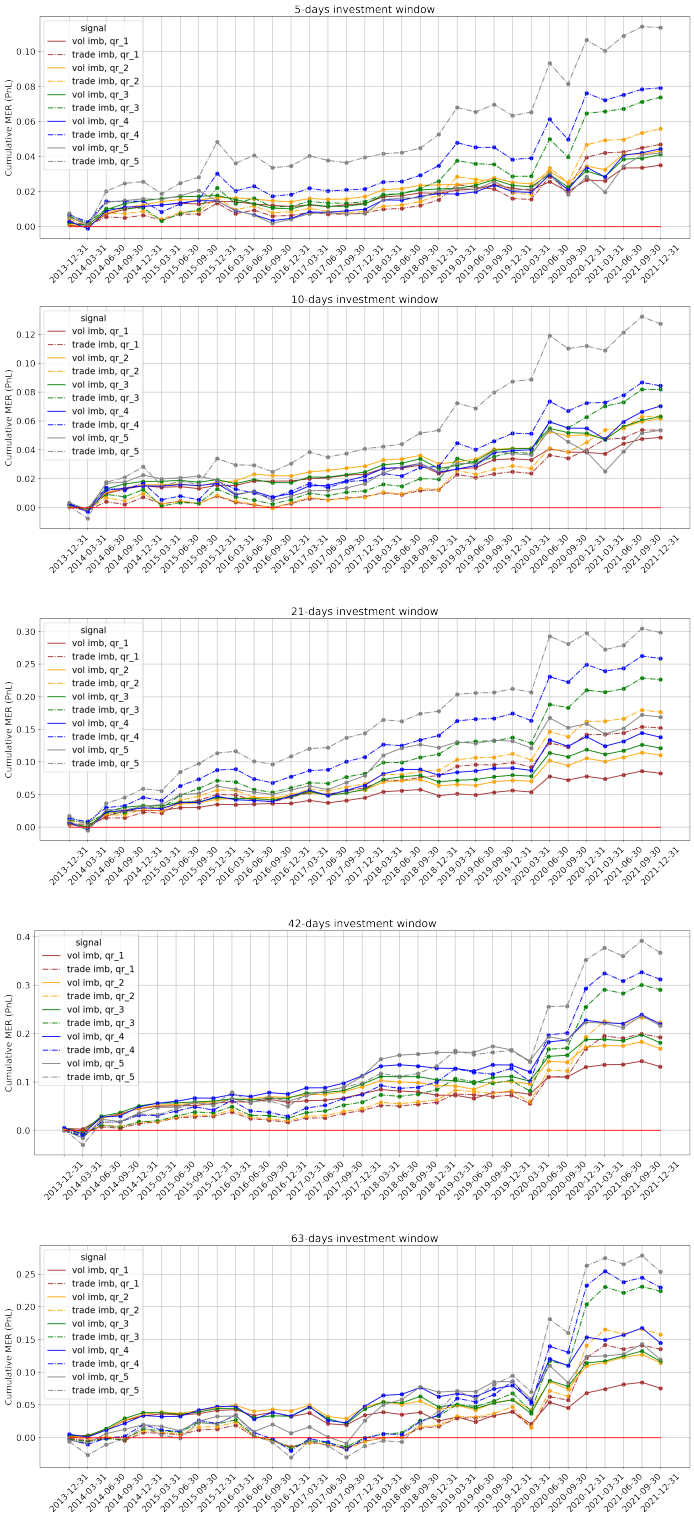


Figure A.1: Time series of cumulative PnL when trading imbalances applying the threshold $N = 50$.

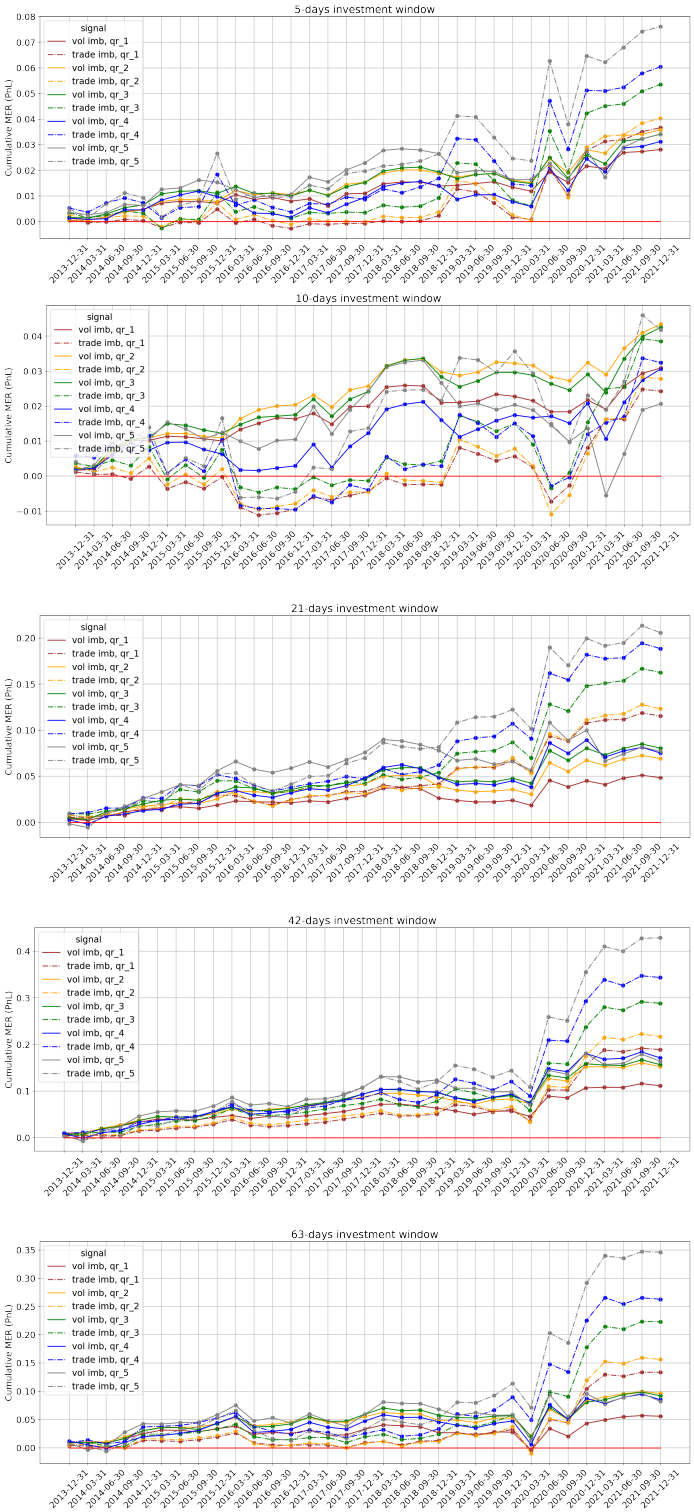


Figure A.2: Time series of cumulative PnL when trading imbalances applying the threshold $N = 150$.

