



## Trajectory-based clustering for enhanced attractive region mining in urban taxi services

Muhammad Toqeer, Kifayat Ullah Khan & Waqas Nawaz

To cite this article: Muhammad Toqeer, Kifayat Ullah Khan & Waqas Nawaz (2024) Trajectory-based clustering for enhanced attractive region mining in urban taxi services, International Journal of Digital Earth, 17:1, 2356160, DOI: [10.1080/17538947.2024.2356160](https://doi.org/10.1080/17538947.2024.2356160)

To link to this article: <https://doi.org/10.1080/17538947.2024.2356160>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 22 May 2024.



Submit your article to this journal [↗](#)



Article views: 13




View related articles [↗](#)



View Crossmark data [↗](#)



# Trajectory-based clustering for enhanced attractive region mining in urban taxi services

Muhammad Toqeer<sup>a</sup>, Kifayat Ullah Khan<sup>b</sup> and Waqas Nawaz <sup>c</sup>

<sup>a</sup>Department of Artificial Intelligence and Data Science, National University of Computer and Emerging Sciences (NUCES-FAST), Islamabad, Pakistan; <sup>b</sup>College of Accounting, Finance, and Economics, Birmingham City Business School, Birmingham City University, Birmingham, UK; <sup>c</sup>Department of Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia

## ABSTRACT

Trajectory data, increasingly available due to location tracking technologies, holds immense potential for intelligent traffic management and urban planning. Traditional 'attractive region' mining methods often rely on density-based clustering, neglecting the inherent path information within trajectories. To address this, we propose a novel graph-based approach for attractive region discovery. By transforming trajectory data into graphs, we effectively leverage path and connectivity information for clustering with locality-sensitive hashing. Our study introduces the pARM, pgARM, and hgARM algorithms, demonstrating their superiority over GridDBScan through experiments on real-world datasets. We employ Davies–Bouldin metric and visualization techniques to highlight the robustness of our approach, especially for datasets with varied degree distributions. Although our method may have slightly longer processing times for smaller grid sizes, it achieves execution times comparable to GridDBScan for larger grids. We rigorously analyze performance variations within our algorithms using execution time, clustering coefficient, and modularity scores, providing guidance for their optimal application.

## ARTICLE HISTORY

Received 24 January 2024

Accepted 11 May 2024



## KEYWORDS


Intelligent transportation; urban planning; trajectory data mining; attractive region mining; locality-sensitive hashing; clustering

## 1. Introduction

Intelligent transportation systems use geospatial data to improve the travel experience of the public by identifying better routes with reduced congestion while improving safety (Chavhan et al. 2021; De Souza Allan et al. 2016; Olayode et al. 2023). Trajectory data mining (J. Liu et al. 2017; Zhang et al. 2023) plays a crucial role in such intelligent systems in understanding travel patterns that help predict traffic flow (M. Li et al. 2021), anticipate path-related problems (Z. Wang, Fu, and Ye 2018), and find attractive or hot regions (Nikitopoulos et al. 2018), which is our focus in this article. Travel patterns are standard phenomena caused by various aspects during travel and are essential for individual mobility and location-based route planning (H. Cai et al. 2016).

Attractive Region Mining (ARM) becomes nontrivial due to the computational complexity of the task and the lack of accurate and timely information, as emphasized by Hamdi et al. (2022) and Cheng et al. (2021). Most studies on detecting travel patterns are based on data from public

**CONTACT** Waqas Nawaz  [wnawaz@iu.edu.sa](mailto:wnawaz@iu.edu.sa)  Department of Information Systems, Islamic University of Madinah, Prince Naif Ibn Abdulaziz, 3087-Al Jamiah Dist., 42351, Madinah, Saudi Arabia

 Supplemental data for this article can be accessed online at <http://dx.doi.org/10.1080/17538947.2024.2356160>.

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

transportation (S. Li et al. 2019), studying traffic flow modeling in urban areas, and on the mobility of trucks on freeways (Olayode et al. 2023). However, with the development of location-aware technologies (Pachni-Tsitiridou and Fouskas 2019) and using Geocoding like Baidu API (J. Liu et al. 2017), the collection of travel data from residents is becoming more accessible, especially in urban areas where a taxi is the most common mode of transportation. Real-time analysis of large-scale taxi trips can provide valuable insights into traffic patterns and structures (Deng et al. 2020; W. Li et al. 2023; Paulsen, Rasmussen Thomas, and Nielsen Otto 2021; Y. Wang et al. 2020). Previous studies have analyzed spatial-temporal distribution from taxi trajectory data to identify attractive areas (Zheng et al. 2018), infer travel purposes (Hou et al. 2021), extract congestion areas (Fu et al. 2022), forecast traffic flow (Lan et al. 2022), and estimate travel time (Huang et al. 2022).

Grid-based approaches are prevalent to efficiently identify attractive regions with spatial and temporal constraints while mining trajectory data such as latitude and longitude information (G. Cai et al. 2014; Ohadi et al. 2020). As a state-of-the-art approach, the GridDBScan algorithm (Zheng et al. 2018) is a modified version of the density-based clustering algorithm that extracts travel patterns for space and time. This algorithm performs outlier-free density-based clustering for ARM using taxi trajectory data. However, the density-based clustering approach lacks edge connectivity between spatial data points, which results in inaccurate attractive areas. It does not focus on the natural phenomenon of paths, i.e. links or edges. This approach compares the means of each pair of grid cells. We understand that comparing means puts less emphasis on outlier points and leads to biased results with power-law distribution data (O’Hagan et al. 2016). In other words, the mean-based approach may not be practical when the distribution of data points is skewed.

To overcome the core issue of the existing work for the ARM problem (Zheng et al. 2018), we formulate a solution that is free from dependence on the mean-based approach because of its sensitivity to the underlying distribution of the data. For this purpose, we propose three variants to solve the problem. First, we opt for a pairwise approach, the pairwise attractive region mining algorithm (pARM), which computes the direct distance between pairs of grid points instead of the cell means. Our pARM approach provides accuracy almost similar to that of the mean-based approach at the expense of quadratic time complexity due to point-to-point checking and lack of travel patterns. Therefore, we propose a second variant called pairwise graph-based attractive region mining (pgARM) which performs pairwise distance checking among nodes in a spatial trajectory graph. Any spatial trajectory involves a starting and ending point, forming a graph. The vertices are the endpoints of a taxi travel/route, and the edges represent the paths between the endpoints. In this way, we compute the distances among the vertices for attractive-region mining, considering travel patterns inherently. Although modeling data in graphs has rich semantics, it still suffers from complexity issues. Thus, we propose a third algorithm, hashing-based graph attractive region mining (hgARM), to solve the problem of higher computational complexity. In this regard, we use Locality Sensitive Hashing (LSH), a fast similarity search algorithm, to provide speed-ups by avoiding unnecessary similarity computation among distant data points in hgARM. We conducted a thorough experimental analysis of the proposed algorithms using four publicly available real-world datasets to reveal their superiority over the other variants. We now summarize the key contributions of this article as follows.

- **Proposing a new solution to solve the problem of ARM:** We propose three variants of attractive region mining (ARM) algorithms in a successive order, where each algorithm is superior to its predecessor in accuracy and efficiency.
- **Use of semantic features in ARM:** Modeling spatial trajectory data in graph structure to consider travel patterns as semantic features in ARM.
- **Modelling LSH in the ARM domain:** We propose a strategy to improve the efficiency of pairwise distance computations by modeling a well-known approach called LSH in the context of ARM.

Since several acronyms are used in our paper, we provide their details in the table below.

Acronym	Description
ARM	Attractive Region Mining
pARM	Pairwise Attractive Region Mining
pgARM	Pairwise Graph-Based Attractive Region Mining
hgARM	Hashing and Graph-Based Attractive Region Mining
LSH	Locality Sensitive Hashing
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
GridDBScan	Grid and Density-Based Spatial Clustering of Applications with Noise
DPC	Density Peaks Clustering

## 2. Related work

In this section, we navigate through related concepts and ideas presented in the literature for attractive region and trajectory data mining.

