Intraday Clustering for Financial State Detection

Group P

Kamal Nour | 329597

Yassin Alnuaimee | 326279

Contents

1 Abstract

This project, undertaken as part of the course **FIN-525: Financial Big Data**, investigates intraday clustering techniques for detecting and potentially predicting temporal market states. Using a high-frequency dataset of trade and best bid-offer (BBO) data for the ETF S&P 500 during the month of May, we aim to construct and analyze correlation matrices and apply clustering algorithms, such as the Louvain method, to uncover patterns in intraday market dynamics. Managing this multi-gigabyte dataset poses computational challenges, requiring efficient data processing and memory management. This work has broad applications in algorithmic trading, risk management, and portfolio optimization, offering insights into the complex behavior of one of the world's most influential financial indices.

2 Introduction

In financial markets, high-frequency data holds a wealth of information that can reveal the intricate dynamics of trading activity on a granular timescale. This project, part of the **FIN-525: Financial Big Data** course, focuses on leveraging intraday clustering techniques to detect temporal market states and analyze their underlying structure.

We use a dataset of trade and best bid-offer (BBO) data for the ETF S&P 500, a benchmark index that tracks the largest 500 publicly traded companies in the United States. The data spans the month of May, a historically eventful period in financial markets. For example, May 6, 2010, witnessed the *Flash Crash*, where the Dow Jones Industrial Average plunged nearly 1,000 points within minutes, only to recover most of the loss shortly thereafter. While not every May is marked by such dramatic events, it remains a period of interest for exploring short-term volatility and market anomalies. The data captures key market microstructure features, such as trade volumes, price changes, bid-ask spreads, and order book imbalances, making it ideal for studying short-term fluctuations and emergent patterns in market behavior.

A unique challenge in this project is the dataset's size, which spans several gigabytes. Processing such large volumes of data requires careful optimization to ensure memory efficiency and computational scalability, especially when constructing high-dimensional correlation matrices and applying clustering algorithms.

The project aims to explore how clustering techniques, particularly the Louvain method, can identify distinct intraday market states. These states provide valuable insights for applications in algorithmic trading, risk management, and portfolio optimization, enabling better decision-making in dynamic market conditions. For instance, understanding periods of high volatility or low liquidity can inform strategies tailored to current market states.

By combining advanced data analytics with robust computational strategies, this project contributes to the study of market microstructure and provides practical insights into the behavior of a globally significant financial index.

3 Methodology

3.1 Data Preprocessing and Cleaning

The first step in undertaking any data driven project is to conduct an Exploratory Data Analysis on a subset of the data at hand, deemed interesting, in order to first assess the falacies within the given resources and then come up with a mechanism that generalizes well to the entire data set. Concretely, two stocks were chosen to conduct the EDA, namely, Agilent (A) and Advanced Auto Parts (AAP) since the first is an example of a complete subset of the 392 stocks available, and the second is also complete but more importantly presents some inconsistencies that will need to be handled early on, allowing us to set up a regularization pipeline. Ultimately, the goal of this first step is to generate a clean, regularized matrix of logarithmic returns.

3.1.1 The Challenge of Large Scale

A key aspect of the wrangling process is the choice of Polars over Pandas for handling large-scale datasets. Polars is designed as a high-performance, columnar data processing library that leverages Rust's efficiency to handle large datasets with ease. Unlike Pandas, which operates row-wise and is memory-intensive, Polars supports lazy execution, which was leveraged to reduce computation time: by delaying the execution of operations on the various data frames until the final data frame is formed, making Polars particularly well-suited for high-frequency financial data, where the volume and velocity of data demand a scalable and efficient solution. Quantitatively, Table 1 shows the runtime difference between executing a time stamp type transformation on a single column in Pandas, versus the entire trade-data cleaning pipeline in Polars"for the Agilent subset. The tenfold Wall Time difference between the two libraries, is the main selling-point for the use of Polars throughout the pre-processing.

Library	CPU Time (s)	Wall Time (s)
Pandas	0.672	1.540
Polars	0.032	0.144

Table 1: Performance comparison of Pandas and Polars on Agilent Data.

3.1.2 Trade Data Cleaning

The cleaning of trade data begins with the conversion of raw timestamps into a time-aware format suitable for analysis. The Excel-style timestamps are adjusted to standard datetime objects and localized to the specified exchange timezone. This step ensures that all subsequent temporal analyses align with market conventions and trading hours.

One major issue addressed during the cleaning process is the inclusion of non-regular trades. The script applies filters to exclude special-condition trades by selecting only those marked as “uncategorized”: the AAP Dataset alone contains around 560 entries of these special trades. Taking these trades into consideration complexifies the subsequent analeses tasks by introducing anomalies in the analyzed market’s behavior. In Addition, trades occurring outside regular trading hours are removed based on specified opening and closing times: 8 of these outliers were found in the Agilent dataset. This step reduces noise and ensures the dataset reflects standard trading activity, which is critical for deriving meaningful insights into price trends and market behavior.

Another challenge in trade data is the fragmentation caused by sub-trades. In order to mitigate the effects of the aforementioned anomaly, these sub-trades are aggregated into volume-weighted average prices (VWAP), a standard practice in financial data analysis. This aggregation not only simplifies the dataset but also provides a more accurate representation of the trading price.

Finally, irrelevant columns, such as raw flags, are dropped to streamline the dataset and focus on economically relevant variables.

3.1.3 BBO Data Cleaning and Processing

For the best bid and offer (BBO) data, the cleaning process emphasizes the removal of invalid and redundant quotes. The timestamps are generated using the same paradigm as the trade data processing. Invalid quotes—such as those with negative prices or where bid prices exceed ask prices—are filtered out. This ensures the data used in subsequent analyses adheres to logical market constraints and reflects genuine market conditions.

The issue of redundant quotes with identical timestamps is addressed by retaining only the latest quote at each timestamp. This streamlines the dataset, focusing on actionable information without compromising temporal accuracy. Additionally, quotes outside regular trading hours are filtered out, ensuring consistency with the trade data and aligning the datasets for subsequent joining and analysis.

3.1.4 Return Matrix Computation

During the preprocessing stage of individual stock data, troubleshooting on a stock-by-stock basis revealed challenges that became more complex when scaling to the broader market space, such as the S&P500, all stemming from the same source: joining all the individual return data frames into one. The EDA focused exclusively on Agilent stock, which led to an oversight of critical data type mismatches that only became evident when attempting to construct a pipeline over multiple stocks. For example, Advanced Auto Parts was the first stock to reveal such inconsistencies, highlighting the need for a more robust preprocessing strategy. Instead of discarding inconsistent data, as is often the standard approach, the columns across all stocks were cast to the most common data type to create a more complete dataset. This method ensured a more inclusive analysis, prioritizing the retention of valuable data while maintaining consistency across the dataset.