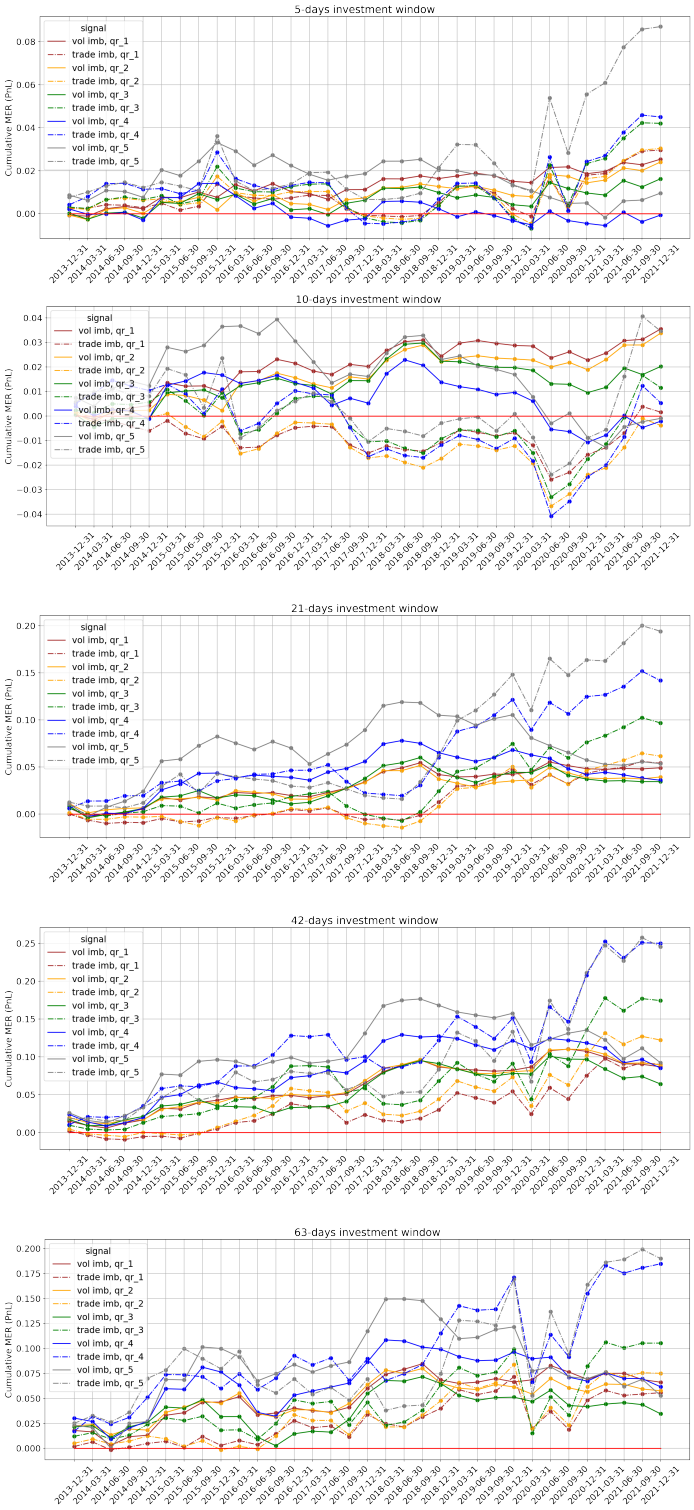


Figure A.3: Time series of cumulative PnL when trading imbalances applying threshold the $N = 500$.

A.2 Evolution of the popularity of stocks within SIC sectors

For each period p and SIC sector, we compute the average number of active funds N on the related stocks. Figure A.4 shows this evolution of "popularity" for a sample of sectors (chosen from the number of securities recovered) across time, and compares it to the cumulative average MER when holding the related stocks.