Clustering and sequential pattern mining are commonly used techniques in trajectory data mining (Gaffney and Smyth 1999; Gao and Yu 2017; Guan, Liu, and Chen 2013; Kharrat et al. 2008; Lee, Han, and Whang 2007; Z. Li et al. 2010; D. Liu, Cheng, and Yang 2015; Mao, Ji, and Liu 2016; Qi and Liu 2018; Saptawati 2017; Takimoto, Sugiura, and Ishikawa 2017). Many researchers have utilized textual features for clustering trajectory data, with each point in the trajectory described and captured as text to use semantic trajectories (Takimoto, Sugiura, and Ishikawa 2017). The research on studying the freeway traffic flow to learn about volume and the congestion caused by the trucks used machine learning for this purpose (Olayode et al. 2023). Another notable approach is to use public cloud-based APIs from platforms like Baidu to perform trajectory pattern mining and trajectory clustering for the smart city (J. Liu et al. 2017). Similarly, Visualization is a crucial component of mining trajectory data, as traffic data has become a vital part of daily life due to the time people spend on the road (Y. Li and Ren 2022). Visualizing traffic data using taxi trajectory charts allows humans to interpret it naturally and investigate congested areas and traffic jams. Analyzing traffic flow patterns can provide valuable information and recommendations to taxi drivers and passengers (Tran, Leyman, and De Causmaecker 2022).

Feature extraction is non-trivial for individual trajectories and complex attributes (Chen et al. 2022; H. Li et al. 2022), making it challenging to determine representative routes or common trends shared by multiple moving objects. To address these challenges, various techniques have been introduced in the literature, such as using the regression mixture model and the expectation-maximization algorithm to categorize similar trajectories (Gaffney and Smyth 1999), changing the trajectory partition to line segments (Kharrat et al. 2008), and using the Trajectory-Hausdorff distance to build a group of closed trajectory segments. Furthermore, a micro- and macro-clustering framework was introduced by Lee, Han, and Whang (2007) and used by Z. Li et al. (2010). The authors in Guan, Liu, and Chen (2013) proposed a method to calculate the distance between two trajectories using a trajectory clustering algorithm based on the Hausdorff minimum distance, in which the relative and local distances are combined. Gao and Yu (2017) proposed a method for measuring the distance between two sub-trajectories using the DTW (dynamic time warping) algorithm.

Literature studies also focused on the identification of attractive regions (D. Liu, Cheng, and Yang 2015; Mao, Ji, and Liu 2016; Qi and Liu 2018; Saptawati 2017; Takimoto, Sugiura, and Ishikawa 2017; Zheng et al. 2018). A density-based approach (D. Liu, Cheng, and Yang 2015), density peaks clustering (DPC), uses the density peaks feature to identify hot zones. The authors combined the DPC algorithm with image analysis techniques to improve efficiency. DPC experiments revealed that it is capable of detecting long-term hotspots. In another study (Mao, Ji, and Liu 2016), the authors presented a method to group the origin and destination points for travel trajectories, specifically household travel patterns, by identifying appealing locations. This approach allows

the visualization of clusters based on spatial distribution and temporal direction. Another study (Takimoto, Sugiura, and Ishikawa 2017) analyzes semantic trajectories to identify patterns and regions of interest. They developed a clustering algorithm called SimDB-SCAN that clusters areas based on similarities and incorporates user preferences using Flickr photos. They combined the region of interest (ROI) with trajectory points to create efficient models, which differs from previous studies that primarily used numerical data to group trajectory points.

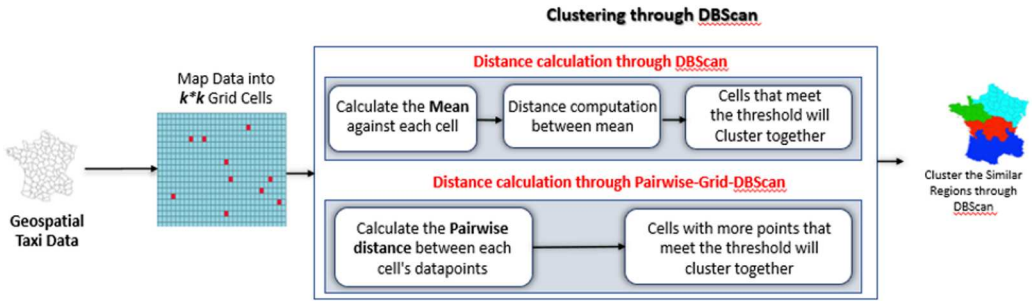
Various grid-based approaches have been introduced in the literature to identify exciting regions (Qi and Liu 2018; Saptawati 2017; Zheng et al. 2018). For example, the authors in (Saptawati 2017) divided the trajectory dataset into smaller grids before applying DBSCAN clustering to each grid for each timestamp, resulting in high congestion levels in some areas. The R-FDBSCAN (Qi and Liu 2018) approach (DBSCAN-based spatial clustering algorithm) has an additional parameter  $R$  controlling the size of the clusters. This algorithm has demonstrated numerous advantages about time performance and clustering results. GridDBScan (Zheng et al. 2018) uses density-based clustering to identify hot regions based on the distances between the mean points of each cell. We understand that their mean-based approach may not be practical when the distribution of data points is skewed. Moreover, none of the above methods consider the rich semantics of travel patterns, e.g. pickup and drop-off locations, to identify attractive regions.

### 3. Methodology

This section explains our proposed incremental strategy for the ARM problem. Initially, we suggest an alternative approach called pairwise attractive region mining (pARM), in contrast to the mean-based approach (Zheng et al. 2018), where the distance between each data point in adjacent grid cells is calculated rather than computing the distancing among means of grid cells. Our pairwise distance computation approach for attractive region mining (pARM) works and provides almost similar accuracy to the mean-based method, detailed in Section 3.1. However, it has two issues; first, it involves quadratic time complexity for point-to-point checking; second, we understand that the overall problem of attractive region mining for spatial trajectories naturally gets modeled as a graph mining problem. Any spatial trajectory involves a starting and ending point; hence, we get a graph. Therefore, we first model the problem as a pairwise distance checking among nodes in a spatial trajectory graph. The vertices are the endpoints of a journey or trajectory, and the edges are the paths between the endpoints. We compute the distances among the vertices to mine the attractive regions. This approach is called pairwise graph-based Attractive Region Mining (pgARM), explained in Section 3.2. Modeling data as a graph seems appropriate; however, the time complexity issue remains the same. To solve this problem, we propose a model based on locality-sensitive hashing (LSH). LSH is a fast similarity search algorithm that provides speed-ups by avoiding unnecessary similarity computation among distant data points. We refer to this approach as hashing and graph-based attractive region mining (hgARM) and explain it in Section 3.3. In this way, we propose a series of algorithms to effectively solve the ARM problem using spatial trajectory data.

#### 3.1. Pairwise attractive region mining (pARM): a brute-force strategy

In this section, we elaborate on our brute-force strategy to explore hot (aka attractive) regions in the spatial domain using pairwise computations. The idea behind our proposed approach, i.e. pairwise attractive region mining (pARM), is to perform pairwise distance computations among points of adjacent grid cells. Figure 1 shows an overview of our proposed solution, which compares its core characteristics to the current state-of-the-art solution (Zheng et al. 2018). It uses the same spatial point distribution to perform density-based clustering of cells corresponding to the cluster distance. However, the mean-based approach to compare cell distances leads to bias in data sets where the data distribution is skewed (O'Hagan et al. 2016). Therefore, we propose a pairwise



**Figure 1.** A comparative overview of GridDBScan (Zheng et al. 2018) and the proposed pARM approach.

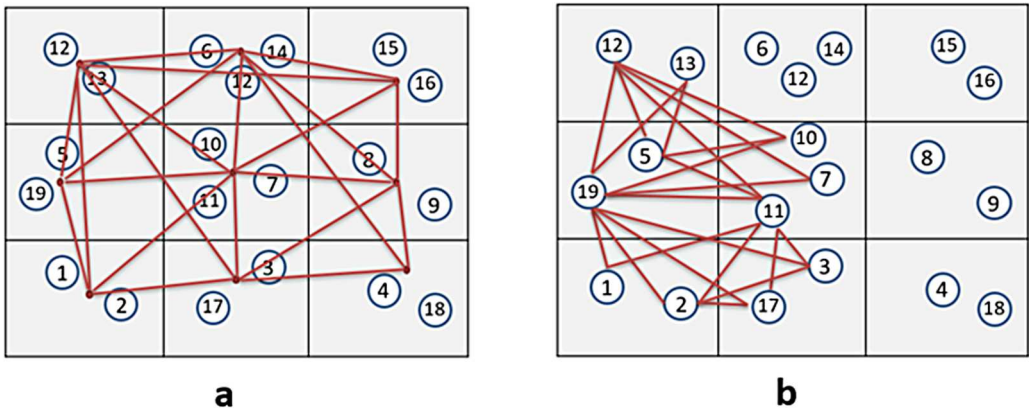
approach better suited to domains of skewed data distribution problems, such as power law. Figure 2 illustrates the difference between the two techniques.

