Network Modeling Of NYC Traffic Patterns

Visual Methods for Interpreting Complex Graphs

Sean Fitch
Data Analytics CSCI-4600

Abstract

Network visualization present significant challenges due to their multi-dimensional and intricate nature. This study develops a novel methodology for creating visually interpretable graph representations of transportation networks, using the Manhattan taxi zone network as a comprehensive case study. By integrating advanced algorithmic techniques, including the Fruchterman-Reingold layout algorithm, Louvain community detection, and HDBSCAN clustering, the research proposes a systematic approach to network visualization that preserves critical structural characteristics of complex graphs.

The methodology focuses on aligning graph modularity with spatial clustering, utilizing the Fowlkes-Mallows Index to evaluate the correspondence between edge-weight-based and spatial clustering techniques. Analyzing over 63 million high-volume for-hire vehicle trips in Manhattan during 2022, the study demonstrates a data-driven approach to network representation that reveals underlying graph patterns. The method addresses the fundamental challenge of transforming raw network data into an interpretable visualization that maintains the graph's intrinsic structural properties.

Results highlight the methodology's potential and limitations, showing improved cluster separation through optimization techniques while acknowledging the method's parameter sensitivity. The approach provides a framework for understanding complex urban transportation networks, offering insights into spatial and structural relationships that traditional visualization methods often obscure. Future work is proposed to refine the methodology, including parameter optimization and generalization to larger and more diverse datasets.

Introduction

Complex networks, such as urban transportation systems, present significant challenges in data visualization due to their intricate and multi-dimensional nature. The New York City taxi zone network serves as an exemplary case study for this project, featuring over 63 million trips between Manhattan zones in 2022, with numerous interconnected relationships that resist simple visual representation (NYC Taxi and Limousine Commission).

As a class project, this paper aims to address the fundamental problem of creating network visualizations that provide meaningful insights into graph structures. Urban transportation networks, with their dense interconnections, offer an ideal test case for developing advanced visualization techniques. The core challenge lies in developing a methodology that can transform raw network data into an interpretable graph that preserves the underlying network's structural characteristics.

The project explores multiple visualization techniques to develop a systematic approach that aligns graph layout with community detection methods. By investigating how different algorithmic techniques can be combined, the paper focuses on the methodological challenge of producing visually interpretable representations of complex networks.

The proposed approach combines the Fruchterman-Reingold layout (Fruchterman and Reingold), Louvain clustering (Blondel et al.), HDBSCAN (Campello et al.), and Fowlkes-Mallows Index (Fowlkes and Mallows) to create a framework for network visualization. The goal is to develop a methodology for generating network visualizations that effectively communicate the underlying graph's structural characteristics, using the NYC transportation network as a sophisticated test case for complex network visualization.

Data Description

The dataset utilized in this project originates from the New York City High-Volume For-Hire Vehicle (FHV) Trip Records, as described by the NYC Open Data portal (NYC Taxi and Limousine Commission):

> These records are generated from the trip record submissions made by High Volume For-Hire Vehicle (FHV) bases. On August 14, 2018, Mayor de Blasio signed Local Law 149 of 2018, creating a new license category for TLC-licensed FHV businesses that currently dispatch or plan to dispatch more than 10,000 FHV trips in New York City per day under a single brand, trade, or operating name, referred to as High-Volume For-Hire Services (HVFHS). This law went into effect on Feb 1, 2019. Each row represents a single trip in a FHV dispatched by a high volume base. The trip records include fields

capturing the high volume license number, the pickup and drop-off date, time, and taxi zone location ID, which correspond with the NYC Taxi Zones open dataset.

The data processing involved several critical steps to transform the original dataset into a manageable and focused dataset for an initial network visualization. The original dataset encompassed 60 GB of zipped files covering all taxi trips in NYC from 2011 to 2023. Due to the enormous size of the raw data—with the 2023 FHV dataset alone approaching 4000 GB as a CSV—extensive preprocessing was required.

Initial attempts to download the data proved futile due to the size of files. A script was created to process data in 100k-1M row batches from the NYC Open Data API, however, after reaching ~40M rows processed, the queries began constantly timing out. The data used here was therefore loaded from a copy of the dataset available on Kaggle (Mohanasundaram). A custom notebook was created to load the data, process it, and produce a compressed file to download.

Key preprocessing steps included first filtering to High-Volume FHV trips from 2022. Then, the resultant 212 million rows were filtered to those where the pickup and dropoff were in Manhattan, resulting in a final dataset of 63,110,170 trips. Columns were further reduced by combining fee-related columns into a single fee column (excluding tips) and removing High-Volume For-Hire Service (HVFHS) metadata, since the network visualization relates solely to passenger behavior. The processed dataset occupies 3.8 GB as a pandas DataFrame or 1.37 GB in parquet format.