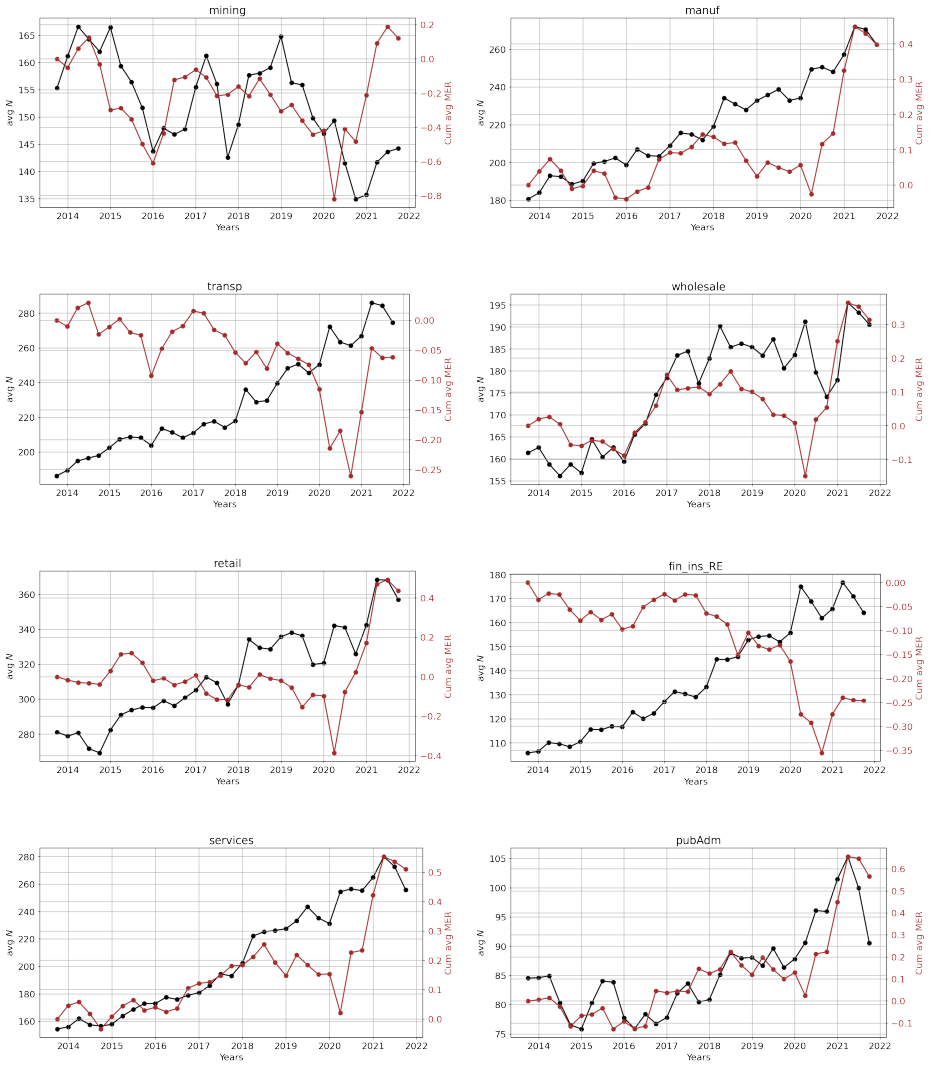


Figure A.4: Evolution of the average popularity of stocks within sectors and related overall performance.

Appendix B

Appendix of Chapter 4

B.1 Lists of Indices

B.1.1 LT data set

The full list of indices with related (ID) is reported for the LT data set. Indices are divided into 4 asset classes, namely *equity, commodity, fixed income* and *cash*.

Equity. ACWI-MSCI All Country World (0), ACWIxUS-MSCI All Country World ex-US (1), EAFE-MSCI EAFE (2), EM-MSCI Emerging Markets (3), DM-MSCI World ex-US (4), World-MSCI World (5), S&P 500 (6), EQ-Dow Jones US Total Stock Market (7), Russell 2000 (8), NREQ-S&P North Am. Natural Resources Sect. (9).

Commodity. COMM - BBG Commodity (10).

Fixed income. BBG Barcalys: Global Treas. (11), Global Treas. ex-US (12), IG-US Aggr. Bond (13), US Long Treas. (14), US Long IG Corp. Bonds (15), IT-US Interm. Treas. Bond (16), TIP- US TIPS (17), US 1-10Y TIPS (18), MUNI-Municipal Bond (19). Then, HY-ICE BofAML US HY Constr. (20) and BBG US STRIPS 25-30Y (21).

Cash. BBG Barclays US 1-3 Months Treasury Bills (22).

B.1.2 BBG data set

The full list of indices considered in the Bloomberg BBG data set follows. These are divided into 6 asset classes, namely *commodity, currency, equity, volatility, bond spreads* and *interest rates*.

Commodity. BBG Commodity Index. Futures ′22: GOLD 100 OZ Jun, NATURAL GAS Jul, BRENT CRUDE Aug, WTI CRUDE Jul, SOYBEAN Jul, CORN Jul, COPPER Jul, Low Su Gasoil G Jun, SILVER Jul, LIVE CATTLE Jun, SOYBEAN OIL Jul, WHEAT (CBT) Jul, LME PRI ALUM Jun, LME NICKEL Jun, NY Harb ULSD Jul, SOYBEAN MEAL Jul, LME ZINC Jun, SUGAR #11 (WORLD) Jul, COFFEE C Jul, KC HRW WHEAT Jul, LEAN HOGS Jun, COTTON NO.2 Jul.

Currency (-USD). AUD, BRL, CAD, CHF, CLP, CNY, COP, CZK, EUR, GBP, HUF, IDR, INR, JPY, KRW, MXN, MYR, NOK, NZD, PEN, PHP, PLN, RUB, SEK, SGD, THB, TRY, ZAR.

Equity (MSCI Index in USD). World, North America, EAFE, Europe, Nordic, Pacific, Far East, Canada, USA, Austria, Belgium, Denmark, Finland, France, Germany, Israel, Ireland, Italy, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, UK, Australia, Hong Kong, Japan, New Zealand, Singapore, EM Europe, EM Latin Am, EM Asia, North America Small Cap, EAFE Small Cap, Pacific Small Cap. For each one of Europe, USA and EM: Energy, Materials, Industrials, Consumer Discret., Consumer Staples, Health Care, Financials, Infor. Tech., Comm. Services, Util. Sector.

Volatility. Cboe, Cboe NDX, Cboe RSL2000, VSTOXX, MOVE, NIKKEI, HSI, JPM JP, Cboe20+Y TreasBnd, JPM G7, JPM EM.

Bond Spreads. US Corp: 1-5Y, 5-10Y, 10-25Y, 25+Y. Pan-Euro Corp. Pan-Euro Corp: 1-3Y, 3-5Y, 5-7Y, 7-10Y, 10+Y. Global Agg Index, US IG Corp, US HY Corp, US Gov-Related, US ABS, US CMBS, US MBS, CAD Credit, EuroAgg Gov-Related, EuroAgg Securitized, EuroAgg High Yield. Sterling Agg: Corp, Sterling Aggregate: Gov-Related, Japan Corp, Australia Corp, China Corp, EM USD IG, EM USD HY.

Int. Rates (2,10,30Y). AU, CA, CL, CN, CZ, DE, DK, FI, GB, HK, HU, ID, JP, KR, MX, MY, NO, NZ, PL, SE, SG, SZ, TH, US CMT, ZA.

Appendix C

Appendix of Chapter 5

C.1 Sub-universes of pools, cases A/B1/B2/C1/C2/C3

The pools found most significant by the proposed methodology are here reported, for cases $A/B1/B2/C1/C2/C3$. We provide three different tables, i.e. Table C.1 for case A first, followed by Table C.2 for cases $B1, B2$, and concluding with Table C.3 for cases $C1, C2, C3$. Each table lists both pools relevant for liquidity consumption (LT data) and liquidity provision (LP data) investigations, and it further highlights the thresholds chosen to define the giant components as described in Section 5.3.