In parallel, the task of generating a log return matrix revealed additional complexities stemming from the granularity of the stock data. The raw data provided timestamps at a very fine level of detail—down to the second—which became problematic when attempting to merge data across different stocks based on dates. The overly specific nature of the timestamps often resulted in mismatches, leaving the merged dataset empty. To address this, the data was aggregated by time window, first by hours then by finer-grained chunks in order to generate individual chunks that were compatible with one another.This adjustment not only simplifies the merging process but also enhances the likelihood of producing a more complete and reliable dataframe.

These challenges highlight the importance of building a clear and flexible pipeline for processing stock data, both for individual stocks and the entire market. By tackling data type inconsistencies and timestamp granularity issues early, the pipeline can handle the transition from analyzing single stocks to the full market more effectively. These improvements set a strong foundation for further analysis, ensuring the dataset is complete, reliable, and accurately reflects overall market behavior. This first step results in mechanisms that can generate log return matrices over a subset of the S&P500 stocks over time, aggregated by the desired time window, introducing the most crutial hyperparameter of this project: time granularity.

3.2 Constructing and Refining Correlation Matrices

With a robust pipeline for return computation, the generation of a correlation matrix emerges as a major step in analyzing financial market behavior as it quantifies the relationships between timestamps. However, raw correlation matrices often include noise that obscures meaningful patterns. To address this, cleaning methods such as eigenvalue clipping and random matrix filtering are employed to refine the matrix, ensuring it accurately reflects significant relationships. These cleaned matrices not only enhance the quality of the graphs used for clustering, but also improve the detection of clusters, representing communities of cohesive market activity.

3.2.1 Eigenvalue Clipping

Eigenvalue clipping is based on the theoretical distribution of eigenvalues in random matrix theory, specifically the Marčenko-Pastur distribution. When data exhibits random noise, the eigenvalues of its correlation matrix form a bulk distribution within a range defined by λ_{plus} and λ_{minus}, derived as:

where $q = \frac{N}{T}$, N is the number of variables, and T is the number of observations.

In eigenvalue clipping, eigenvalues above λ_{plus} are deemed to represent genuine correlations, while those within the bulk are attributed to noise. To clean the matrix:

1. **Bulk Averaging:** Eigenvalues within the bulk are averaged to preserve the trace of the original matrix.
2. **Scaling:** Eigenvectors corresponding to noisy eigenvalues are scaled by the averaged eigenvalue.
3. **Preservation of Signal Eigenvectors:** Eigenvectors of significant eigenvalues (greater than λ_{plus}) retain their original scaling.

This process yields a cleaned correlation matrix that minimizes noise while maintaining its overall structure. Notably, eigenvalue clipping is computationally efficient, making it suitable for large datasets.

Conducting Eigenvalue clipping on the thirty-minute aggregated returns correlationmatrix yields the results in Figure 1: eigenvalues in the bulk are redistributed to follow the theoritical Marčenko-Pastur distribution, highlighting the strong correlation relations between different timestamps. The two bottom graphs highlight the sanity of the cleaning proceedure.

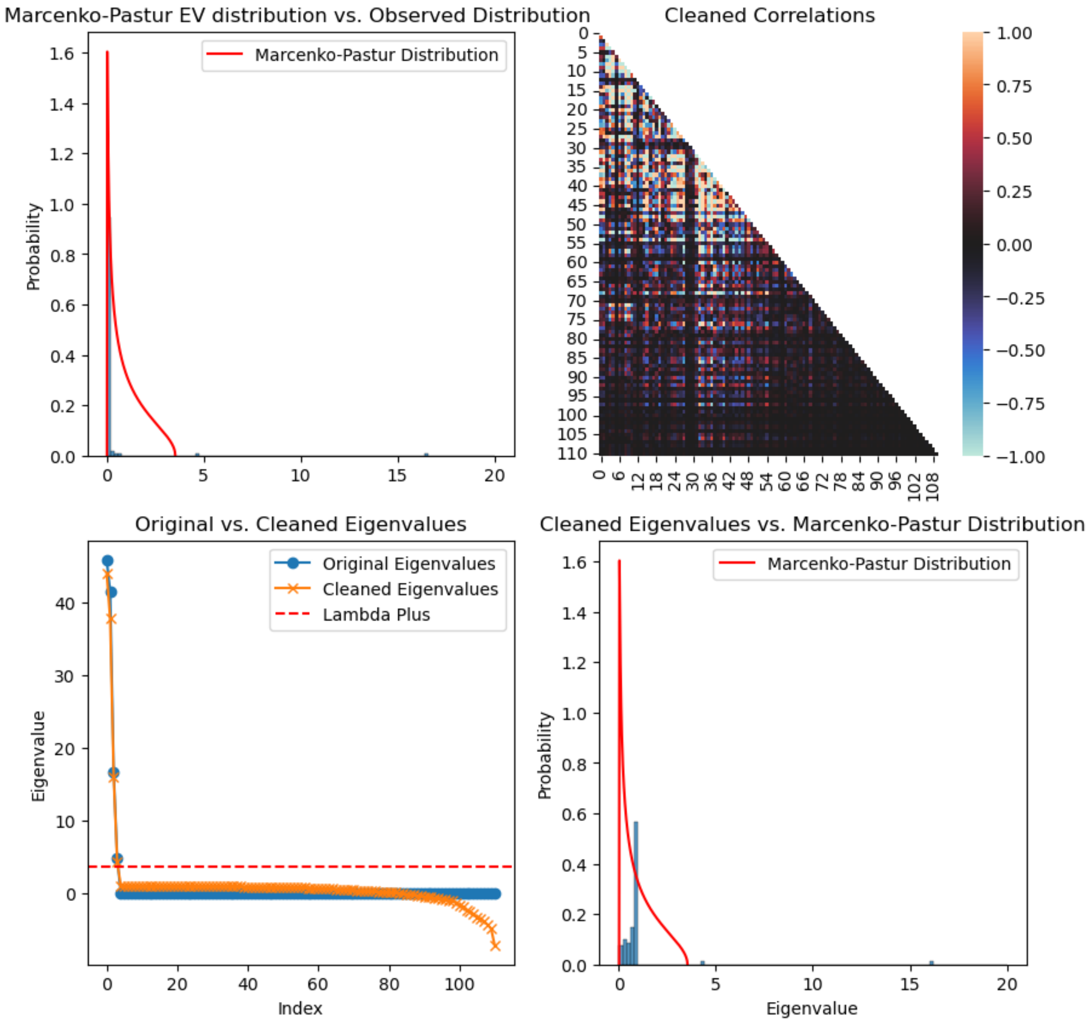

Figure 1: Eigen Values and their Disrtibutions after Clipping

3.2.2 Random Matrix Filtering

Random Matrix Filtering adopts a different approach to refining correlation matrices by explicitly isolating and removing noise. This technique also relies on eigenvalue decomposition but focuses on subtracting a modeled random component. The steps involved in this method are as follows:

1. **Random Component Modeling (C0):** Using random matrix theory (RMT), the bulk eigenvalues of the correlation matrix are modeled to estimate the random component (C0). The largest eigenvalue, often associated with the market mode in financial contexts, is excluded to avoid overestimating noise.

2. **Subtraction:** The random component (C0) is subtracted from the original correlation matrix (C), leaving a matrix that predominantly reflects significant correlations.

3. **Reconstruction:** A cleaned matrix (C') is reconstructed using the remaining significant eigenvalues and their corresponding eigenvectors.

By explicitly removing noise modeled by the random component, C minus C0 cleaning achieves a direct separation of signal and noise. This method is particularly effective in scenarios where a dominant eigenvalue—such as the market mode—skews the overall structure of the correlation matrix. Removing this bias enables a clearer view of subtle yet meaningful relationships among variables, which might otherwise remain undetected.

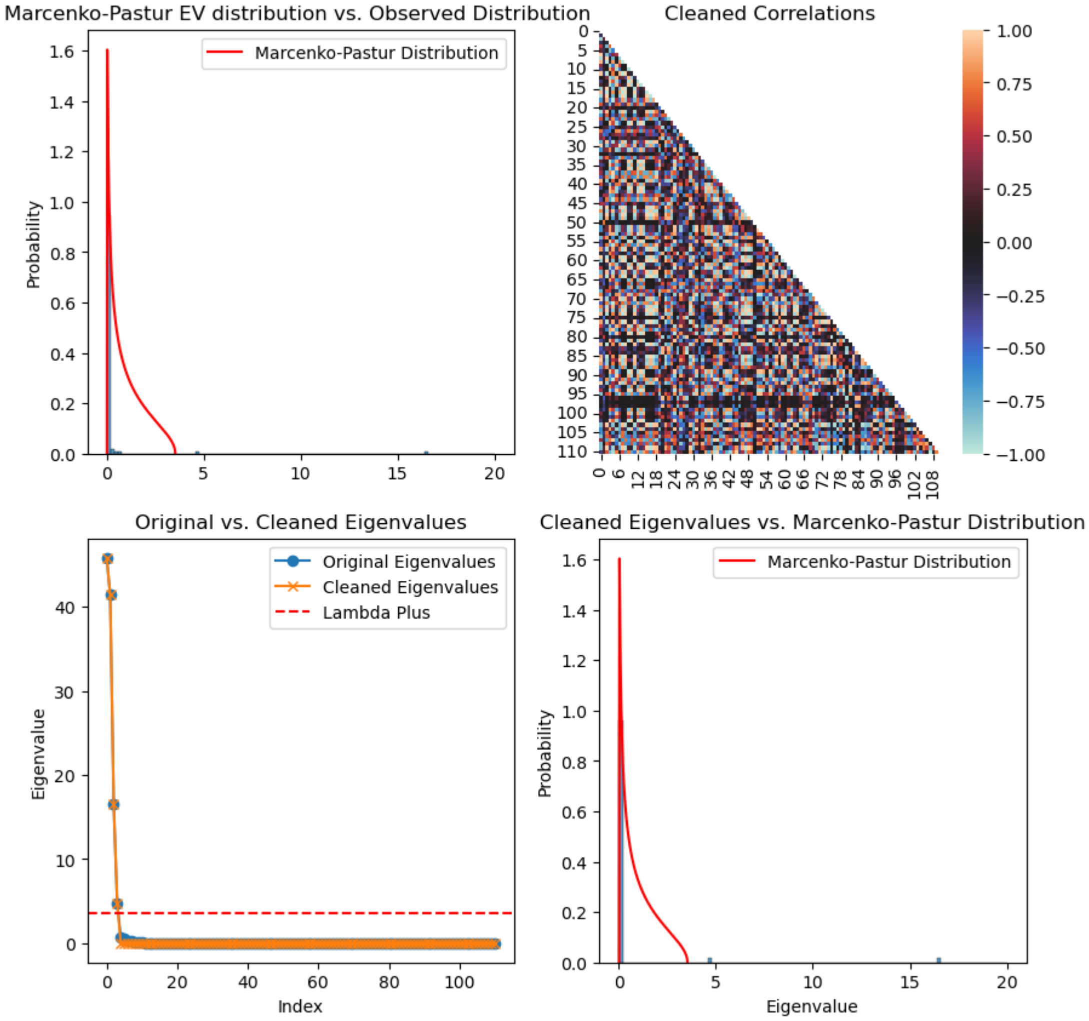

Figure 2: Eigen Values and their Disrtibutions after C minus C0

Random Matrix Filtering on the aforementionned correlation matrix yields a distribution that differ from the previous technique (Figure 2): instead of redistributing the eigenvalues, this method truncates them, flooring the values below the λ_{plus} thershold to zero, which might seem more drastic than eigenvalue clipping but in practice alligns well with the application at hand as described in the subsequent section.

3.2.3 Comparison

Although both eigenvalue clipping and Random Matrix Filtering aim to refine correlation matrices by mitigating noise, they differ significantly in their methodology and focus. Eigenvalue clipping emphasizes preserving the overall structure of the matrix by averaging noisy eigenvalues, ensuring that the cleaned matrix retains its trace. On the other hand, Random Matrix Filtering directly isolates and removes the random component, making it more suitable for contexts where specific biases, such as the market mode, need to be neutralized.

In the context of our application where the cleaned correlation matrix is used to construct a weighted graph where the weight of an edge between two timestamps is represent by the absolute value of the correlation coeffitient between the two nodes, it is projected that hte second method proposed is more suitable: clustering a densly connected graph is first computationally more expensive and second yields less robust results than a sparser graph [2]. That being said, as eigenvalues are floored to zero with Random Matrix Filtering, many edges are thus discarded, hence leading to a sparser graph. In theory, Eigenvalue Clipping is the least favorable candidate for this application. Building on this theoretical insight, the following section examines how these assumptions hold up in practice and evaluates the resulting clustering performance.