The dataset includes the following columns:

- request_datetime (datetime) - date/time when passenger requested to be picked up
- pickup_datetime (datetime) - The date and time of the trip pick-up
- dropoff_datetime (datetime) - The date and time of the trip drop-off
- PULocationID (integer): Pickup Location ID - TLC Taxi Zone in which the trip began
- DOLocationID (integer): Dropoff Location ID - TLC Taxi Zone in which the trip ended
- trip_miles (float) - total miles for passenger trip
- trip_time (integer) - total time in seconds for passenger trip
- fee (float) - Sum of base fare, tolls, bcf, sales tax, congestion surcharge, and airport fee
- hvfhs (categorical) - One of Juno, Uber, Via, and Lyft

PULocationID and DOLocationID are integers corresponding to the NYC Taxi Zones open dataset (NYC Taxi and Limousine Commission), which are geographic regions of NYC used to anonymize taxi data. HVFHS corresponds to the dispatching company's name.

This significant data reduction and preprocessing enabled the creation of a manageable dataset suitable for network visualization analysis.

Preliminary Analysis

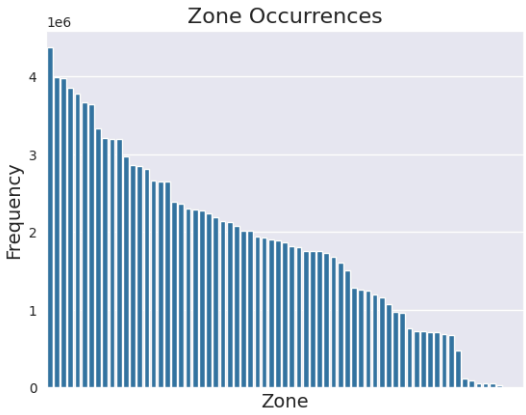


Fig. 1: Bar Plot of Zone Occurrences

Initial data exploration involved comprehensive visualization and statistical analysis of the NYC High-Volume For-Hire Vehicle trip dataset from 2022. We first examined the spatial distribution of trips by analyzing zone occurrences, which revealed significant variation in trip frequencies across different New York City zones (Fig. 1). The zone occurrence analysis showed a wide range of trip frequencies, from zero trips in inaccessible areas like Governor's Island to 4.37 million trips in high-traffic zones such as the East Village. When plotted, these occurrences demonstrated an approximately linear distribution, highlighting the substantial variability in transportation patterns across different city regions.

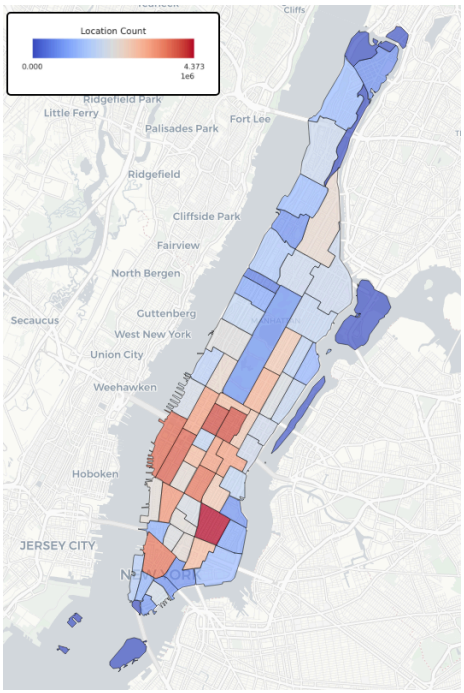
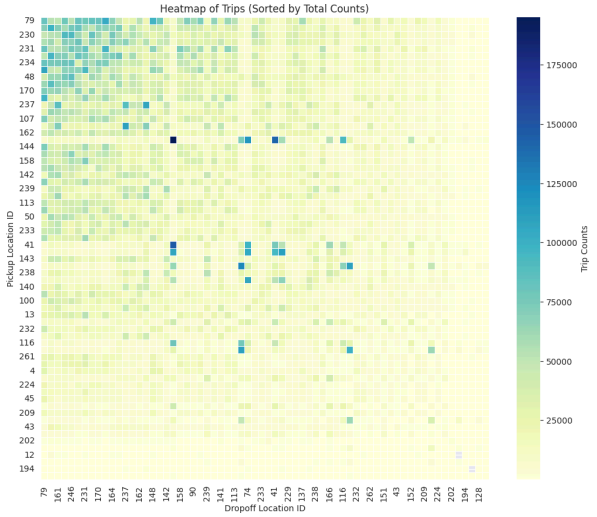


Fig. 2: Heatmap of Zone Occurrences by Taxi Zone

Geographical visualization of zone occurrences using the NYC Taxi Zones dataset (Fig. 2) provided further insights into the spatial dynamics of transportation. By mapping the trip frequencies, we could identify high-traffic corridors and low-activity areas. The results are unsurprising, with midtown and downtown areas having much higher occurrences.