Table C.1: Final set of pools to consider for case A, i.e. for analyses spanning the six months from January to June 2022. Thresholds are listed as (threshold for common origins, for common senders, and for bridges)

Case	Agent Type	Thresholds	Final Pools
A	LT	(2,000, 100, 800)	DAI-WETH/3000, CEL-WETH/3000, USDC-UOS/10000, DAI-USDC/100, SPELL-WETH/3000, WETH-CRV/10000, USDC-USDT/500, DAI-FRAX/500, WETH-BTRFLY/10000, GALA-WETH/3000, WETH-USDT/3000, WBTC-USDC/3000, DAI-USDT/500, UNI-WETH/3000, WETH-ENS/3000, DAI-USDC/500, WBTC-WETH/500, MATIC-WETH/3000, DAI-WETH/500, WETH-USDT/500, USDC-WETH/500, LINK-WETH/3000, WBTC-WETH/3000, FXS-WETH/10000, FRAX-USDC/500, USDC-WETH/3000, USDC-WETH/10000, LUSD-USDC/500, HEX-USDC/3000, USDC-NCR/500, SHIB-WETH/3000, DYDX-WETH/3000, USDC-USDT/100, HEX-WETH/3000

Continued on next page

Table C.1 – continued from previous page

Case	Agent Type	Thresholds	Final Pools
A	LP	(30, 3, NA)	WETH-CRV/10000, MKR-WETH/3000, WETH-USDT/3000, WBTC-USDC/3000, UNI-WETH/3000, WETH-ENS/3000, WBTC-WETH/500, MATIC-WETH/3000, DAI-WETH/500, WETH-USDT/500, USDC-WETH/500, LINK-WETH/3000, WBTC-WETH/3000, USDC-WETH/3000, SHIB-WETH/3000, WBTC-USDT/3000, USDC-USDT/100, USDC-USDT/500, SHIB-WETH/10000

Table C.2: Final set of pools to consider for cases $B1, B2$. Thresholds are listed as (threshold for common origins, for common senders, and for bridges)

Case	Agent Type	Thresholds	Final Pools
B1	LT	(1,500, 60, 600)	DAI-WETH/500, FRAX-USDC/500, SPELL-WETH/3000, agEUR-USDC/500, USDC-USDT/100, LUSD-USDC/500, USDC-UST/100, WBTC-WETH/3000, DAI-USDC/100, USDC-NCR/500, WETH-USDT/500, XSGD-USDC/500, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, WETH-BTRFLY/10000, FEI-USDC/500, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, USDC-USDT/500, XSGD-WETH/500, WBTC-WETH/500, FXS-WETH/10000, GALA-WETH/3000, WBTC-USDC/3000, WETH-CRV/10000, WETH-USDT/3000, USDC-UOS/10000, HEX-WETH/3000, USDC-WETH/500, USDC-GF/3000
B1	LP	(20, 3, NA)	DAI-WETH/500, WBTC-USDC/3000, USDC-WETH/3000, LINK-WETH/3000, WETH-CRV/10000, WBTC-WETH/3000, MATIC-WETH/3000, WETH-ENS/3000, USDC-USDT/100, UNI-WETH/3000, SHIB-WETH/3000, USDC-WETH/500, WETH-USDT/500, MKR-WETH/3000, SHIB-WETH/10000, WBTC-WETH/500
B2	LT	(1,500, 80, 600)	DAI-WETH/500, FRAX-USDC/500, FEI-USDC/100, USDC-USDT/100, UST-WETH/3000, HDRN-USDC/10000, USDC-STG/3000, CEL-WETH/3000, DAI-USDC/500, WBTC-WETH/3000, DAI-USDC/100, WETH-USDT/500, APE-WETH/3000, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, USDC-WETH/10000, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, USDC-USDT/500, APE-USDC/3000, WBTC-WETH/500, WBTC-USDC/3000, WETH-USDT/3000, WETH-LUNA/10000, USDC-UOS/10000, BUSD-USDC/500, HEX-WETH/3000, WETH-LOOKS/3000, SHIB-WETH/3000, USDC-WETH/500, DAI-FRAX/500

Continued on next page

Table C.2 – continued from previous page

Case	Agent Type	Thresholds	Final Pools
B2	LP	(15, 3, NA)	WBTC-USDT/3000, DAI-WETH/500, WBTC-USDC/3000, LINK-WETH/3000, USDC-WETH/3000, WETH-USDT/3000, WETH-LUNA/10000, HEX-USDC/3000, WBTC-WETH/3000, MATIC-WETH/3000, USDC-USDT/500, HEX-WETH/3000, UNI-WETH/3000, USDC-WETH/500, WETH-USDT/500, SHIB-WETH/10000

Table C.3: Final set of pools to consider for cases $C1, C2, C3$. Thresholds are listed as (threshold for common origins, for common senders, and for bridges)

Case	Agent Type	Thresholds	Final Pools
C1	LT	(1,000, 50, 400)	DAI-WETH/500, FRAX-USDC/500, SPELL-WETH/3000, agEUR-USDC/500, USDC-USDT/100, USDC-UST/100, WBTC-WETH/3000, DAI-USDC/100, USDC-NCR/500, WETH-USDT/500, XSGD-USDC/500, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, WETH-BTRFLY/10000, FEI-USDC/500, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, SOS-WETH/10000, XSGD-WETH/500, WBTC-WETH/500, FXS-WETH/10000, GALA-WETH/3000, WBTC-USDC/3000, WETH-CRV/10000, WETH-USDT/3000, HEX-WETH/3000, USDC-WETH/500
C1	LP	(10, 3, NA)	DAI-WETH/500, WBTC-USDC/3000, LINK-WETH/3000, USDC-WETH/3000, WETH-CRV/10000, WETH-BTRFLY/10000, WBTC-WETH/3000, WETH-ENS/3000, MATIC-WETH/3000, UNI-WETH/3000, USDC-WETH/500, SHIB-WETH/3000, WETH-USDT/500, MKR-WETH/3000, SHIB-WETH/10000
C2	LT	(1,000, 50, 500)	DAI-WETH/500, FRAX-USDC/500, WETH-WRLD/10000, USDC-USDT/100, DAI-USDC/500, USDC-UST/100, WBTC-WETH/3000, DAI-USDC/100, USDC-NCR/500, WETH-USDT/500, XSGD-USDC/500, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, WETH-BTRFLY/10000, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, XSGD-WETH/500, WBTC-WETH/500, FXS-WETH/10000, GALA-WETH/3000, WBTC-USDC/3000, WETH-USDT/3000, HEX-WETH/3000, USDC-RSS3/3000, WETH-LOOKS/3000, SHIB-WETH/3000, USDC-WETH/500, DAI-FRAX/500