The pARM approach performs pairwise attractive region mining using the direct distance computation between pairs of grid points instead of the cell means. The critical steps of our approach are to convert data points in the grid as data mapping, finding neighbors, and distance calculations, as illustrated in Algorithm 1. To improve spatial data analysis, our algorithm first converts the data points to a grid with longitude on the Y-axis and latitude on the X-axis, then divides them into cells by determining the horizontal and vertical edges for equal parts, as demonstrated in the following Equation (1), where *long* and *lat* denote longitude and latitude, respectively.

$$\begin{aligned} Edge_{horizontal} &= \frac{long_{max} - long_{min}}{k_{GridSize}} \\ Edge_{vertical} &= \frac{lat_{max} - lat_{min}}{k_{GridSize}} \end{aligned} \quad (1)$$

**Algorithm 1** Pairwise Attractive Region Mining (pARM)

- 1: **Input:**  $D$  a set of data points having longitudes and latitudes,  $\lambda$  a Distance coefficient (O'Hagan et al. 2016; Zheng et al. 2018),  $R$  radius to set limit of adjacent cells,  $K$  for grid cells
- 2: **Output:** Clusters of data points i.e. attractive regions
- 3: Divide  $D$  into  $K \times K$  grid cells
- 4: Map each point  $p \in D$  into its respective grid cell
- 5: Label a given cell of the grid as attractive if its member points are greater than  $\lambda$
- 6: For a given grid cell, compute pairwise spatial distances for each of its member data points against those of adjacent cell data points
- 7: Repeat the previous step until the pairwise distances of all cells are computed against all their adjacent cells
- 8: Using the majority voting scheme, group cells into attractive regions



**Figure 2.** Illustration of pairwise comparison instead of mean-based approach. (a) GridDBScan (Zheng et al. 2018) based on the mean of the cell for comparison. (b) Finding similar cells using a pairwise comparison is our first proposed approach.

### 3.1.1. Cluster identification using pARM

We introduce the concept of radius for a cell to achieve the aforementioned objective of pARM. The radius aims to set up the distance between the grid cells. For instance, the radius of '1' means that adjacent grid cells are neighbors of the selected cell. Similarly, the radius of 2 defines two adjacent cells in each direction as neighbors of the cell under consideration, and so on. Once the grid is created and neighbors are defined, we calculate the pairwise spatial distances between the members of adjacent cells. For example, if there are 4 points in cell 1 and 3 points in cell 2, the total pairs are  $4 * 3 = 12$ . In this way, spatial pairwise distance is computed among members' data points of all the grid cells that fall within the threshold of a given radius. Once the distance computation is complete, we start to create clusters. Since we have several grid cells, various adjacent cells are merged into groups. We use a majority voting scheme to combine cells into clusters. If most of the pairings of data points fall within the given user-defined distance threshold, we declare that both grid cells belong to the same group.

### 3.1.2. Discussion on pARM

The brute-force clustering approach of pARM does identify the clusters, but these clusters do not capture the intrinsic path aspect of spatial trajectories. A trajectory is usually a path that has starting and ending locations. These paths provide the information on the network that can be used to create more meaningful clusters. Moreover, it suffers from its quadratic time complexity because of  $n \times n$  comparisons between data points, which makes it impractical for very large-sized networks.

While motivating our work, GridDBScan (Zheng et al. 2018) uses density-based clustering on mean points of grid cells to find taxi travel patterns. However, this approach may not be optimal for datasets with skewed distributions. Our proposed variant, pARM, addresses this limitation by performing point-to-point distance computations within each grid cell. Although computationally more intensive, pARM establishes a foundation for developing further variants that elegantly solve the ARM problem. This novel problem formulation is the key strength of pARM, demonstrated by the effectiveness of its two extensions.

In this regard, to capture the real essence of trajectories, we model the data as a graph and propose a new approach called pgARM.

---

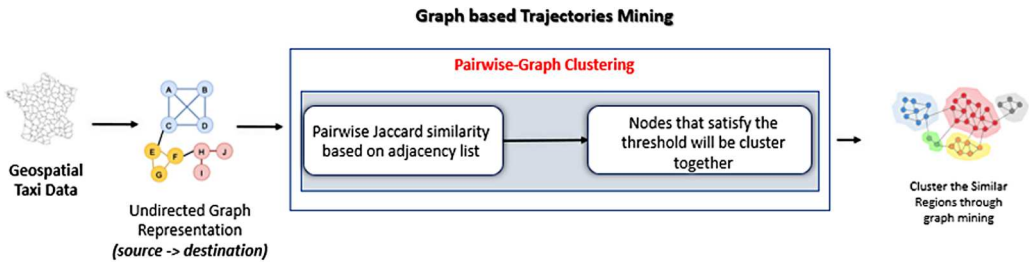
#### Algorithm 2 Pairwise Graph-based Attractive Region Mining (pgARM)

---

- 1: **Input:**  $D$  a set of data points that have longitudes and latitudes,  $\lambda$  Jaccard similarity threshold,  $R$  radius to set the limit of adjacent cells,  $K$  for grid cells
  - 2: **Output:** Clusters of data points, i.e. attractive regions
  - 3: Divide  $D$  into  $K \times K$  grid cells
  - 4: Map each point  $p \in D$  into its respective grid cell
  - 5: Map, a pair of data, points  $p, q \in D$  as source and destination points of a taxi's trajectory, to mark them as nodes of the graph  $G$  and the path between them to be an edge
  - 6: For a given pair of adjacent grid Cells  $i$  and  $j$ , if a node  $u \in Cell_i$  and node  $v \in Cell_j$  has neighborhood similarity above  $\lambda$ , then declare the pair  $u, v$  as member of same cluster
  - 7: Repeat the previous step until all pairs of nodes that are members of adjacent grid cells are marked into clusters
  - 8: Cluster each group of cells into an attractive region using a majority voting scheme
- 

### 3.2. Pairwise graph-based attractive region mining (pgARM)

We model trajectory data as a graph to learn the travel patterns and identify more meaningful clusters. In a graph, a node represents the source or destination of a trajectory, while an edge represents the path between them. We compute source node groups with similar destinations using this data modeling format. In this way, all nodes meeting the similarity criteria are clustered together. In the following, we discuss each step of the proposed algorithm in detail, while Figure 3 shows its general working.



**Figure 3.** Bird's eye view of proposed pgARM.

### 3.2.1. Graph formation from trajectory data

We transform the data points into an undirected graph where a node is a particular location on the trajectory, specified by its longitude and latitude. Typically, a taxi trajectory comprises the origin and destination as two points. We connect both points through an edge to represent a path followed by a taxi. Using this method, we create a graph representation of the trajectory, as shown in Figure 4.

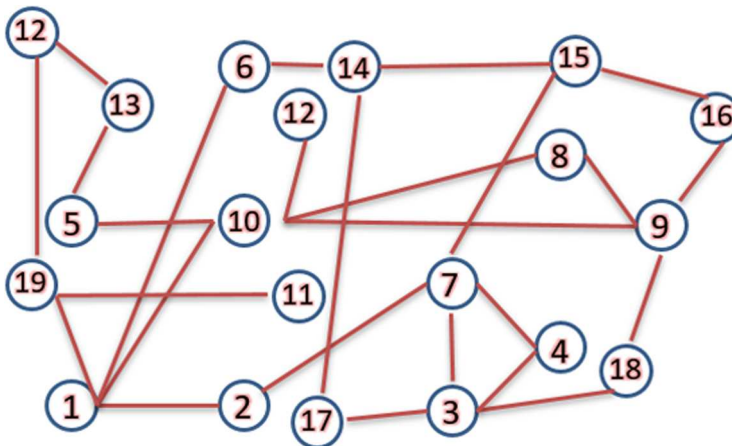
### 3.2.2. Cluster formation from trajectory graph

The next step of the algorithm is to find the adjacent nodes. In this regard, any two nodes are considered a pair if they have a count of common neighbors above a given threshold. Consequently, the nodes that meet the similarity threshold are clustered together. The similarity among any pair of nodes  $u$  and  $v$  is calculated using the Jaccard coefficient. This measure compares similarity using the adjacency list of each node, as illustrated in Equation (2), where  $N$  represents neighbors.

$$JSimilarity(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|} \quad (2)$$

### 3.2.3. Discussion on pgARM

The pARM approach suffers from the problems of higher computational complexity. Moreover, like GridDBScan (Zheng et al. 2018), it also misses using the use of an essential important and tangible concept of a path between the endpoints of any trajectory. In this regard, pgARM is effective, as it focuses on using the path information, i.e. the link structure among the data points, to perform



**Figure 4.** Representing trajectories of taxis as an undirected graph.



density-based clustering. Considering the connectivity among the endpoints of the trajectories, this variant aims to discover more meaningful clusters of the trajectories.