3.3 Market State Detection Using Clustering

Identifying market states through clustering involves grouping timestamps into communities that share similar market behavior patterns. Three clustering methods were applied in this project: the Louvain method, the Girvan-Newman algorithm, and the Marsili-Giada method. Each approach has distinct strengths and weaknesses, making them suitable for different types of market analysis.

Our choice to explore these methods is guided by the work of Hendricks, Gebbie, and Wilcox [1], who demonstrated the utility of clustering algorithms, particularly the Louvain method, for detecting temporal market states in financial data. The comparison of methods allows us to assess their effectiveness in identifying meaningful intraday patterns in the dataset. Below, we describe each method, present visualizations of the resulting clusters, computed on 30-minute aggregated data, and justify our decision to use the Louvain method as our primary approach.

3.3.1 The Louvain Method

The Louvain method is a greedy optimization algorithm designed to detect communities in large networks. It operates on a graph where nodes represent timestamps, and edge weights reflect the correlation between returns at these timestamps. The algorithm maximizes modularity, a measure of the density of links within communities relative to those between communities.

The Louvain method begins by assigning each node to its own community. It then evaluates the modularity gain for moving nodes between communities and optimizes modularity through local adjustments. Once local optimization is complete, the graph is compressed by treating each community as a single node, and the process repeats until no further modularity improvement is possible.

This method is particularly effective for timestamp clustering due to its efficiency and ability to handle large datasets, making it a natural choice for financial applications. As demonstrated in the referenced paper, the Louvain method successfully identified intraday market states using financial correlation matrices. Its ability to uncover homogeneous periods within the trading day aligns with our project's goals.

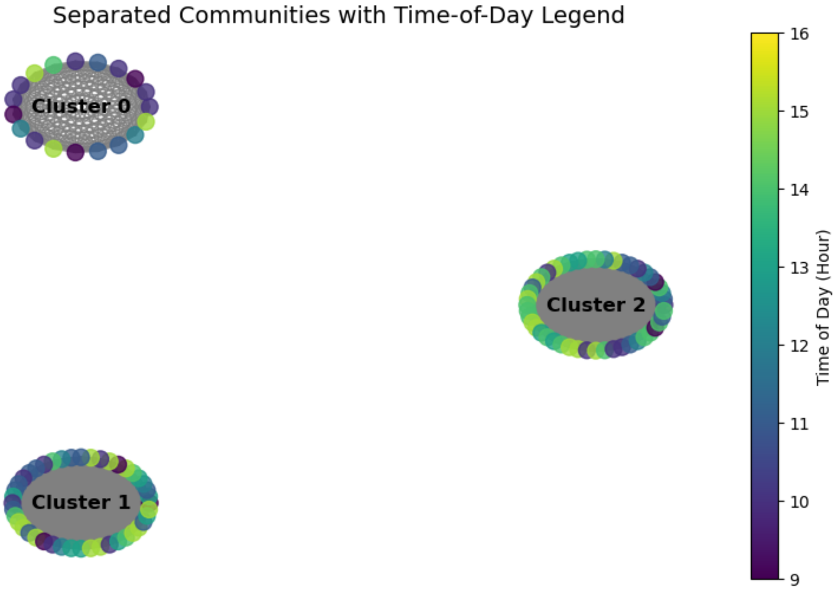

Figure 3: Louvain Clustering for 30-min time window

3.3.2 The Girvan-Newman Algorithm

The Girvan-Newman algorithm is a divisive clustering method that isolates communities by iteratively removing edges with the highest edge betweenness centrality. In the context of timestamp clustering, this approach identifies temporal clusters by focusing on edges that act as bridges between distinct periods.

While effective at uncovering key transitional periods, such as shifts between volatility phases, the Girvan-Newman algorithm is computationally intensive and less scalable for large financial datasets. Additionally, its sensitivity to noise can complicate the detection of clusters in high-frequency financial data.

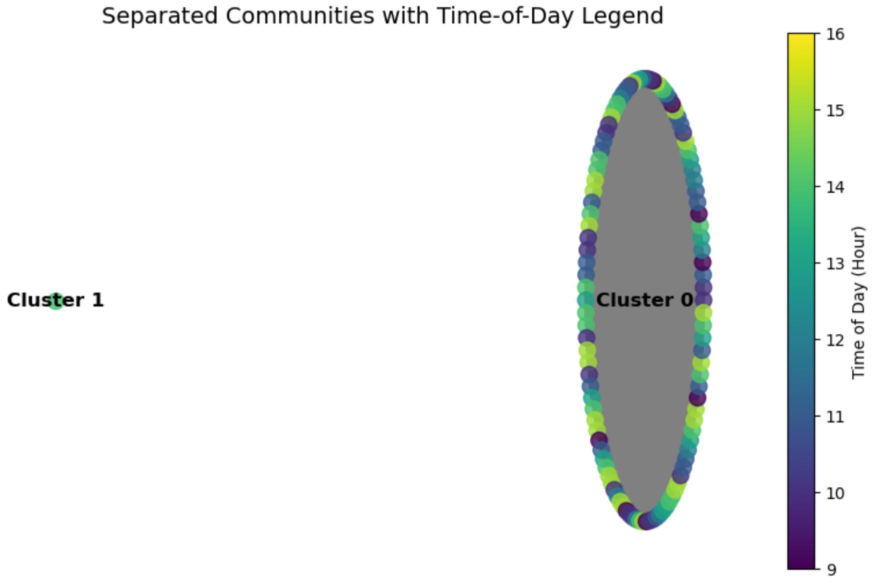

Figure 4: Girvan-Newman Clustering for 30-min time window

3.3.3 The Marsili-Giada Method

The Marsili-Giada method is inspired by statistical mechanics and uses a Potts model framework to detect communities. It optimizes a fitness function that balances strong intra-community correlations with weaker inter-community correlations, effectively minimizing the system's energy.

This method excels in detecting subtle temporal patterns and fine-tuning the resolution of clusters. However, it requires careful parameter selection and is computationally intensive, which can be a limitation for large datasets.