Continued on next page

Table C.3 – continued from previous page

Case	Agent Type	Thresholds	Final Pools
C2	LP	(10, 3, NA)	DAI-WETH/500, WBTC-USDC/3000, LINK-WETH/3000, USDC-WETH/3000, WETH-USDT/3000, WETH-LUNA/10000, WETH-WRLD/10000, WBTC-WETH/3000, MATIC-WETH/3000, USDC-USDT/100, UNI-WETH/3000, SHIB-WETH/3000, USDC-WETH/500, WETH-USDT/500, MKR-WETH/3000, SHIB-WETH/10000
C3	LT	(1,000, 75, 500)	DAI-WETH/500, FRAX-USDC/500, FEI-USDC/100, USDC-USDT/100, UST-WETH/3000, HDRN-USDC/10000, CEL-WETH/3000, WBTC-WETH/3000, DAI-USDC/100, WETH-USDT/500, UNI-WETH/3000, APE-WETH/3000, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, USDC-USDT/500, APE-USDC/3000, WBTC-WETH/500, WBTC-USDC/3000, WETH-CRV/10000, WETH-USDT/3000, WETH-LUNA/10000, BUSD-USDC/500, HEX-WETH/3000, WETH-LOOKS/3000, SHIB-WETH/3000, USDC-WETH/500
C3	LP	(10, 3, NA)	WBTC-USDT/3000, DAI-WETH/500, WBTC-USDC/3000, USDC-WETH/3000, UNI-USDC/3000, WETH-USDT/3000, WETH-LUNA/10000, HEX-USDC/3000, HEX-WETH/3000, WETH-USDT/500, USDC-WETH/500

Bibliography

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774 (2023).

[2] Adams, H. Uniswap Whitepaper. `https://hackmd.io/@HaydenAdams/HJ9jLsfTz`.

[3] Adams, H., Zinsmeister, N., and Robinson, D. Uniswap v2 core. Tech. rep., Uniswap, Tech. Rep. (2020).

[4] Adams, H., Zinsmeister, N., Salem, M., Keefer, R., and Robinson, D. Uniswap v3 core. Tech. rep., Uniswap, Tech. Rep. (2021).

[5] Anantharama, N., Angus, S., and O'Neill, L. CANarEx: Contextually Aware Narrative Extraction for Semantically Rich Text-as-data Applications. In Findings of the Association for Computational Linguistics: EMNLP 2022 (2022), pp. 3551–3564.

[6] Ang, A., and Bekaert, G. How Regimes Affect Asset Allocation. Financial Analysts Journal 60, 2 (2004), 86–99.

[7] Angelov, D. Top2vec: Distributed representations of topics. arXiv preprint arXiv:2008.09470 (2020).

[8] Angeris, G., Kao, H.-T., Chiang, R., Noyes, C., and Chitra, T. An analysis of Uniswap markets. arXiv e-prints (2019), arXiv–1911.

[9] Anton, M., and Polk, C. Connected stocks. The Journal of Finance 69, 3 (2014), 1099–1127.

[10] Arthur, D., Vassilvitskii, S., et al. k-means++: The advantages of careful seeding. In Soda (2007), vol. 7, pp. 1027–1035.

[11] Ash, E., Gauthier, G., and Widmer, P. Relatio: Text semantics capture political and economic narratives. Political Analysis 32, 1 (2024), 115–132.

[12] Bailey, D. H., Borwein, J. M., de Prado, M. L., and Zhu, Q. J. Pseudomathematics and financial charlatanism: The effects of backtest over fitting on out-of-sample performance. Notices of the AMS 61, 5 (2014), 458–471.

[13] Bailey, D. H., and Lopez de Prado, M. The Sharpe ratio efficient frontier. Journal of Risk 15, 2 (2012), 13.

[14] Bailey, D. H., and López de Prado, M. The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting and non-normality. Journal of Portfolio Management 40, 5 (2014), 94–107.

[15] Bar-On, Y., and Mansour, Y. Uniswap liquidity provision: An online learning approach. In International Conference on Financial Cryptography and Data Security (2023), Springer, pp. 247–261.

[16] Barth, D., and Kahn, R. J. Hedge funds and the Treasury cash-futures disconnect. OFR WP (2021), 21–01.

[17] Berg, J. A., Fritsch, R., Heimbach, L., and Wattenhofer, R. An empirical study of market inefficiencies in Uniswap and SushiSwap. In International Conference on Financial Cryptography and Data Security (2022), Springer, pp. 238–249.

[18] Bergault, P., Bertucci, L., Bouba, D., and Guéant, O. Automated Market Makers: Mean-variance analysis of LPs payoffs and design of pricing functions. Digital Finance (2023), 1–23.

[19] Bhaskar, A., Fabbri, A. R., and Durrett, G. Prompted opinion summarization with GPT-3.5. arXiv preprint arXiv:2211.15914 (2022).

[20] Białkowski, J., Darolles, S., and Le Fol, G. Improving VWAP strategies: A dynamic volume approach. Journal of Banking & Finance 32, 9 (2008), 1709–1722.

[21] Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. Econometric measures of systemic risk in the finance and insurance sectors. Journal of Financial Economics (2011).

[22] Billio, M., and Pelizzon, L. Interconnectedness and systemic risk: hedge funds, banks, insurance companies. BANCARIA 6 (06 2014), 81–91.

[23] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008, 10 (2008), P10008.

[24] Borghesi, C., Marsili, M., and Micciché, S. Emergence of time-horizon invariant correlation structure in financial returns by subtraction of the market mode. Phys. Rev. E 76 (Aug 2007), 026104.

[25] Brown, G. W., Howard, P., and Lundblad, C. T. Crowded trades and tail risk. The Review of Financial Studies 35, 7 (2022), 3231–3271.