The pgARM approach captures information from the data network to maintain edge connectivity between points. Because of pairwise comparisons, it has quadratic time complexity. Ideally, we need a strategy that captures information from the graph and provides a low-cost solution.

### 3.3. Hashing-based graph attractive region mining (hgARM)

We introduce an efficient approach based on the hashing mechanism to overcome the computational overhead due to extensive pairwise comparisons, named attractive region mining (hgARM) to efficiently identify similar nodes without explicit pairwise comparisons. Figure 6 illustrates the overall idea of the hgARM approach, and Figure 5 illustrates the flow of the hgARM. It describes how the initial data points of a taxi are mapped to the corresponding grid cells of  $n \times n$  size. The next step is to convert those points into nodes and edges to create their minhash vector representation. Subsequently, our algorithm generates hash tables for each band and groups the data points into clusters based on their similarities, i.e. smaller distance among data points. Finally, majority voting is applied to determine the attractive regions. The step-by-step working of the hgARM strategy is illustrated in Algorithm 3 and is explained in subsequent paragraphs. LSH is applied to the trajectory graph of Figure 4 to identify the clusters. Locality Sensitive Hashing (LSH) has the property of avoiding unnecessary computations of similarity between unlikely pairs. In this way, hgARM is more efficient and scalable.

#### 3.3.1. Applying LSH on trajectory graph for cluster formation

LSH uses multiple hash functions on each node to find similar nodes for cluster formation. These are random permutations of the adjacency list of every node, as shown in step (a) of Figure 7 as  $\pi_1$  to  $\pi_4$ . Using these hash functions, we generate a minhash matrix of size  $N \times K$ , where  $N$  denotes the total number of nodes in the trajectory graph, and  $K$  represents the number of hash functions used. The minhash matrix obtained is shown in step (b) of Figure 7. The minhash matrix is a similarity-preserving matrix of the input graph that is much smaller in size and memory-preserving.

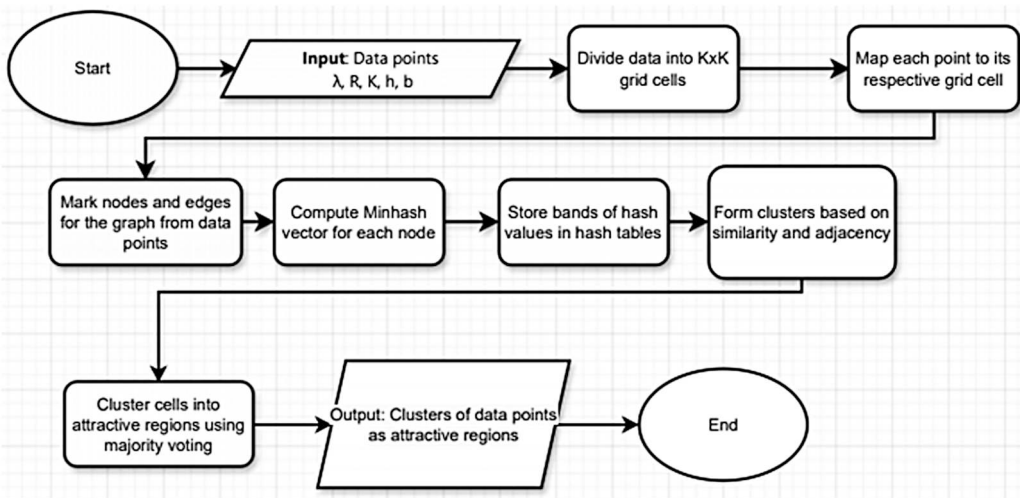
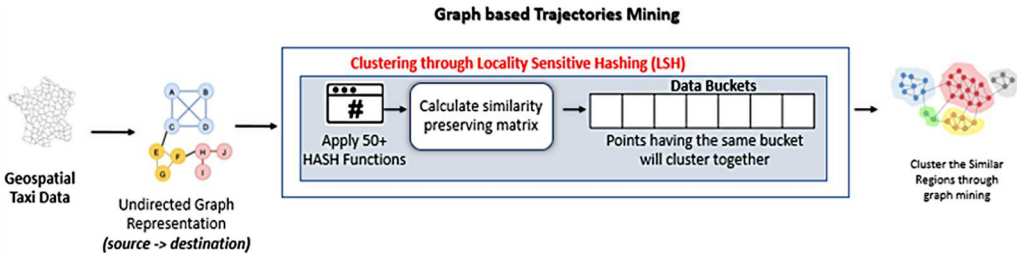


Figure 5. Flowchart diagram hgARM.



**Figure 6.** Bird's eye view of our proposed hgARM approach.

**Algorithm 3** Hashing-based Graph Attractive Region Mining (hgARM)

- 1: **Input:**  $D$  a set of data points having longitudes and latitudes,  $\lambda$  Jaccard similarity threshold,  $R$  radius to set limit of adjacent cells,  $K$  for grid cells,  $h$  number of hash functions for LSH,  $b$  number of bands/hash tables in LSH.
- 2: **Output:** Clusters of data points i.e. attractive regions
- 3: Divide  $D$  into  $K \times K$  grid cells
- 4: Map each point  $p \in D$  into its respective grid cell
- 5: Map pair of the data points  $p, q \in D$  as source and destination points of a taxi's trajectory to mark them as nodes of the graph  $G$  and the path between them to be an edge
- 6: Repeat Steps 4 and 5 for each  $u \in G$
- 7: Compute minhash vector  $m$  for each  $u$ , by applying  $h$  hash functions on its neighborhood
- 8: Divide  $m$  into  $b$  bands of  $r$  rows each and store each band's members as a hashed value in a bucket of the respective hash table
- 9: Retrieve all pairs of nodes  $u, v$  from adjacent grid Cells  $i$  and  $j$ , where  $u \in Cell_i$  and  $v \in Cell_j$  and has neighborhood similarity above  $\lambda$ , to declare the pair  $u, v$  as member of same cluster
- 10: Repeat the previous step until all pairs of nodes that are members of adjacent grid cells are marked into clusters
- 11: Cluster each group of cells into an attractive region using a majority voting scheme

As soon as we obtain the minhash matrix, we divide it into  $b$  bands having  $r$  rows each, as illustrated in Step (c) of Figure 7. This way, the objective is to group the nodes with the same minhash codes into buckets. Step (d) shows the status of the nodes after applying a hash function on each band. All nodes with the same minhash codes in a band produce the same hash value. For example, the minhash codes for nodes 2, 7, and 19 are in a gray row; therefore, these nodes have the same hash code 8 in step (d). In this way, all such nodes having the same minhash codes in a specific band fall into the same buckets of the corresponding hash tables. Finally, we perform unions of the buckets of each hash table to find clusters of similar nodes, i.e., the points having a lesser distance from each other.