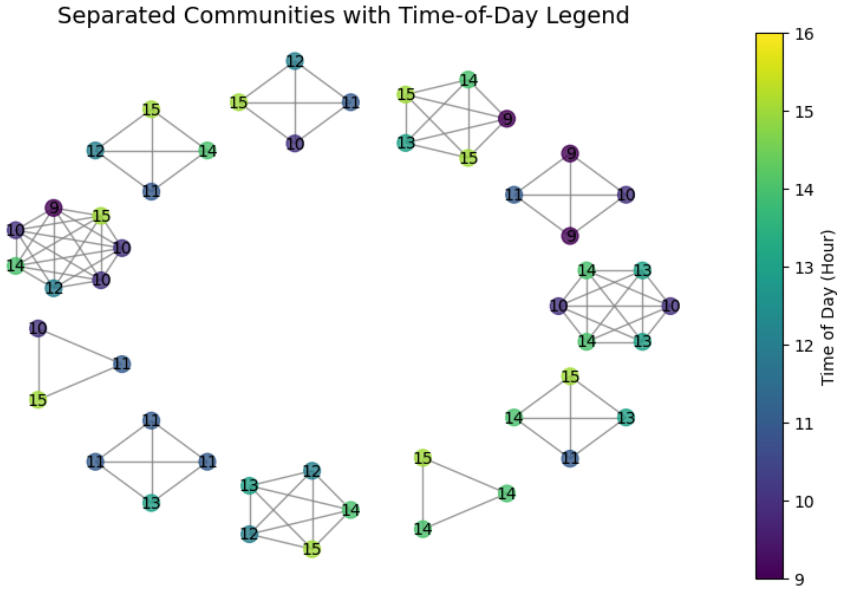

Figure 5: Marsili Giada Clustering for 30-min time window

3.3.4 Conclusion

Each method offers unique advantages for clustering timestamps and identifying market states. The Louvain method stands out due to its efficiency, scalability, and demonstrated effectiveness in the context of financial datasets, as shown in the work of Hendricks et al. Furthermore, the results obtained in our analysis indicate that the Louvain method provides the clearest and most interpretable clusters, making it the optimal choice for our study.

The results of our clustering analysis provide further insights into these methods. The Louvain method (Figure 3) successfully segmented the trading day into three phases: morning (9-10), midday (10-13), and late afternoon (13-16). These phases align with well-known market behaviors, such as the morning reaction to overnight news, midday consolidation, and late-day activity as markets prepare to close.

The Girvan-Newman algorithm (Figure 4), while computationally intensive, revealed a broader segmentation, isolating a small cluster corresponding to an outlier period and a larger cluster spanning the majority of the trading day. This behavior reflects the method's focus on dividing the network by removing high-betweenness edges, which may miss finer temporal patterns. It is important to mention that a sparsification scheme that removes edges with weights under a certain threshold, was tested for this clustering alternative due to its reliance on spatial attributes of the graph. Even with this modification the results were unsatisfying.

The Marsili-Giada method (Figure 5) provided the most granular clustering, uncovering subtle shifts in market behavior with fine temporal resolutions. The high level of detail observed in its results highlights its suitability for capturing microstructural patterns, such as short-lived volatility or localized trading strategies.

4 Results

This section presents the clustering results for different time aggregation windows, namely 1-hour, 30-minute, 15-minute, and 5-minute intervals. The aim is to investigate how the temporal granularity impacts the clustering of timestamps and the insights derived from these clusters. The results are presented through visualizations of the identified clusters, their timestamp distributions, and associated metrics. The analysis highlights key characteristics of each time aggregation and lays the groundwork for a comparative discussion in the subsequent section.

4.1 Clusters with 1-Hour Time Aggregation

Figure 6 illustrates the clusters identified by the Louvain algorithm for the 1-hour time window. Each cluster represents a distinct community of timestamps, color-coded and separated by their correlation-based similarity.

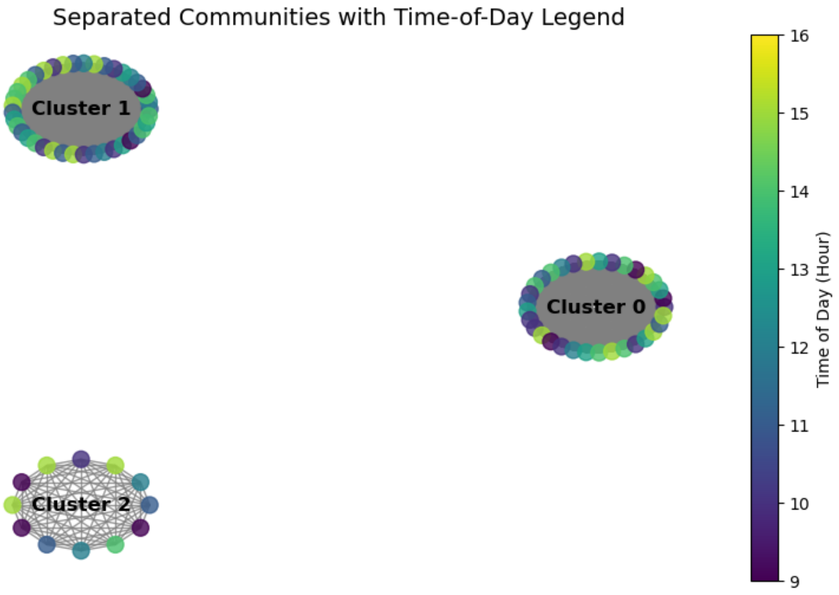

Figure 6: Louvain Clustering for 1-hour time window

The plot reveals three distinct clusters: Cluster 0, Cluster 1, and Cluster 2, representing periods of homogeneous market behavior. Cluster 2 correspond to specific, isolated periods, while Cluster 0 and cluster 1 span a broader range of timestamps, capturing more generalized market conditions. The separation between clusters highlights the Louvain algorithm's ability to identify sharp transitions in market states. However, this level it is not yet clear how the financial day can be divided into behavioral periods at this level of granularity.

Figure 7 displays the distribution of timestamps across the identified clusters.

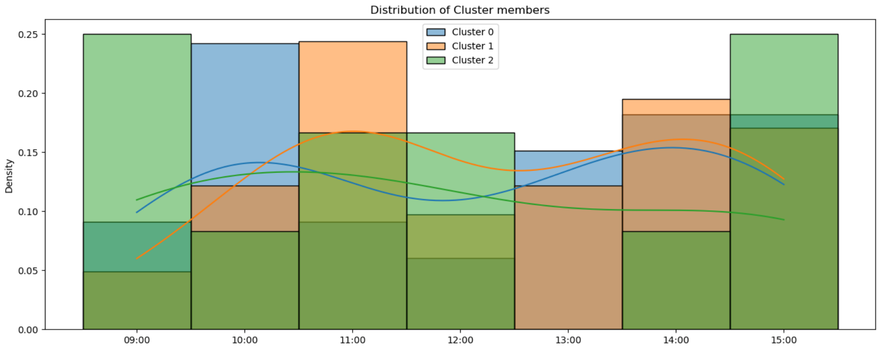

Figure 7: Timestamp distribution over clusters for 1-hour time window

The timestamp distribution shows that Cluster 0 captures most of the market activity, especially during the morning hours, while Clusters 1 and 2 are concentrated around specific intervals, such as midday and late afternoon.