Pickup-dropoff pair frequency analyses (Fig. 3a, b) revealed a high-order decline in trip pair frequencies. This steep drop-off in inter-zone trip frequencies was a primary motivation for focusing subsequent analyses exclusively on Manhattan trips, as the data for other boroughs exhibited less coherent patterns and would certainly require scaling methods such as log or exponential.



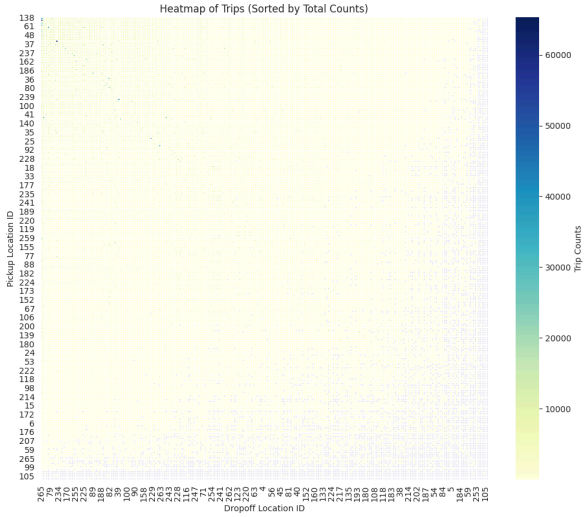


Fig. 3a: Heatmap of pickup-dropoff pairs for Manhattan Fig. 3b: Heatmap of pickup-dropoff pairs for all boroughs

Temporal distribution analysis (Fig 4) uncovered several significant non-uniform patterns across different time scales. The second-level distribution demonstrated extreme deviation from uniformity, with a chi-squared test yielding a p-value of 0.0 and a test statistic of 440,000. Even after excluding the peak at the zeroth second, the distribution remained statistically distinct from a uniform distribution, with a chi-squared test revealing a p-value of 1.6e-189 and a test statistic of 1,086.

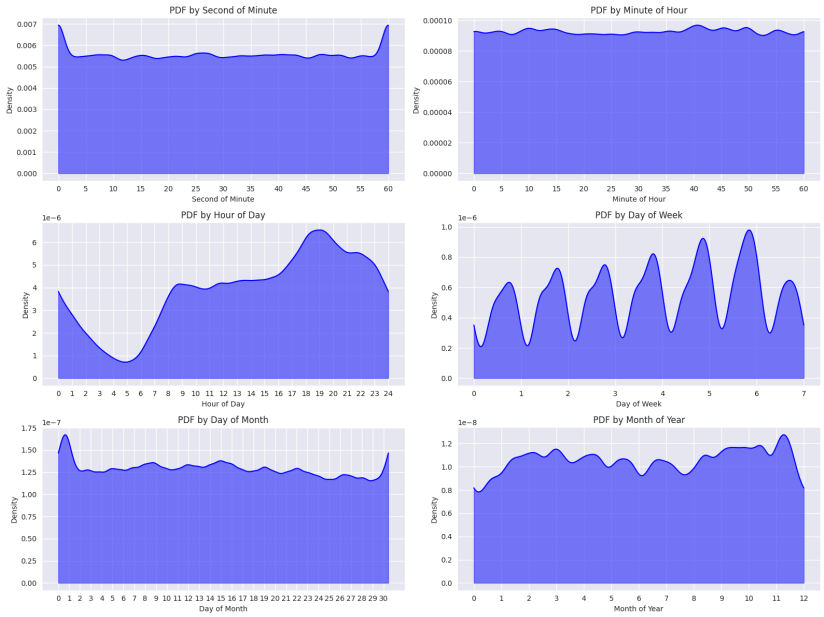


Fig. 4: Gaussian KDE of trip request time across multiple time scales (Circularized to account for edge effects)

Hour-level time distributions exhibited similar non-uniformity. Minute-of-hour frequencies showed statistically significant deviations from a uniform distribution, with chi-squared tests resulting in a test statistic of 52,000 and p-value of 0.0. Grouping minutes into five-minute intervals produced a test statistic of 32,000 with a p-value of 0.0, indicating there are significant patterns over a 5 minute time scale.

Broader temporal analyses revealed nuanced trip patterns. Daily trip frequencies showed distinct variations, with a notable dip between 1 and 7 AM, stabilization from 8 AM to 4 PM, and a peak at 7 PM. Weekly patterns mirrored these sub-daily trends, with increasing trip frequencies from Saturday to Friday and a subsequent drop on Saturday. Monthly distributions indicated a higher frequency of trips on the first of each month, with subsequent peaks occurring approximately weekly. Yearly distributions are irregular, but indicate a drop in trips in January, with the highest frequency being early December.