[26] Brown, S. J., and Schwarz, C. Do market participants care about portfolio disclosure? Evidence from hedge funds' 13F filings. Evidence from Hedge Funds' 13F Filings (December 17, 2020) (2020).

[27] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. Advances in neural information processing systems 33 (2020), 1877–1901.

[28] Brownlees, C. T., Cipollini, F., and Gallo, G. M. Intra-daily volume modeling and prediction for algorithmic trading. Journal of Financial Econometrics 9, 3 (2011), 489–518.

[29] Bucci, F., Mastromatteo, I., Eisler, Z., Lillo, F., Bouchaud, J.-P., and Lehalle, C.-A. Co-impact: Crowding effects in institutional trading activity. Quantitative Finance 20, 2 (2020), 193–205.

[30] Buterin, V. Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform, 2013.

[31] Bybee, L., Kelly, B. T., Manela, A., and Xiu, D. The structure of economic news. Tech. rep., National Bureau of Economic Research, 2020.

[32] Caccioli, F., Shrestha, M., Moore, C., and Farmer, J. D. Stability analysis of financial contagion due to overlapping portfolios. Journal of Banking & Finance 46 (2014), 233–245.

[33] Cao, S., Da, Z., Jiang, D., and Yang, B. Do hedge funds strategically misreport their holdings? Evidence from 13F restatements. Evidence from 13F Restatements (September 12, 2022) (2022).

[34] Cartea, Á., Cohen, S. N., Graumans, R., Labyad, S., Sánchez-Betancourt, L., and van Veldhuijzen, L. Statistical predictions of trading strategies in electronic markets. Available at SSRN 4442770 (2023).

[35] Cartea, Á., Drissi, F., and Monga, M. Decentralised finance and automated market making: Execution and speculation. arXiv preprint arXiv:2307.03499 (2023).

[36] Cartea, Á., Drissi, F., and Monga, M. Decentralised finance and automated market making: Predictable loss and optimal liquidity provision. arXiv preprint arXiv:2309.08431 (2023).

[37] Cartea, Á., Drissi, F., and Monga, M. Execution and statistical arbitrage with signals in multiple automated market makers. In 2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW) (2023), IEEE, pp. 37–42.

[38] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16 (jun 2002), 321–357.

[39] Chen, H., and Koga, H. Gl2vec: Graph embedding enriched by line graphs with edge features. In Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part III 26 (2019), Springer, pp. 3–14.

[40] Chen, R., Feng, Y., and Palomar, D. Forecasting intraday trading volume: a Kalman filter approach. Available at SSRN 3101695 (2016).

[41] Chen, W., Rabhi, F., Liao, W., and Al-Qudah, I. Leveraging State-of-the-Art Topic Modeling for News Impact Analysis on Financial Markets: A Comparative Study. Electronics 12, 12 (2023).

[42] Chouliaras, A. Institutional Investors, Analysts' Recommendations, Annual Reports, Textual Analysis and Stock Returns: Evidence from SEC EDGAR 10-K and 13-F Forms. Analysts' Recommendations, Annual Reports, Textual Analysis and Stock Returns: Evidence from SEC EDGAR (2015).

[43] Cohen, S. N., Sabaté-Vidales, M., Šiška, D., and Szpruch, Ł. Inefficiency of CFMs: hedging perspective and agent-based simulations. In International Conference on Financial Cryptography and Data Security (2023), Springer, pp. 303–319.

[44] Conrad, T., Vinciguerra, A., and Méroué, G. About constant-product automated market makers. arXiv preprint arXiv:2301.08558 (2023).

[45] Cont, R., Cucuringu, M., Glukhov, V., and Prenzel, F. Analysis and modeling of client order flow in limit order markets. Quantitative Finance 23, 2 (2023), 187–205.

[46] Cont, R., Cucuringu, M., and Zhang, C. Price impact of order flow imbalance: Multi-level, cross-asset and forecasting, 2021.

[47] Costa, L. d. F. Further generalizations of the Jaccard index. arXiv preprint arXiv:2110.09619 (2021).

[48] Cucuringu, M., Davies, P., Glielmo, A., and Tyagi, H. SPONGE: A generalized eigenproblem for clustering signed networks. In The 22nd International Conference on Artificial Intelligence and Statistics (2019), PMLR, pp. 1088–1098.

[49] Cucuringu, M., Li, H., Sun, H., and Zanetti, L. Hermitian matrices for clustering directed graphs: insights and applications. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (26–28 Aug 2020), S. Chiappa and R. Calandra, Eds., vol. 108 of Proceedings of Machine Learning Research, PMLR, pp. 983–992.

[50] Dias, J. G., Vermunt, J. K., and Ramos, S. Clustering financial time series: New insights from an extended hidden Markov model. European Journal of Operational Research 243, 3 (2015), 852–864.

[51] Ellul, A., Jotikasthira, C., Kartasheva, A. V., Lundblad, C. T., and Wagner, W. Insurers As Asset Managers and Systemic Risk. Kelley School of Business Research Paper No. 18-4 and SMU Cox School of Business Research Paper No. 18-18 (06 2020).

[52] Fan, Z., Marmolejo-Cossío, F. J., Altschuler, B., Sun, H., Wang, X., and Parkes, D. Differential Liquidity Provision in Uniswap v3 and Implications

for Contract Design. In Proceedings of the Third ACM International Conference on AI in Finance (2022), pp. 9–17.

[53] Freeman, L. C. A Set of Measures of Centrality Based on Betweenness. Sociometry 40, 1 (1977), 35–41.

[54] Frtisch, R., Käser, S., and Wattenhofer, R. The economics of automated market makers. In Proceedings of the 4th ACM Conference on Advances in Financial Technologies (2022), pp. 102–110.

[55] Gentzkow, M., Kelly, B., and Taddy, M. Text as Data. Journal of Economic Literature 57, 3 (September 2019), 535–74.

[56] Girardi, G., Hanley, K. W., Nikolova, S., Pelizzon, L., and Sherman, M. G. Portfolio similarity and asset liquidation in the insurance industry. Journal of Financial Economics 142, 1 (2021), 69–96.

[57] Goldstein, I., Jiang, H., and Ng, D. T. Investor flows and fragility in corporate bond funds. Journal of Financial Economics 126, 3 (2017), 592–613.