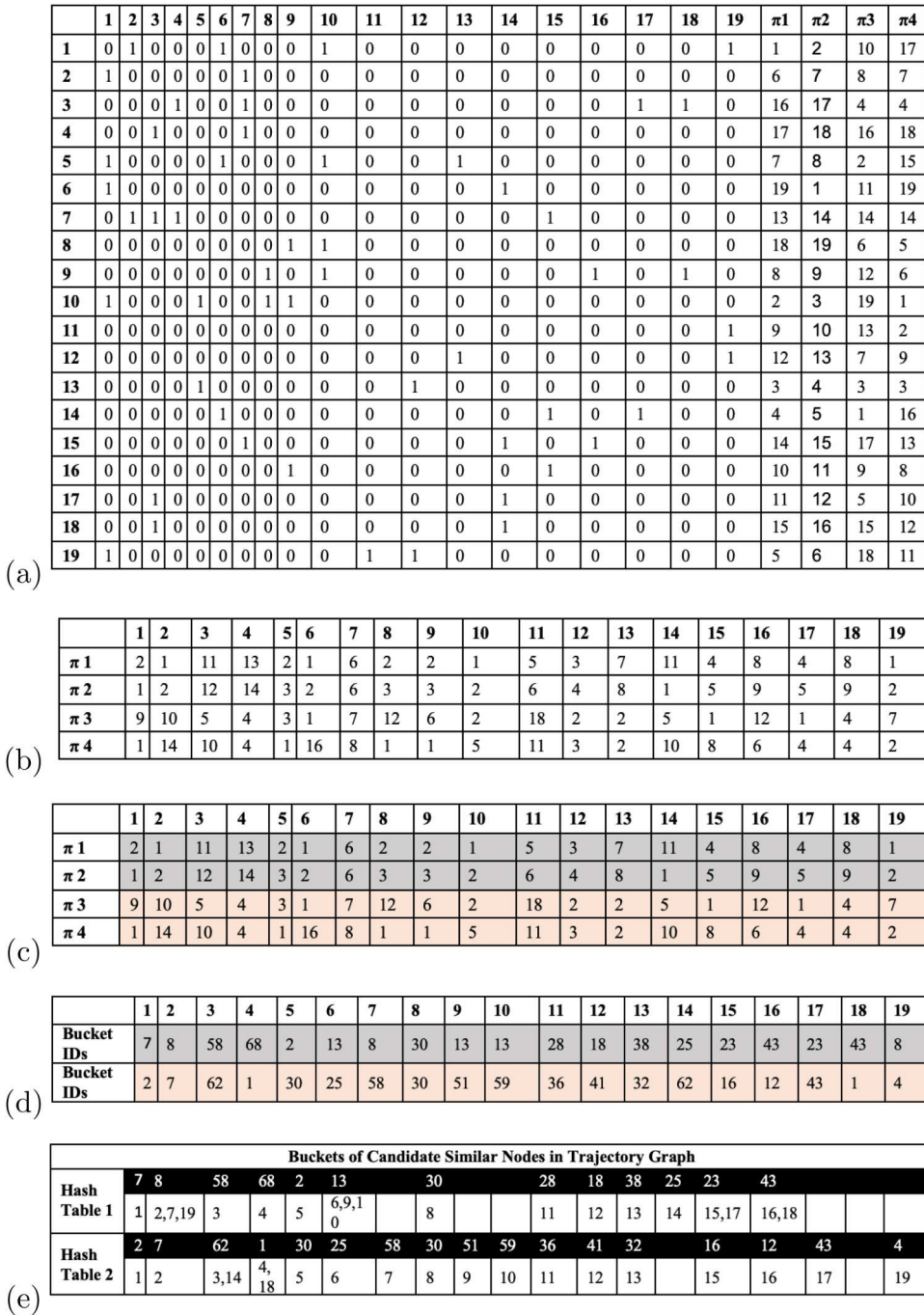
**3.3.2. Discussion on hgARM**

Our hgARM approach addresses the key challenges of the ARM problem faced by previous variants and GridDBScan. Uniquely, it integrates path information while optimizing computational efficiency. Using LSH (Locality-Sensitive Hashing), distance computations are intelligently limited to likely points. This innovation makes hgARM a highly effective and scalable solution for large datasets.

**4. Experiments**

In this section, we present an experimental evaluation of our solution. We performed experiments on the Ubuntu 20.04 LTS system with a core i5 processor with 8 GB RAM. All algorithms were implemented in Python 3.7. The experiments were carried out on four publicly available data sets, namely T-Drive,<sup>1</sup> ECML/PKDD,<sup>2</sup> and Road Networks of California<sup>3</sup> and Texas.<sup>4</sup>

The T-Drive trajectory sample offers a rich dataset: a week of activity from 10,357 taxis, comprising 15 million points and covering 9 million kilometers of travel. ECM-L/PKDD is a comprehensive dataset that details taxi operations in Porto, Portugal. The data covers a full year (01/07/2013–30/



**Figure 7.** Illustration of how to apply LSH on a trajectory graph/ (a) adjacency matrix representation of a toy graph in Figure 4 along with four hash functions of random permutations. (b) Minhash matrix. (c) Division of matrix into two bands. (d) Combined hash codes for every band using an arbitrary hash function. (e) Hash tables contain buckets of candidates with similar nodes.

06/2014) and includes trajectories of all 442 taxis in the city. Taxi rides are classified according to their origin (central, stand-alone, or street-hailed), and customer phone numbers are included when applicable. Each trip record features attributes such as trip ID, origin details, taxi ID,

timestamp, day type, missing data indicators, and a detailed GPS polyline tracking the trip's route at 15-second intervals. We model California and Texas road networks as graphs. Intersections and road endpoints become nodes, while the roads form undirected edges that connect these nodes.

We visualize the datasets in Figure 8 to better understand them. These visualizations clearly show how roads and paths are designed, highlighting intersections, areas of heavy traffic, and traffic flow. Each image shows the intricate layout of rural and urban road networks, allowing quick identification of congested areas and streamlined paths while separating packed and less crowded locations.

We compared our proposed algorithm pARM with that from Zheng et al. (2018), which we call GridDBScan. Using this experiment, we aim to verify that the mean-based approach of GridDBScan can be replaced with pARM. This forms the basis for our proposal to formulate the problem as a graph mining task. We compare GridDBScan and pARM based on the Davies–Bouldin score and execution time for clustering the data points. Davies–Bouldin score is a measure to investigate the compactness and separation of the clusters. It verifies the goodness of the clusters in terms of variation within the groups and separation between the clusters. We also compare the visual output of both algorithms for effectiveness evaluation.

We evaluate our algorithms, namely pgARM and hgARM, based on their execution times for data clustering, clustering coefficients, and modularity scores of the resultant clusters. For experiments using LSH, we used 50 hash functions of random permutations.

#### 4.1. Evaluation on spatial trajectories

We present an empirical evaluation of GridDBScan and pARM when the trajectory data is modeled in its native spatial format.

GridDBScan is a density-based clustering algorithm that identifies partitions with high-density areas of points separated by low-density areas. We use the Davies–Bouldin score, a relative validation measure for arbitrarily formed density-based clusters. The measure evaluates the quality

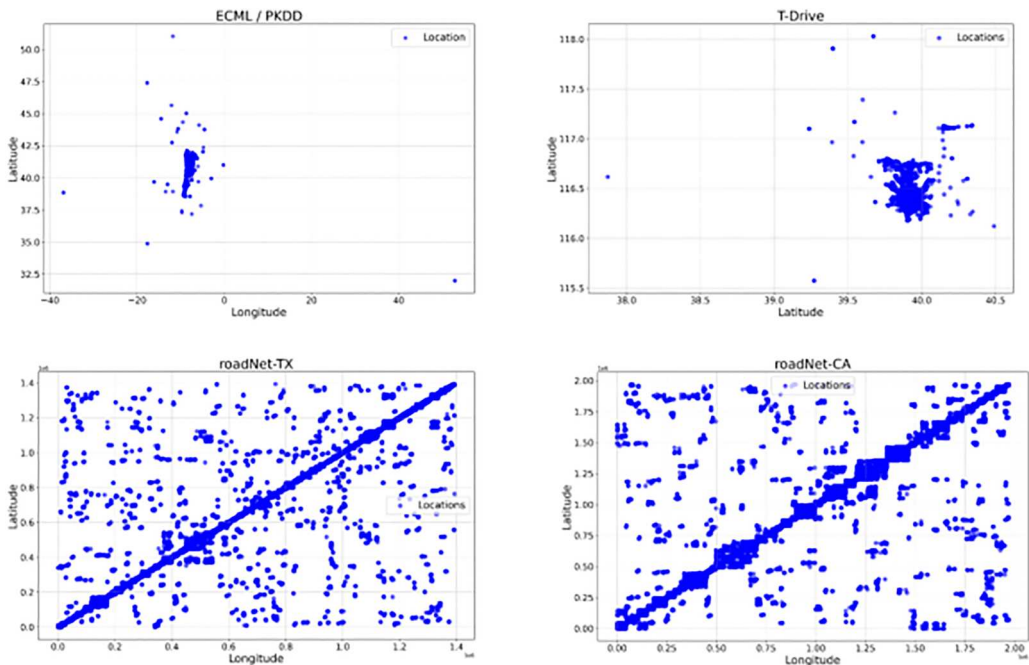
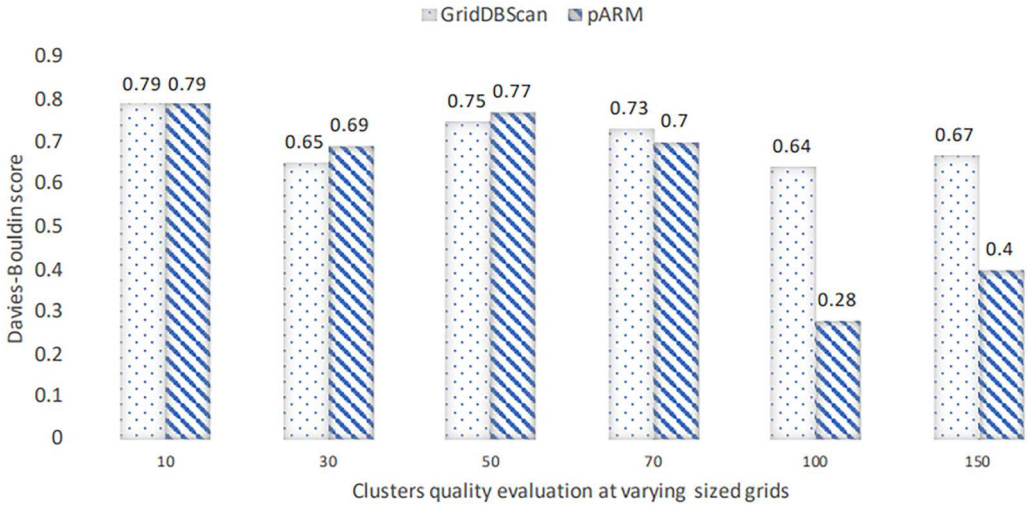


Figure 8. The visualization of the raw datasets used in the experiments.