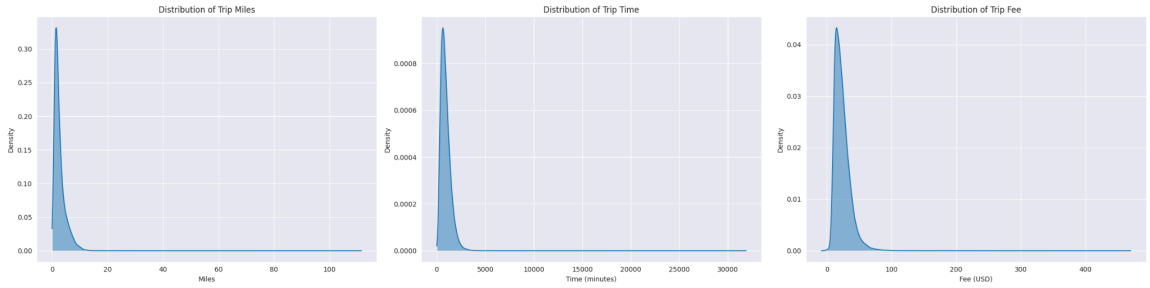


Fig. 5a: Gaussian KDE of Trip Miles, Time, and Fee

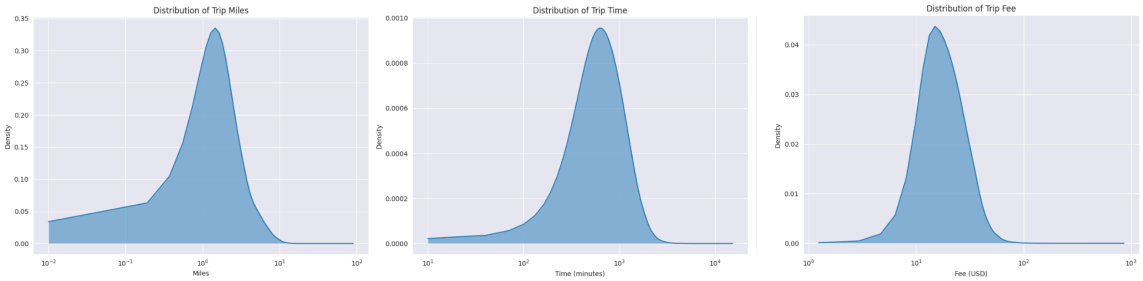


Fig. 5b: Gaussian KDE of log of Trip Miles, Time, and Fee

Variable distributions for trip miles, trip time, and trip fees (Fig. 5a) demonstrated remarkable similarities, suggesting strong correlations between these metrics. They all displayed characteristics of a log-normal distribution. Interestingly, both base passenger fare and driver pay exhibited occasional negative values, warranting further investigation. A logarithmic plot of fees further emphasized the interrelationship between these variables (Fig 5b) (The fee plot starts at 1 to exclude small and negative values).

These preliminary analyses provided critical insights into the complex dynamics of New York City's high-volume for-hire vehicle transportation network, setting the stage for more advanced network modeling and visualization techniques.

Model

The primary objective of this methodology is to create a network layout that visually represents the internal weight structure of the graph. This involves pursuing layouts with two key characteristics: the graph should exhibit strong, visually separable features, and these features should respect the underlying graph characteristics. For an overview of earlier attempts that informed this approach, refer to the appendix.

To achieve this, the methodology compares two types of clustering: one derived from the graph's edge weights and the other based on the visual properties of the network layout. A high-quality network visualization incorporating edge weights should ideally reflect the community structure identified by edge-weight-based clustering, as both rely on the graph's modularity and edge relationships. In a meaningful visualization, spatial clusters in the layout should roughly correspond to the edge-weight-based clusters.

To represent the internal edge structure, the Louvain clustering algorithm was selected. This is a modularity-based community detection algorithm that partitions nodes into clusters with dense intra-cluster edges and sparse inter-cluster edges. For representing the visual properties of the network, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) was used. This density-based clustering algorithm identifies groups of data points based purely on spatial proximity, making it robust to clusters of arbitrary shapes and densities.

The alignment between these clustering methods was evaluated using the Fowlkes-Mallows Index (FMI), a cluster similarity metric. The FMI measures the quality of clustering results by comparing pairs of data points assigned to the same cluster in both clustering outputs. Scores range from 0 to 1, with a value of 1 indicating perfect alignment between clusters.

To generate the graph layout, we used the Fruchterman-Reingold algorithm, which is a graph layout algorithm used to position nodes in a graph in a visually appealing manner, by simulating nodes as charged particles and edges as springs.