Figure 8 presents cluster metrics, including time homogeneity (variance) and temporal modularity (mean time difference).

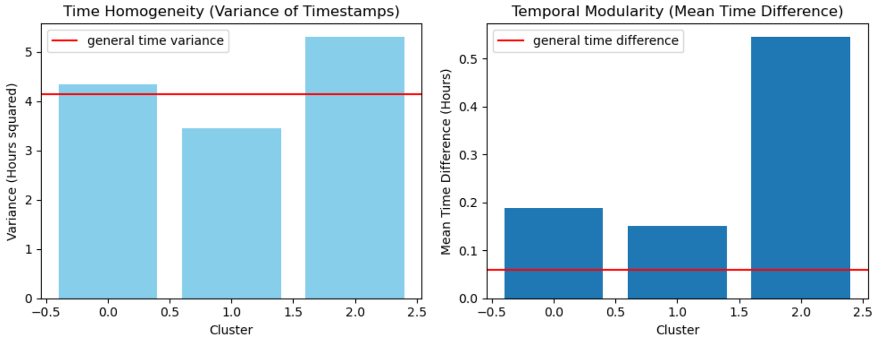

Figure 8: Cluster metrics for 1-hour time window

Cluster 2 exhibits the highest time variance, indicating greater variability and potentially more volatile market conditions during these periods. In contrast, Cluster 0 shows lower time variance, reflecting more cohesive market behavior. Temporal modularity further highlights the distinctiveness of Cluster 2, which has the highest mean time difference, separating it clearly from the other clusters.

4.2 Clusters with 30-Minute Time Aggregation

Figure 3 shows the identified clusters for the 30-minute time window. Each cluster represents distinct periods of market behavior. The spatial separation of clusters indicates clear temporal divisions within the trading day.

The clustering identified again three distinct communities within the temporal network, each characterized by unique patterns of connectivity and temporal distribution. Cluster 2 encompasses nodes spanning a wide range of time-of-day values (9:00–16:00) with a relatively mixed temporal distribution, suggesting diverse temporal behavior within this group. Cluster 1 also spans the full time range but exhibits a denser and more cohesive structure, indicating stronger intra-cluster connections and a higher degree of temporal and spatial correlation. In contrast, Cluster 0 is separated from the other two groups. Its nodes display a more structured temporal distribution toward the afternoon, highlighting distinct and isolated characteristics that may reflect specialized processes or activities.

Figure 9 illustrates the timestamp distribution across clusters, showing the temporal activity patterns.

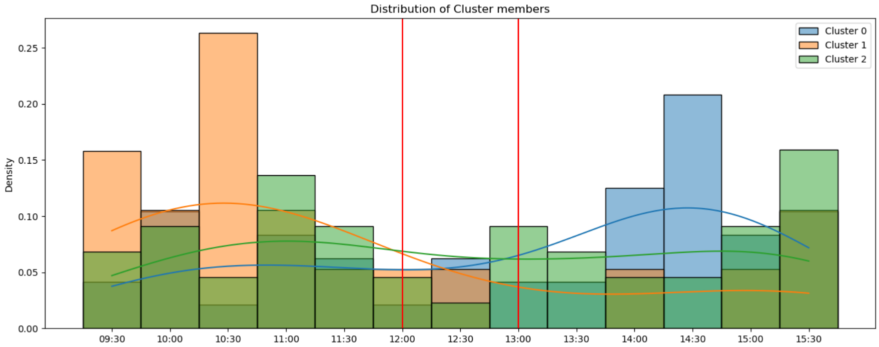

Figure 9: Timestamp distribution over clusters for 30-minute time window

Cluster 0 dominates in the late afternoon, corresponding to periods of consistent market activity. Cluster 1 is concentrated in the morning hours, reflecting potential volatility during market opening. Cluster 2 appears throughout the day, representing transitional or less active states. The overlap between clusters suggests temporal shifts in market conditions.

Figure 10 presents the cluster metrics, providing insights into their characteristics.

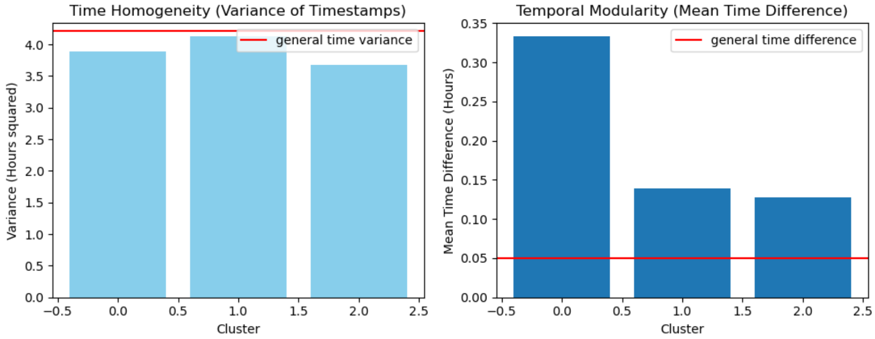

Figure 10: Cluster metrics for 30-minute time window

The variance metric indicates that Cluster 0 and Cluster 2 are more temporally homogeneous, with lower intra-cluster variability. Cluster 1 exhibits higher variance, suggesting more diverse and volatile market behavior. Temporal modularity highlights the distinctiveness of each cluster, with Cluster 0 showing the most separation, indicating strong internal cohesion and clear boundaries from other clusters.

4.3 Clusters with 15-Minute Time Aggregation

Figure 11 illustrates the clustering results for the 15-minute time window, revealing three distinct clusters. Clusters 0 and 2 appear dense, whereas Cluster 1 is comparatively sparser. The finer temporal resolution of 15 minutes makes it challenging to interpret the dominant temporal states solely from the clustering plot, as there is significant overlap among the clusters. This is where Figure 12 proves useful, providing a clearer visualization of the timestamp distribution across the clusters.