**Figure 9.** Davies–Bouldin score of GridDBScan and pARM on different sizes of grids of T-Drive dataset.

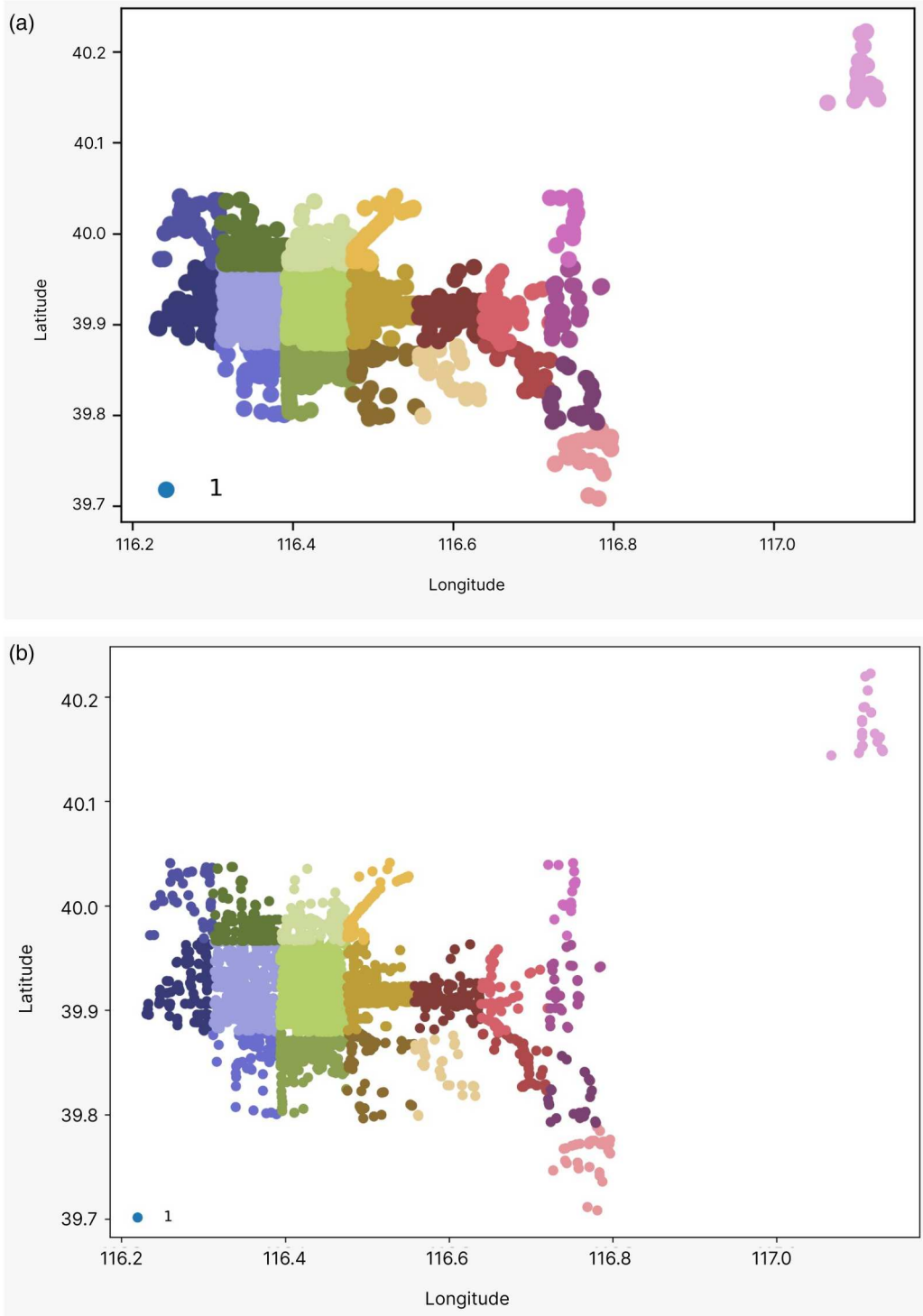
of clusters based on the relative density of connections between pairs of items. The index is based on a unique kernel density function, which is used to compute the density of objects (Moulavi et al. 2014). Figure 9 shows the results of the Davies–Bouldin score for the two methods, where we notice a better Davies–Bouldin score for large grids. We observe that pARM produces Davies–Bouldin scores similar to GridDBScan for grid sizes of 10 to 70. However, GridDBScan produces better results for greater grid sizes like 100 and 150. GridDBScan produces better results because each grid cell has multiple data points, and when we increase the number of grid cells, the data points in each grid cell are reduced. This ultimately leads to ignoring the noise data points, reducing the inter-cluster distance. Interestingly, we observe that the scores again rise when the grid size is increased, i.e., the number of cells above 150.

We also visualize the outputs of both algorithms for a grid size of 30 in Figure 10 for effectiveness evaluation. We observe a similar outcome for both algorithms, proving that our proposed aim works well compared to the state-of-the-art GridDBScan. On the other hand, when comparing the execution time of both algorithms, pARM happens to be an expensive approach than GridDBScan because of its quadratic time complexity. However, the execution time of pARM significantly drops with increasing grid size. This happens because the member data points of each grid cell are reduced; hence, comparisons between adjacent grid cells are also reduced. Therefore, both approaches consume a similar amount of time at a higher grid size, as depicted in Figure 11.

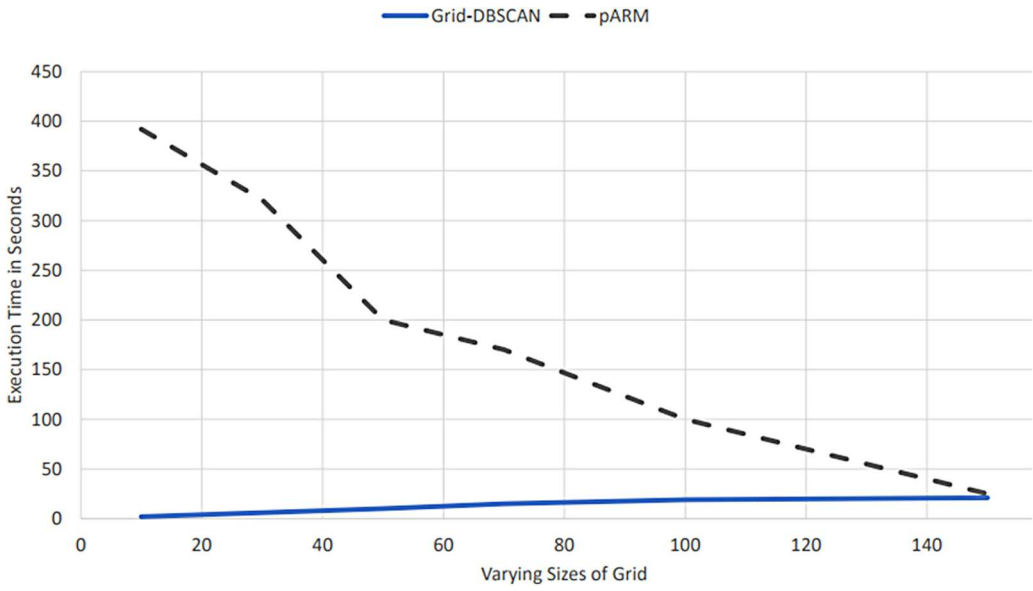
#### 4.2. Evaluation using trajectory graphs

We present an empirical evaluation of our proposed algorithms pgARM and hgARM when the trajectory data are modeled in graph format.