The methodology begins with a weighted adjacency matrix and the desired number of clusters, *n*, as inputs. The steps for generating a network layout and evaluating its alignment with clustering are as follows:

1. **Louvain Clustering**
   Perform a grid search over the resolution parameter to identify a configuration of the Louvain algorithm that produces *n* clusters.
2. **Graph Layout Generation**
   Conduct a grid search over the *k* parameter for the Fruchterman-Reingold algorithm, which controls layout characteristics such as node spacing and force distribution. For each value of *k*, generate a layout to be evaluated.
3. **HDBSCAN Clustering**
   Apply the HDBSCAN algorithm to each layout generated by the Fruchterman-Reingold method.
4. **Cluster Similarity Evaluation**
   For each combination of *k* and layout:
   - Calculate the FMI to measure the alignment between Louvain and HDBSCAN clusters.
   - Track the layout and clustering configuration that yield the highest FMI score.
5. **Optimization and Output**
   Return the maximum FMI score along with the cluster assignments for both Louvain and HDBSCAN methods.

By integrating these steps, the workflow identifies and visualizes a network layout that best represents the internal weight structure while ensuring alignment between graph modularity and spatial clustering features.

Results

Here we compare two graph layouts (Fig. 6a,b), with the optimization method applied or not applied. Both layouts have the centers of nodes colored by identical Louvain clusters, and

exterior of nodes colored by HDBSCAN clusters. The cluster labels were lined up to produce the maximum possible overlap between the two clusterings using integer programming (Kalvelagen). If there is no outer ring, the node was classified as noise by HDBSCAN. Clearly, the layout without optimization produces no visually separable clusters. On the other hand, the layout with optimization produces distinct clusters, though a good number of points are still considered noise.

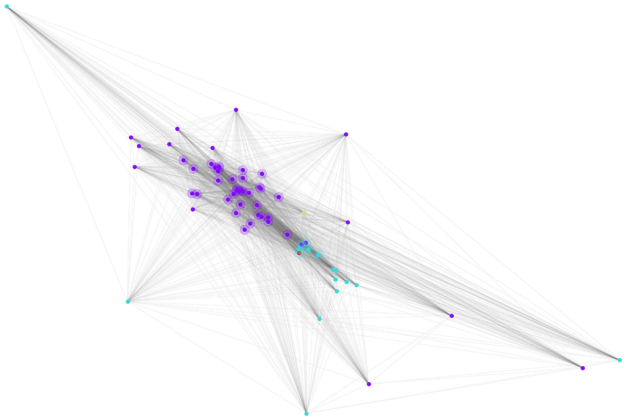


Fig. 6a: Graph layout without optimization

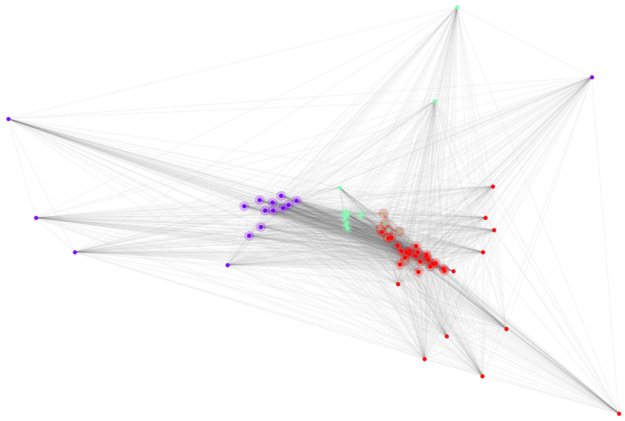


Fig. 6b: Graph layout with optimization

Additionally, here we have confusion matrices for the nodes (Fig. 7a,b). We can see the trace, representing the overlap between the two clustering methods, is much greater with optimization applied. Note that the 1 node classified as noise (-1) for both clusterings was Ellis Island, as this zone had 0 trips due to being an island. On the other hand, Louvain gives labels to all nodes and thus has no other nodes classified as noise.