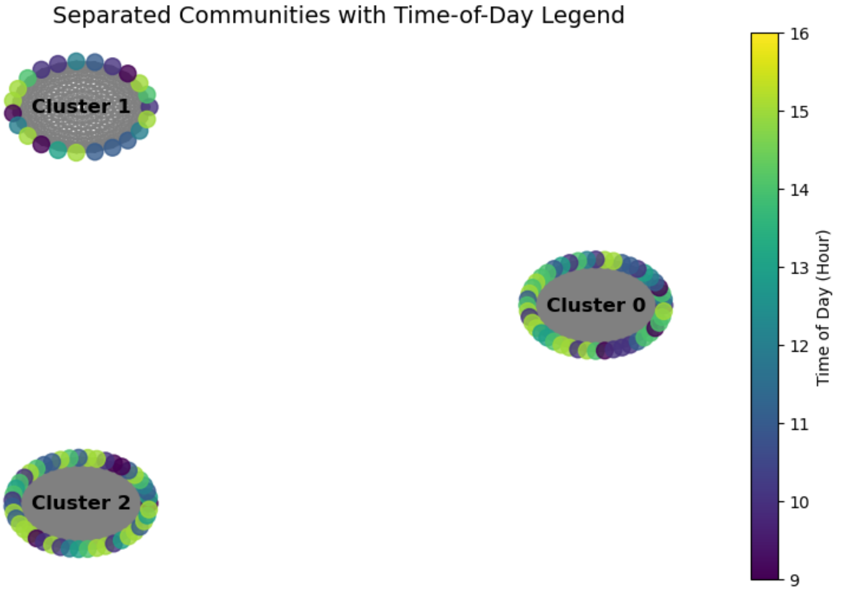

Figure 11: Louvain Clustering for 15-min time window

Figure 12 illustrates the timestamp distribution across clusters. Cluster 0 is prominent during the late afternoon, corresponding to periods of stable market activity as trading draws to a close. Cluster 1 spans the morning, potentially indicating periods of higher activity or volatility. Cluster 2 appears sporadically throughout the trading day, reflecting less cohesive and more dynamic market states. The overlap between clusters suggests temporal shifts in market conditions, emphasizing the ability of the clustering method to capture both stable and transitional behaviors.

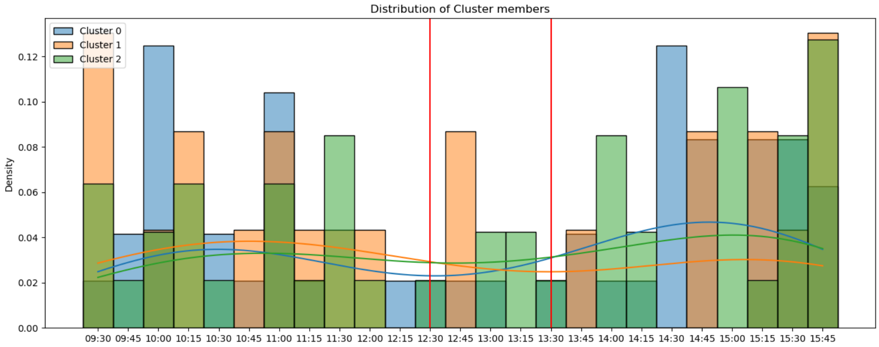

Figure 12: Timestamp distribution over clusters for 15-minute time window

Figure 13 presents cluster metrics, providing quantitative insights into their characteristics. Time homogeneity, measured by variance, shows that Cluster 2 exhibits the lowest variance, indicating consistent behavior within this period. In contrast, Clusters 0 and 1 demonstrate higher variance, reflecting more diverse and potentially volatile market conditions. Temporal modularity highlights the distinctiveness of each cluster. Cluster 0 achieves the highest modularity value, suggesting strong separation and internal cohesion. Clusters 1 and 2, with lower modularity values, indicate less distinct separation and more overlap with general market patterns.

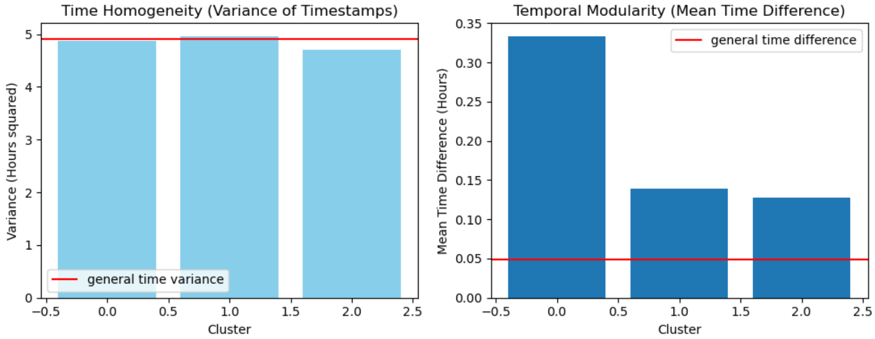

Figure 13: Cluster metrics for 15-minute time window

4.4 Clusters with 5-Minute Time Aggregation

Figure 14 shows the distribution of timestamps across clusters for the 5-minute time aggregation window. The higher temporal resolution reveals more granular details about market behavior, allowing for the detection of finer transitions between states.

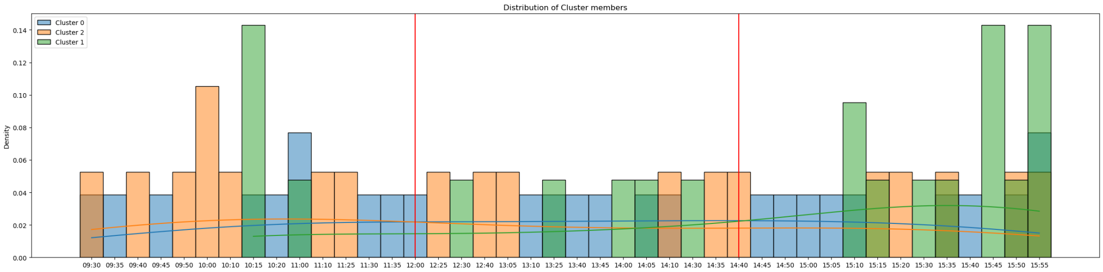

Figure 14: Cluster metrics for 5-minute time window

The clustering highlights three clusters: Cluster 0, Cluster 1, and Cluster 2. Compared to the longer aggregation windows (15-minute, 30-minute, and 1-hour), the timestamp distribution shows increased variability and less uniformity across clusters. This variability suggests that the shorter aggregation window captures brief, fleeting market behaviors that are less evident at larger scales. Clusters appear more evenly distributed over the trading day, with overlapping activity indicating frequent transitions between states.