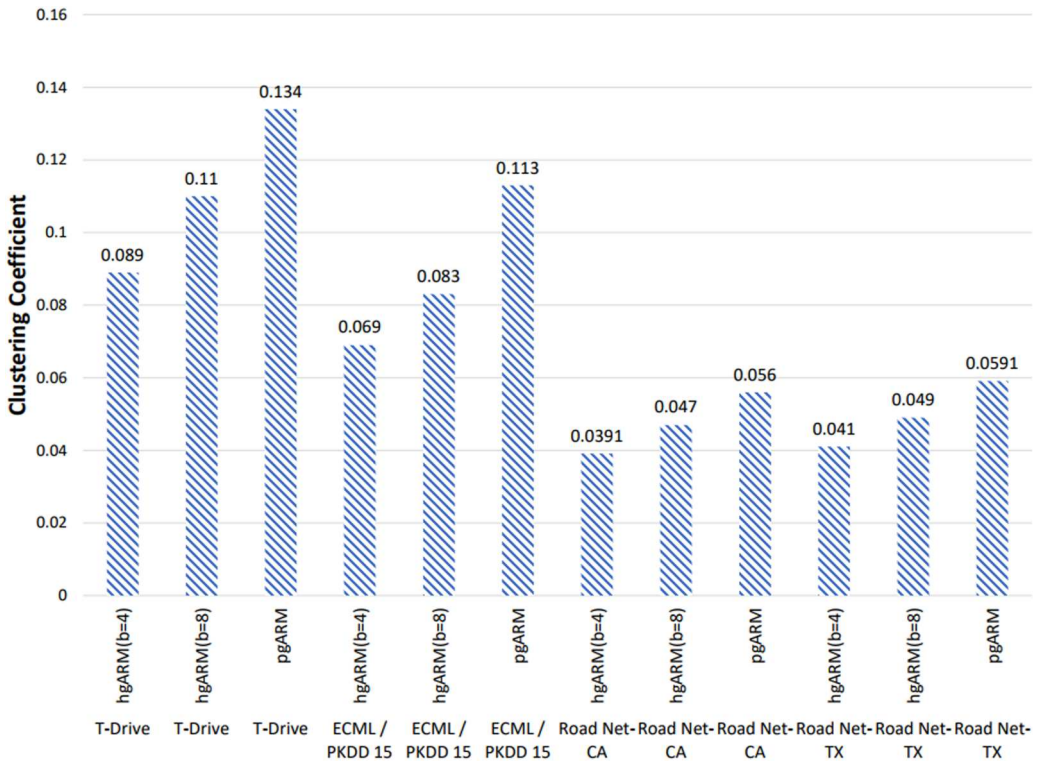
For this evaluation, we compare the clustering coefficient and modularity of the clusters produced by both the approaches and their execution times. The clustering coefficient measures the degree to which the nodes in a graph tend to cluster together. Modularity measures the strength of the division of a network into clusters by comparing the density of edges of the intra-cluster with the inter-cluster. We did not compare pgARM and hgARM with GridDBScan using the Davies–Bouldin score because it does not apply to graph-based clustering. The reason is that it is based on the distance or similarity metrics irrelevant to the graph data. The notion of ‘within-



**Figure 10.** Visualization of clusters produced by (a) GridDBScan and (b) pARM where the size of Grid is 30 of T-Drive dataset.

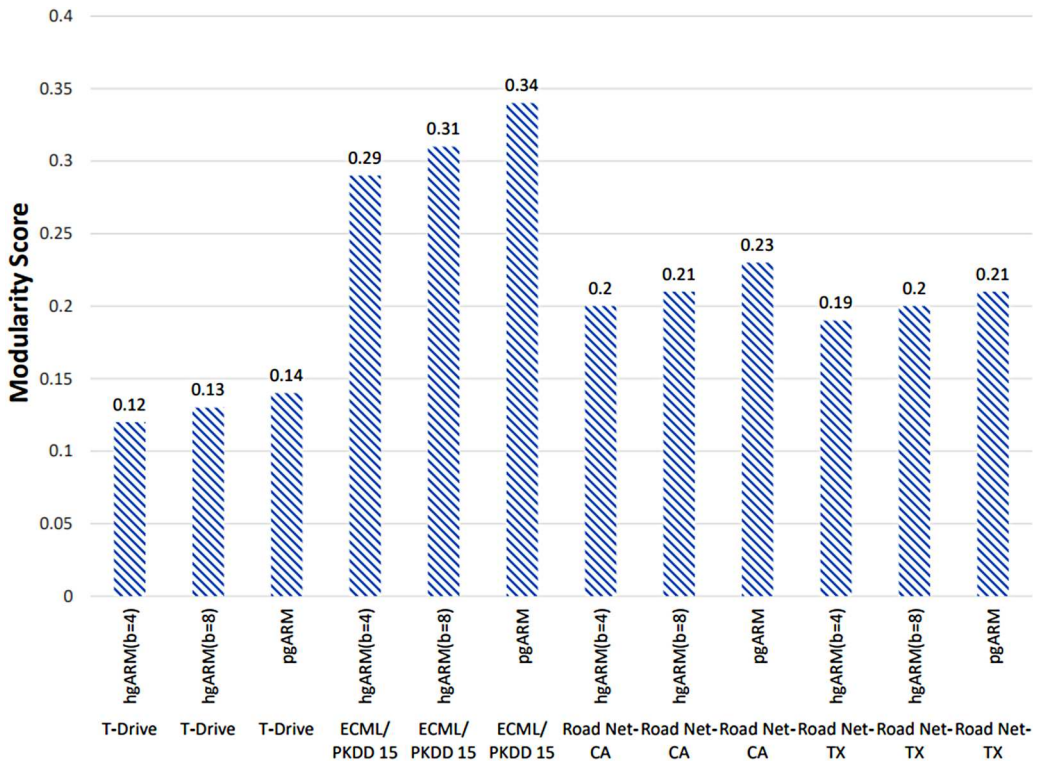


**Figure 11.** Execution time analysis of GridDBScan and pARM of T-Drive dataset.



**Figure 12.** Evaluation of clustering coefficient for the proposed algorithms pgARM and hgARM on all four datasets.

cluster scatter' and 'between-cluster separation' does not have a straightforward interpretation in graph-based clustering.



**Figure 13.** Evaluation of modularity for the proposed algorithms pgARM and hgARM on all 4 datasets.

In each comparison for all four datasets, we observe that pgARM provides better results for the clustering coefficient than hgARM having the minhash columns divided into 4 bands, as shown in Figure 12. This is so because pgARM performs explicit pairwise similarity comparisons among the nodes in the graph. If we observe the comparison of pgARM with 8 bands of hgARM, results are quite promising (see Figure 13) and the same trend holds for modularity. During the evaluation of the clusters using modularity, pgARM consistently outperformed hgARM in terms of identifying close groups or communities for both bands, where the band 8 results are close to pgARM. We can conclude that increasing the number of bands yields better results for the hgARM.

The experiments reveal that both approaches identify accurate groupings to some extent, pgARM performs better in a variety of settings, demonstrating that it is more reliable for this purpose. The difference is especially obvious in the ECML/PKDD dataset, where pgARM stands out firmly. However, neither technique performed well for the T-Drive dataset, implying that the dataset may be more difficult to work with or that both methods suffer in some cases. Although pgARM performs well in modularity or clustering coefficient, it runs poorly in time. On the other hand, hgARM only performs such comparisons between similar nodes and misses some genuinely similar nodes due to its approximation strategy. Consequently, hgARM consumes less execution time than its pgARM and, therefore, is more scalable.

However, as the number of bands increases from 4 to 8, its execution time approximately doubles, as seen in execution time analysis comparisons in Figures 14, 15, 16, and 17. The hgARM approach with 4 and 8 bands differs in execution time because of having a shorter execution time of hash table creation and searching through the LSH index. Dividing the minhash column into 4 bands results in 4 hash tables; hence, less time is required for the hash table creation and searching for closer data points. We observe a similar execution time for both variations of hgARM in all four datasets. In particular, for road networks of Texas and California, since both



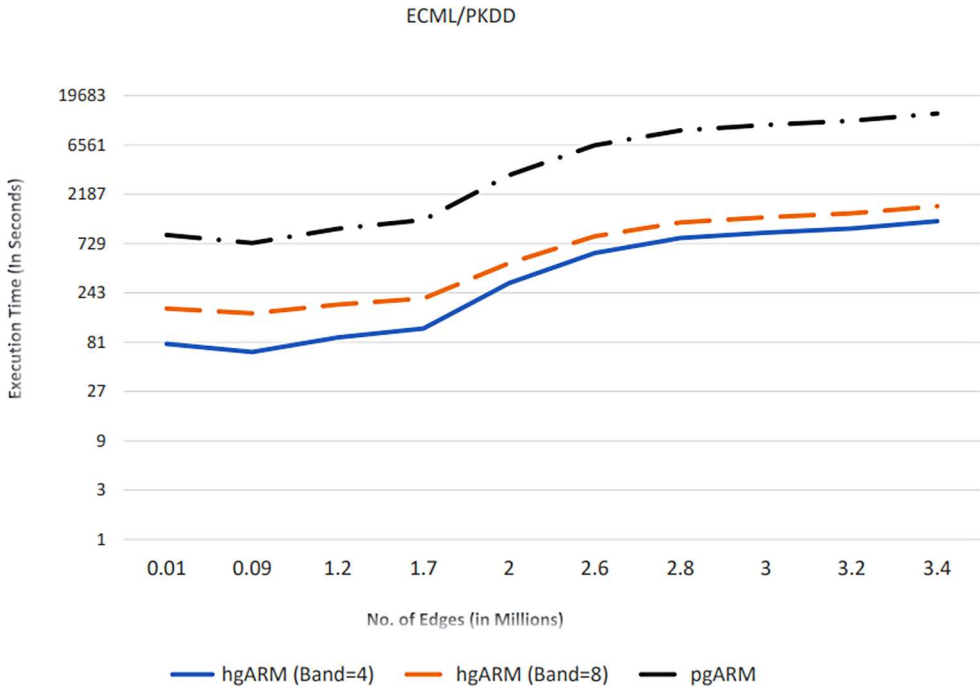


Figure 14. Execution time comparison of proposed graph-based algorithms pgARM and hgARM on ECML/PKDD dataset.