[58] Gómez, S., Jensen, P., and Arenas, A. Analysis of community structure in networks of correlated data. Physical Review E 80, 1 (jul 2009).

[59] Granger, C. W. J. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. Econometrica 37, 3 (1969), 424–438.

[60] Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794 (2022).

[61] Grover, A., and Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (2016), pp. 855–864.

[62] Guo, W., Minca, A., and Wang, L. The topology of overlapping portfolio networks. Statistics & Risk Modeling 33, 3-4 (2016), 139–155.

[63] Gupta, U. GPT-InvestAR: Enhancing Stock Investment Strategies through Annual Report Analysis with Large Language Models. arXiv preprint arXiv:2309.03079 (2023).

[64] Heimbach, L., Schertenleib, E., and Wattenhofer, R. Risks and returns of Uniswap v3 liquidity providers. In Proceedings of the 4th ACM Conference on Advances in Financial Technologies (2022), pp. 89–101.

[65] Heimbach, L., Wang, Y., and Wattenhofer, R. Behavior of liquidity providers in decentralized exchanges. arXiv preprint arXiv:2105.13822 (2021).

[66] Hisano, R., Sornette, D., Mizuno, T., Ohnishi, T., and Watanabe, T. High quality topic extraction from business news explains abnormal financial market volatility. PloS one 8, 6 (2013), e64846.

[67] Huang, S., Jiang, W., Liu, X., and Liu, X. Does liquidity management induce fragility in Treasury prices: Evidence from bond mutual funds. Available at SSRN 3689674 (2021).

[68] Hubert, L., and Arabie, P. Comparing partitions. Journal of classification 2 (1985), 193–218.

[69] Hutto, C., and Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the international AAAI conference on web and social media (2014), vol. 8, pp. 216–225.

[70] Inzirillo, H., and De Quénetain, S. Managing Risk in DeFi Portfolios. arXiv preprint arXiv:2205.14699 (2022).

[71] James, G., Witten, D., Hastie, T., Tibshirani, R., et al. An introduction to statistical learning, vol. 112. Springer, 2013.

[72] Keith Norambuena, B. F., Mitra, T., and North, C. A Survey on Event-Based News Narrative Extraction. ACM Comput. Surv. 55, 14s (jul 2023).

[73] Kirilenko, A., Kyle, A. S., Samadi, M., and Tuzun, T. The flash crash: High-frequency trading in an electronic market. The Journal of Finance 72, 3 (2017), 967–998.

[74] Kitzler, S., Victor, F., Saggese, P., and Haslhofer, B. Disentangling decentralized finance (DeFi) compositions. ACM Transactions on the Web 17, 2 (2023), 1–26.

[75] Kruttli, M., Monin, P., Petrasek, L., and Watugala, S. LTCM redux? Hedge fund Treasury trading and funding fragility during the COVID-19 crisis.

[76] Kuhle, W. Thought Viruses and Asset Prices. Journal of Behavioral Finance 23, 2 (2022), 123–131.

[77] Le, Q., and Mikolov, T. Distributed representations of sentences and documents. In International conference on machine learning (2014), PMLR, pp. 1188–1196.

[78] Ledoit, O., and Wolf, M. Robust performance hypothesis testing with the Sharpe ratio. Journal of Empirical Finance 15, 5 (2008), 850–859.

[79] Lee, D., and Seung, H. Learning the Parts of Objects by Non-Negative Matrix Factorization. Nature 401 (11 1999), 788–91.

[80] Lillo, F., Miccichè, S., Tumminello, M., Piilo, J., and Mantegna, R. N. How news affects the trading behaviour of different categories of investors in a financial market. Quantitative Finance 15, 2 (2015), 213–229.

[81] Lloyd, S. Least squares quantization in PCM. IEEE Transactions on Information Theory 28, 2 (1982), 129–137.

[82] Machicao, J., Corrêa, E. A., Miranda, G. H. B., Amancio, D. R., and Bruno, O. M. Authorship attribution based on Life-Like Network Automata. PLOS ONE 13, 3 (Mar. 2018), e0193703.

[83] Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., Pfetsch, B., Heyer, G., Reber, U., Häussler, T., Schmid-Petri, H., and Adam, S. Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology. Communication Methods and Measures 12, 2-3 (04 2018), 93–118.

[84] Makarov, I., and Schoar, A. Cryptocurrencies and decentralized finance (DeFi). Tech. rep., National Bureau of Economic Research, 2022.

[85] Management, C. C. F. Packed in like sardines: how crowding in trade flow can adversely affect execution costs.

[86] Mankad, S., Michailidis, G., and Kirilenko, A. Discovering the ecosystem of an electronic financial market with a dynamic machine-learning method. Algorithmic Finance 2, 2 (2013), 151–165.

[87] Marti, G., Andler, S., Nielsen, F., and Donnat, P. Clustering financial time series: How long is enough? arXiv preprint arXiv:1603.04017 (2016).

[88] Marti, G., Nielsen, F., Bińkowski, M., and Donnat, P. A review of two decades of correlations, hierarchies, networks and clustering in financial markets. Progress in information geometry: Theory and applications (2021), 245–274.

[89] Michael, N., Cucuringu, M., and Howison, S. Option Volume Imbalance as a predictor for equity market returns. arXiv preprint arXiv:2201.09319 (2022).

[90] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).

[91] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 26 (2013).

[92] Miori, D., and Cucuringu, M. Returns-Driven Macro Regimes and Characteristic Lead-Lag Behaviour between Asset Classes. arXiv preprint arXiv:2209.00268 (2022).

[93] Miori, D., and Cucuringu, M. SEC Form 13F-HR: Statistical investigation of trading imbalances and profitability analysis. arXiv preprint arXiv:2209.08825 (2022).

[94] Miori, D., and Cucuringu, M. DeFi: Modeling and forecasting trading volume on Uniswap v3 liquidity pools. Available at SSRN 4445351 (2023).

[95] Miori, D., and Cucuringu, M. Clustering Uniswap v3 traders from their activity on multiple liquidity pools, via novel graph embeddings. Digital finance (2024).

[96] Miori, D., and Petrov, C. Narratives from GPT-derived networks of news and a link to financial markets dislocations. International Journal of Data Science and Analytics (2024).

[97] Mitali, S. F. Common holdings and mutual fund performance. Available at SSRN 3448494 (2019).