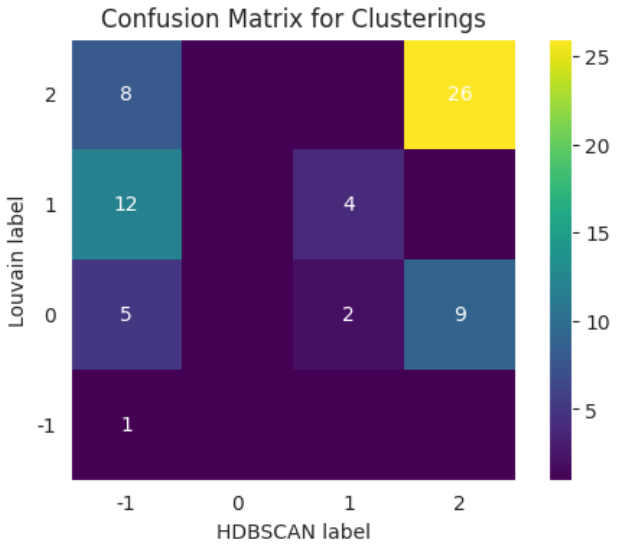


Fig. 7a: Confusion Matrix without optimization

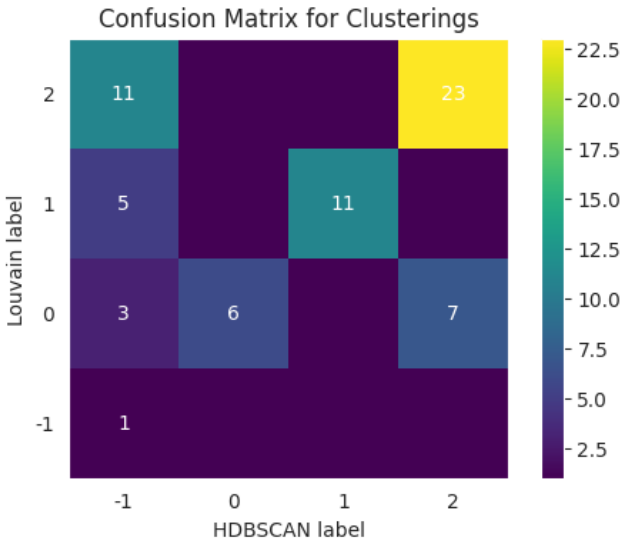


Fig. 7b: Confusion Matrix with optimization

Here we have a plot on the map of the labels applied to each zone, both with and without optimization (Fig. 8a,b). The fill color represents the Louvain cluster while the edge color represents the HDBSCAN cluster. The clusters seem to favor geospatially adjacent groups, as frequency of trips between zones is higher in adjacent zones. Both examples have difficulty separating the edge between midtown and downtown clusters, possibly due to a smooth change in edge characteristics from downtown to midtown.

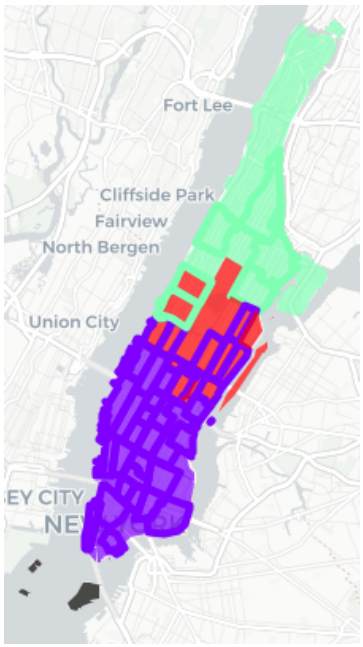


Fig. 8a: Map of clusterings without optimization

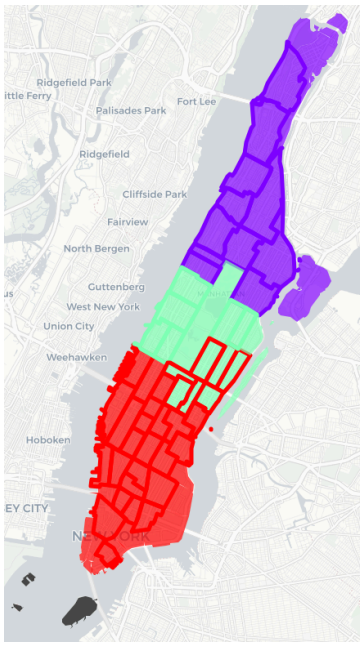


Fig. 8b: Map of clusterings with optimization

Limitations

Despite the success of the methodology in the example graph of the Analysis section, many limitations remain. It is still very parameter-sensitive, and clear patterns do not arise when tweaking individual parameters. This may create a computational barrier when applying the method to larger graphs.

For example, the methodology currently requires the number of clusters to be specified, and on this graph, good results were only produced with $n = 3$. With two clusters, the method failed to find a valid Louvain clustering. Four clusters had poor performance (Fig. 9a,b), and with five plus clusters, the method failed to find layouts that had that many distinct clusters.