However, the reliability of the covariance matrix becomes a concern at this level of granularity due to the greater number of timestamps compared to the number of stocks. This imbalance makes the covariance matrix less stable and, consequently, the clustering results less robust.

5 Discussion

The comparison of clustering results across different time aggregation windows reveals key differences in the ability to capture market dynamics. Increasing temporal granularity generally provides finer insights into short-lived behaviors but introduces challenges in stability and interpretability.

The 1-hour aggregation offers a broad perspective, effectively highlighting general trends in market states with well-defined clusters. This aggregation level captures prolonged periods of stability (e.g., Cluster 2) and distinct transitions (e.g., Clusters 0 and 1). However, it lacks the granularity needed to uncover more transient market behaviors that occur within shorter intervals.

With the 30-minute aggregation, more transitional periods and volatility patterns emerge, offering a greater degree of temporal resolution. The clusters are distinct yet compact, effectively balancing interpretability with the ability to capture shorter-lived states. This resolution introduces some overlap between clusters, reflecting the inherent complexity of intraday market behavior.

The 15-minute aggregation pushes the temporal granularity further, capturing fleeting market behaviors and finer transitions between states. While the increased resolution enhances the level of detail, it also increases overlap between clusters, reflecting more frequent state changes. This can blur the boundaries between clusters and makes interpretation more nuanced compared to higher aggregation levels.

At the finest resolution, the 5-minute aggregation, the clustering results highlight highly transient behaviors. However, the imbalance between the number of timestamps and stocks leads to an unstable covariance matrix, reducing the robustness and reliability of the clustering. While the temporal granularity is appealing, the results at this level are less meaningful due to increased noise and reduced stability.

It is noteworthy that the results correspond well with expectations for financial markets. Intraday market dynamics are often characterized by three primary states: morning activity, midday with reduced trading activity as people take breaks, and a more active afternoon period leading into the market close. This observation aligns with the clustering patterns seen in the 15-minute and 30-minute aggregations, where these distinct states are well captured.

In summary, the choice of aggregation level reflects a trade-off between granularity and robustness. Coarser resolutions (1-hour and 30-minute) provide stable, interpretable clusters suited for identifying overarching market trends, while finer resolutions (15-minute and 5-minute) excel at capturing short-lived transitions at the cost of increased noise and diminished interpretability. This comparison underscores the importance of selecting an appropriate time window to align with the analytical objectives and the constraints of the data.

To further deepen the findings of this project, a Markovian approach was explored as a stepping stone for prediction. The transition matrix Table 2 derived from the clustering analysis provides a probabilistic understanding of how the market evolves between regimes—morning, mid-day, and end-of-day.

Period	**Morning**	**Mid-Day**	**End-of-Day**
Morning	0.326	0.217	0.457
Mid-Day	0.458	0.375	0.167
End-of-Day	0.426	0.106	0.468

Table 2: Transition matrix showing the probabilities of moving between market periods.

Stationary Distribution:

$$\pi = [0.393, 0.205, 0.402]$$

The stationary distribution represents the long-term probabilities of being in each period, where the market spends 39.3% of the time in the morning, 20.5% in mid-day, and 40.2% in the end-of-day.

The diagonal elements (e.g., 0.326 for morning, 0.375 for mid-day, and 0.468 for end-of-day) indicate the likelihood of remaining in the same regime. The relatively high persistence of the end-of-day period (46.8%) suggests that market activity stabilizes during this period. Conversely, the mid-day period has a lower probability of persistence (37.5%), reflecting its transient nature and tendency to transition into either morning (45.8%) or end-of-day (16.7%).

This probabilistic framework has significant implications for market behavior prediction. By using the transition probabilities, one can forecast the likelihood of shifting from one regime to another, providing valuable insights for intraday trading strategies. For instance, a high probability of transitioning from morning to end-of-day (45.7%) suggests a potential increase in volatility or trading activity in the latter period

6 Conclusion

This project aimed to explore intraday clustering of financial data to detect temporal market states and analyze their underlying structure. By applying clustering techniques, particularly the Louvain method, to high-frequency data from the ETF S&P 500 during the month of May, we successfully identified distinct market states and evaluated the impact of temporal granularity on clustering results.

The results demonstrate that financial markets exhibit predictable intraday patterns, aligning with the expected three primary states: morning activity, midday consolidation, and afternoon activity leading into the market close. The 1-hour and 30-minute aggregations captured these states effectively, balancing interpretability and robustness. These resolutions highlighted overarching market trends and transitions, providing meaningful insights into intraday dynamics.

As temporal granularity increased to 15-minute and 5-minute intervals, the clusters revealed more transient and localized market behaviors. While these finer resolutions enhanced the detection of short-lived market states, they introduced challenges such as increased noise, diminished interpretability, and instability in the covariance matrix. The 5-minute aggregation, in particular, highlighted the limitations of working with high-dimensional datasets, where the number of timestamps exceeds the number of stocks.

A key insight from this study is the trade-off between granularity and reliability. Coarser time windows, such as 1-hour and 30-minute, are better suited for identifying stable, interpretable clusters that align with macro-level market dynamics. In contrast, finer resolutions are more appropriate for capturing microstructural patterns but require additional techniques to address the noise and instability inherent to high-frequency financial data.

Future work could focus on enhancing the reliability of clustering results at finer temporal resolutions by incorporating additional cleaning techniques, such as alternative noise-reduction methods or sparse matrix representations. Additionally, exploring hybrid approaches that combine multiple time windows could provide a more comprehensive view of market dynamics across different temporal scales.

Overall, this project demonstrates the utility of clustering algorithms, particularly the Louvain method, in uncovering meaningful patterns in financial markets. These insights have potential applications in algorithmic trading, risk management, and portfolio optimization, offering valuable tools for navigating the complexities of high-frequency trading environments.

References

[1] Dieter Hendricks, T Gebbie, and D Wilcox. “Detecting intraday financial market states using temporal clustering”. In: *arXiv preprint arXiv:1508.04900* (2017). URL: https://arxiv.org/abs/1508.04900.

[2] Pierre Miasnikof et al. *A Statistical Density-Based Analysis of Graph Clustering Algorithm Performance*. 2020. arXiv: 1906.02366 [cs.SI]. URL: https://arxiv.org/abs/1906.02366.