[98] Musmeci, N., Aste, T., and Di Matteo, T. Risk diversification: a study of persistence with a filtered correlation-network approach. arXiv preprint arXiv:1410.5621 (2014).

[99] Musmeci, N., Aste, T., and Di Matteo, T. Interplay between past market correlation structure changes and future volatility outbursts. Scientific reports 6, 1 (2016), 36320.

[100] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system.

[101] Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., and Jaiswal, S. graph2vec: Learning distributed representations of graphs. arXiv preprint arXiv:1707.05005 (2017).

[102] Nelson, W., Zitnik, M., Wang, B., Leskovec, J., Goldenberg, A., and Sharan, R. To Embed or Not: Network Embedding as a Paradigm in Computational Biology. Frontiers in Genetics 10 (2019).

[103] Newman, M. E. Modularity and community structure in networks. Proceedings of the national academy of sciences 103, 23 (2006), 8577–8582.

[104] Nguyen, N., and Nguyen, D. Global Stock Selection with Hidden Markov Model. Risks 9, 1 (2021).

[105] Papenbrock, J., and Schwendner, P. Handling risk-on/risk-off dynamics with correlation regimes and correlation networks. Financial Markets and Portfolio Management 29, 2 (2015), 125–147.

[106] Pasquariello, P. Financial market dislocations. The Review of Financial Studies 27, 6 (2014), 1868–1914.

[107] Pozzi, F., Di Matteo, T., and Aste, T. Exponential smoothing weighted correlations. The European Physical Journal B 85 (06 2012).

[108] Preis, T., Kenett, D., Stanley, H. E., Helbing, D., and Ben-Jacob, E. Quantifying the behavior of stock correlations under market stress. PNAS 2 (01 2012).

[109] Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., and Yang, D. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? arXiv preprint arXiv:2302.06476 (2023).

[110] Qu, H., and Kazakov, D. Quantifying correlation between financial news and stocks. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (2016), IEEE, pp. 1–6.

[111] Quispe, L. V., Tohalino, J. A., and Amancio, D. R. Using virtual edges to improve the discriminability of co-occurrence text networks. Physica A: Statistical Mechanics and its Applications 562 (2021), 125344.

[112] Rahimikia, E., Zohren, S., and Poon, S.-H. Realised volatility forecasting: Machine learning via financial word embedding. arXiv preprint arXiv:2108.00480 (2021).

[113] Ray, P. P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and Cyber-Physical Systems 3 (2023), 121–154.

[114] Rosenberg, A., and Hirschberg, J. V-measure: A conditional entropy-based external cluster evaluation measure. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL) (2007), pp. 410–420.

[115] Rossetti, G., and Cazabet, R. Community discovery in dynamic networks: A survey. ACM Comput. Surv. 51, 2 (feb 2018).

[116] Ruan, R. Liquidity takers behavior representation through a contrastive learning approach. In Proceedings of the Fourth ACM International Conference on AI in Finance (2023), pp. 601–609.

[117] Saracli, S., Dogan, N., and Dogan, I. Comparison of hierarchical cluster analysis methods by cophenetic correlation. Journal of Inequalities and Applications 2013 (12 2013).

[118] Schär, F. Decentralized finance: On blockchain-and smart contract-based financial markets. FRB of St. Louis Review (2021).

[119] Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-Lehman graph kernels. Journal of Machine Learning Research 12, 9 (2011).

[120] So, E. C., and Wang, S. News-driven return reversals: Liquidity provision ahead of earnings announcements. Journal of Financial Economics 114, 1 (2014), 20–35.

[121] Steinert, R., and Altmann, S. Linking microblogging sentiments to stock price movement: An application of GPT-4. arXiv preprint arXiv:2308.16771 (2023).

[122] Stella, M. Text-mining forma mentis networks reconstruct public perception of the STEM gender gap in social media. PeerJ Computer Science 6 (2020).

[123] Strauss, N., Vliegenthart, R., and Verhoeven, P. Intraday news trading: The reciprocal relationships between the stock market and economic news. Communication Research 45, 7 (2018), 1054–1077.

[124] Syed, S., and Spruit, M. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In 2017 IEEE International conference on data science and advanced analytics (DSAA) (2017), Ieee, pp. 165–174.

[125] Szűcs, B. Á. Forecasting intraday volume: Comparison of two early models. Finance Research Letters 21 (2017), 249–258.

[126] Thomson, J. B. On systemically important financial institutions and progressive systemic mitigation. DePaul Bus. & Comm. LJ 8 (2009), 135.

[127] Volpati, V., Benzaquen, M., Eisler, Z., Mastromatteo, I., Tóth, B., and Bouchaud, J.-P. Zooming in on equity factor crowding. arXiv preprint arXiv:2001.04185 (2020).

[128] Wang, Matthew and Lin, Yi-Hong and Mikhelson, Ilya. Regime-switching factor investing with hidden markov models. Journal of Risk and Financial Management 13, 12 (2020).

[129] Weng, M.-H., Wu, S., and Dyer, M. Identification and Visualization of Key Topics in Scientific Publications with Transformer-Based Language Models and Document Clustering Methods. Applied Sciences 12, 21 (2022).

[130] Wermers, R., Yao, T., and Zhao, J. Forecasting stock returns through an efficient aggregation of mutual fund holdings. The Review of Financial Studies 25, 12 (2012), 3490–3529.

[131] Wright, I. D., Reimherr, M., and Liechty, J. A machine learning approach to classification for traders in financial markets. Stat 11, 1 (2022), e465.

[132] Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., and Lu, Y. Temporal Data Meets LLM–Explainable Financial Time Series Forecasting. arXiv preprint arXiv:2306.11025 (2023).

[133] Zhang, S., Wang, R.-S., and Zhang, X.-S. Uncovering fuzzy community structure in complex networks. Physical Review E 76, 4 (2007), 046103.

[134] Zhang, Y.-J., Yang, K.-C., and Radicchi, F. Systematic comparison of graph embedding methods in practical tasks. Phys. Rev. E 104 (Oct 2021), 044315.

[135] Zhao, L., Wang, G.-J., Wang, M., Bao, W., Li, W., and Stanley, H. E. Stock market as temporal network. Physica A: Statistical Mechanics and its Applications 506 (2018), 1104–1112.