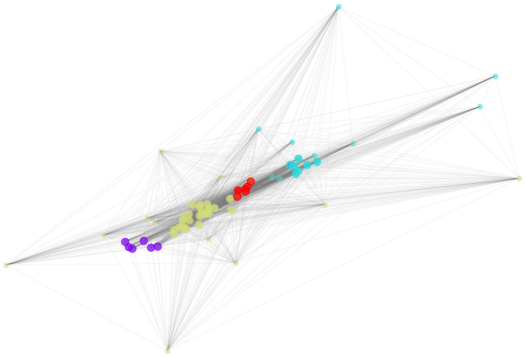


Fig. 9a: Network visualization with $n = 4$

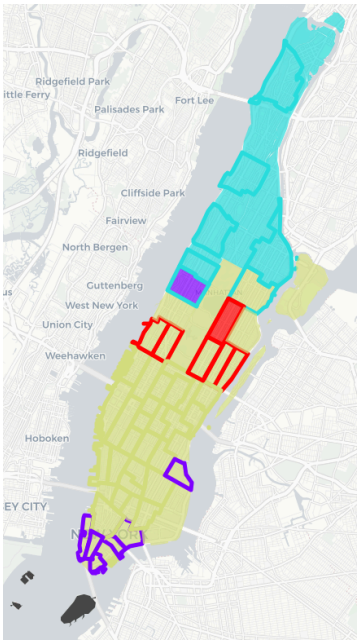


Fig. 9b: clustering results with $n = 4$

Future Work

Future work should prioritize optimizing model parameters by refining heuristics and adopting advanced search methods beyond grid search. This includes determining the optimal number of clusters and Louvain parameters, optimizing HDBSCAN parameters with alternative search algorithms, and reconsidering metrics like silhouette scores for Louvain clustering to minimize the added complexity of the HDBSCAN step. Generalization testing should involve applying the model to diverse and larger datasets, such as the entire NYC taxi dataset. Additionally, further research should focus on defining a robust mathematical representation for graph visualizations that faithfully preserve graph characteristics, as this study only uses Louvain clustering for initial characterization of edge properties.

The original research question was “where should New York City focus construction efforts to improve traffic flow?”. This would be answered by modelling the ratio of travel time to distance across roads, hypothesizing that locations where the ratio was high after accounting for differences in land and road use could benefit from redesign or improvement. Results could be validated by comparing the models’ results with data from NYC road maintenance permits. Additionally, with more experimental design, this data could be incorporated into the initial model by examining places where traffic flow was shown to improve due to road work.

Unfortunately, this research question is far beyond the scope of a project such as this, and therefore the project shifted to a much more manageable problem which was initially planned to be part of the data exploration stage.

Works Cited

Blondel, Vincent D., et al. “Fast unfolding of communities in large networks.” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, 2008, p. 10008, https://dx.doi.org/10.1088/1742-5468/2008/10/P10008. Accessed 10 Dec 2024.

Campello, Ricardo J. G. B., et al. “Density-Based Clustering Based on Hierarchical Density Estimates.” *Advances in Knowledge Discovery and Data Mining*, 2013, pp. 160-172, https://link.springer.com/chapter/10.1007/978-3-642-37456-2_14. Accessed 10 Dec 2024.

Fowlkes, E. B., and C. L. Mallows. “A Method for Comparing Two Hierarchical Clusterings.” *Journal of the American Statistical Association*, vol. 78, no. 383, 1983, pp. 553-569, https://www.tandfonline.com/doi/abs/10.1080/01621459.1983.10478008. Accessed 10 Dec 2024.

Fruchterman, Thomas M.J., and Edward M. Reingold. “Graph drawing by force-directed placement.” *Software: Practice and Experience*, vol. 21, no. 11, 1991, pp. 1129-1164, https://onlinelibrary.wiley.com/doi/10.1002/spe.4380211102. Accessed 10 Dec 2024.

Kalvelagen, Erwin. “Maximise diagonal of matrix by permuting columns in R.” stack overflow, 2020, https://stackoverflow.com/a/61565539. Accessed 10 Dec 2024.

Mohanasundaram, Sripathi. *Newyork city Taxi Trip Records Dataset*, Kaggle, 2023, https://www.kaggle.com/datasets/microize/nyc-taxi-dataset/data. Accessed 9 Dec 2024.

NYC Taxi and Limousine Commission. *2022 High Volume FHV Trip Records*, NYC Open Data, 2023, https://data.cityofnewyork.us/Transportation/2022-High-Volume-FHV-Trip-Records/g6pj-fsah. Accessed 9 Dec 2024.

NYC Taxi and Limousine Commission. *NYC Taxi Zones*, NYC Open Data, 2019, https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc. Accessed 10 Dec 2024.

Appendix: Model Development

To explore effective graph visualizations, a grid search (Fig. 9a) was conducted across various layout algorithms, scaling methods, and parameter combinations. Layout algorithms included Spring, Kamada-Kawai, Shell, and Fruchterman-Reingold, paired with scaling techniques such as Linear, Logarithmic, Square Root, Standard, and Min-Max normalization. Additional parameters like thresholding and k values were also tested. This was narrowed to Fruchterman-Reingold layouts (Fig. 9b), as they had the most promising results for producing distinct clusters. However, no combination produced visually separable clusters.

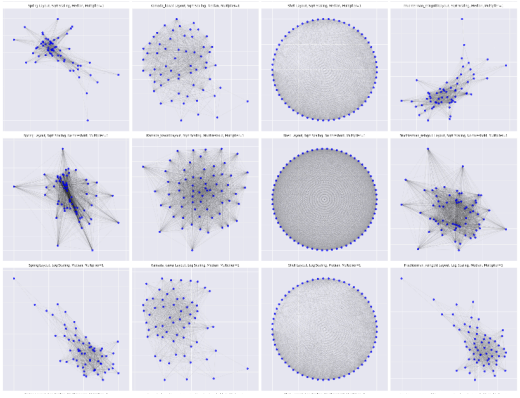


Fig. 9: Sample of Grid search results (Not Comprehensive)

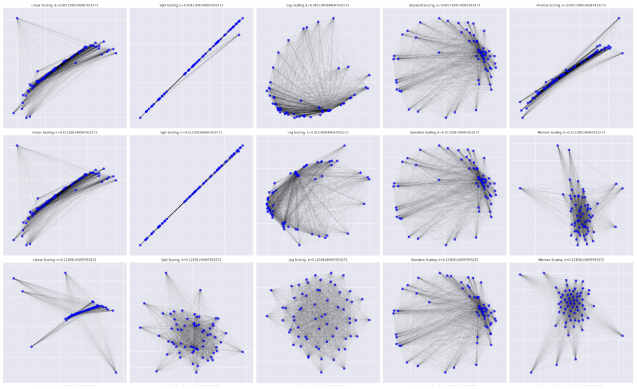


Fig. 9b: Fruchterman-Reingold Grid search results (Not Comprehensive)

Clustering algorithms such as Louvain, Girvan-Newman, Infomap, Label Propagation, and Spectral Clustering were applied for node coloring, again using grid search (Fig. 10). However, the clustering did not improve the underlying problem of node layout. It did however, narrow the scope to Louvain clustering, as it produced the most visually separable clusters.

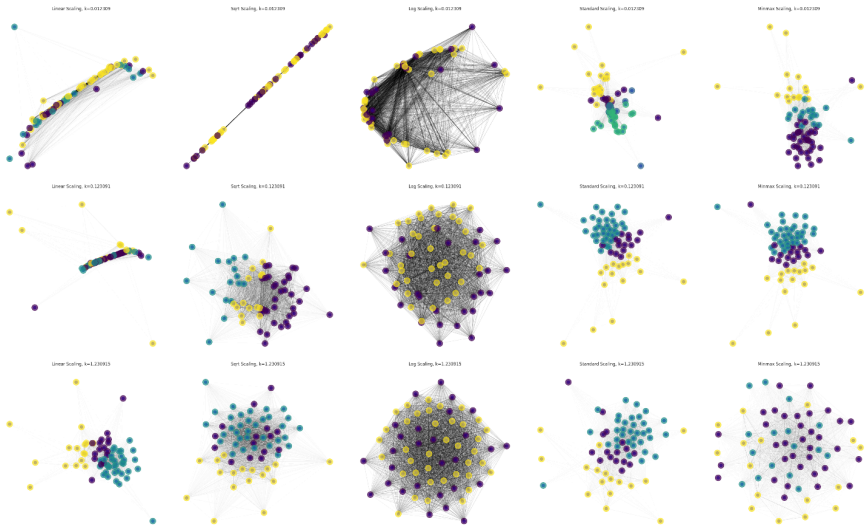


Fig. 10: Sample of Clustering Grid search results (Not Comprehensive)

Finally, a grid search on clustering and layout parameters was performed to maximize the silhouette score of Louvain clusters. However, it revealed several challenges. The process often resulted in imbalanced or very few clusters, and tuning the increasing number of parameters (a minimum number of clusters and a penalty for imbalanced clusters) proved difficult. Moreover, the scores relied far more on the initial clustering rather than the actual layout of the graph (see the horizontal stripes in Fig. 11), indicating that we were optimizing the clustering rather than the node layout. These challenges led us to take the approach described in the Model section.

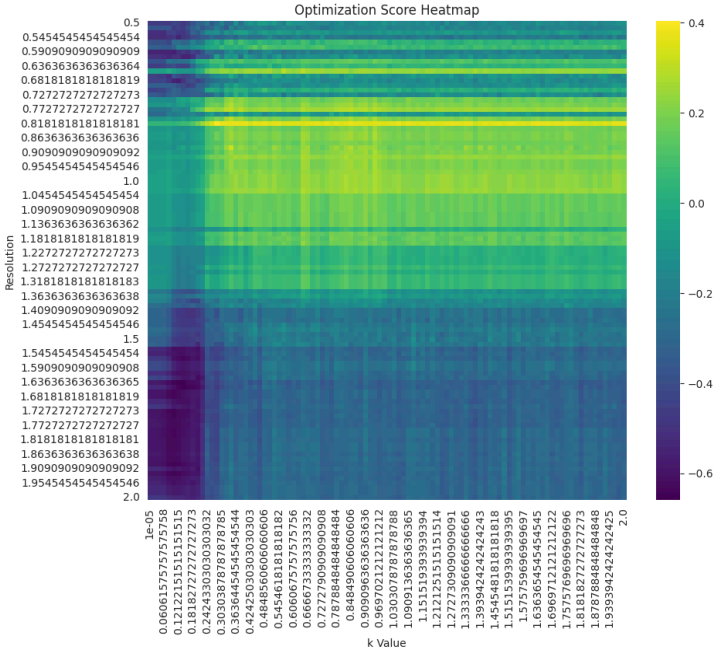


Fig. 11: Heatmap of optimization scores