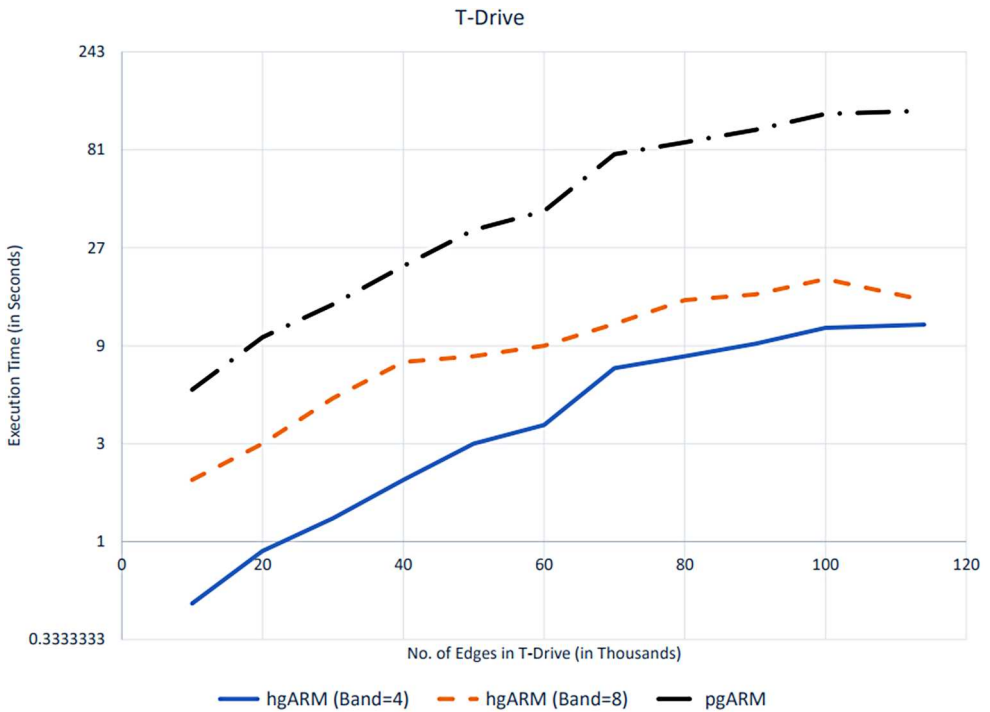
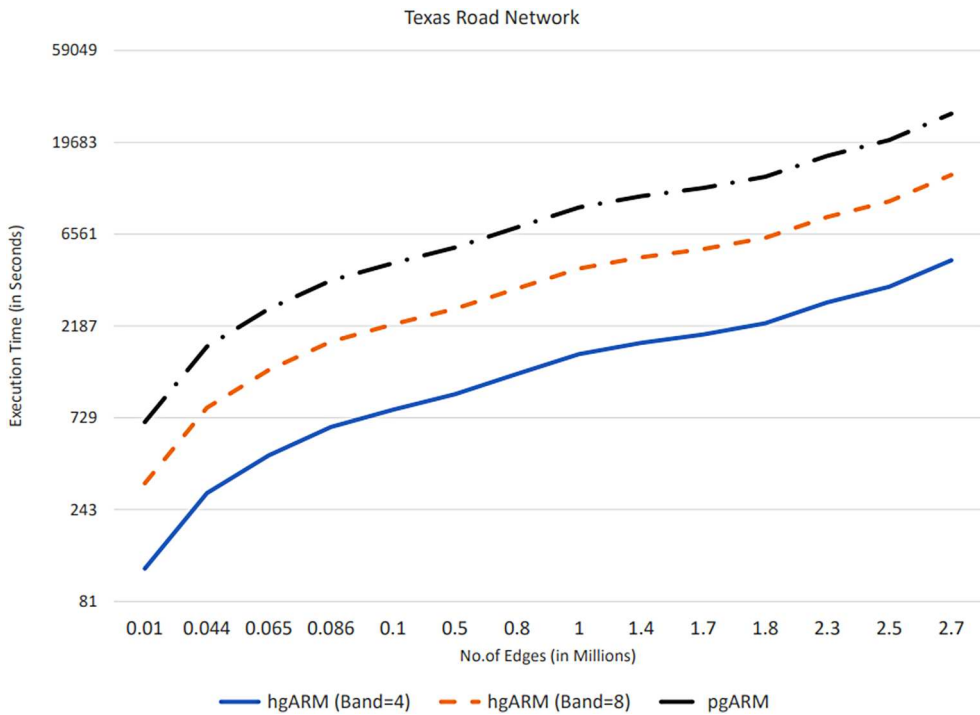
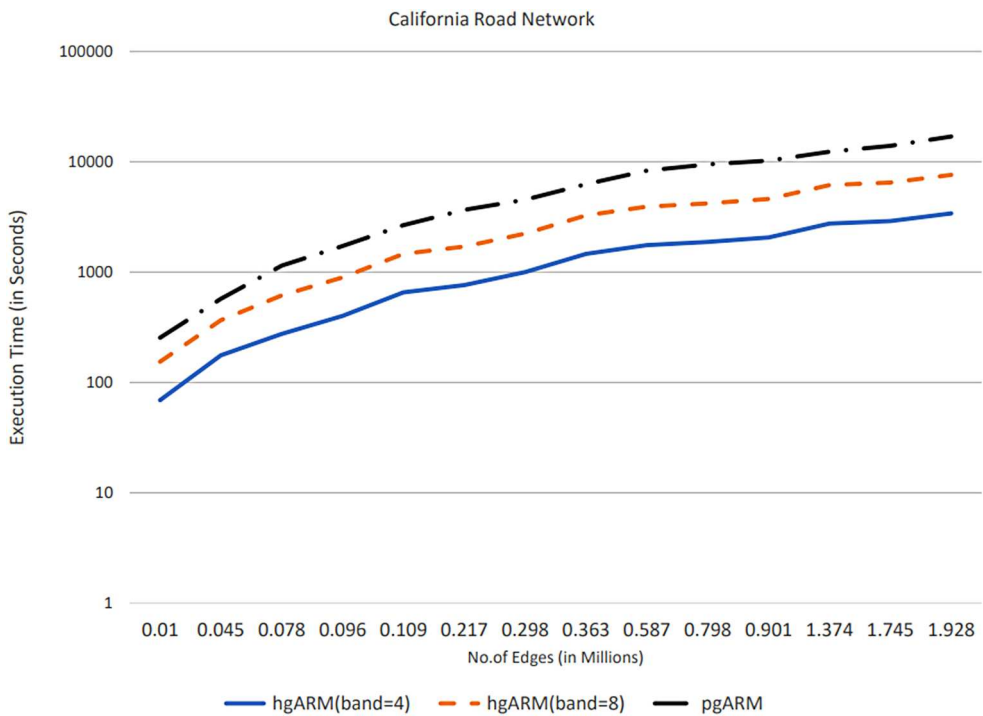


Figure 15. Execution time comparison of proposed graph-based algorithms pgARM and hgARM on T-Drive dataset.



**Figure 16.** Execution time comparison of proposed graph-based algorithms pgARM and hgARM on dataset of Texas road network.



**Figure 17.** Execution time comparison of proposed graph-based algorithms pgARM and hgARM on a dataset of California road network.

datasets possess similar network properties of clustering coefficient, where California has more number of triangles than Texas. However, the fractions of closed triangles are almost the same for both of these datasets. However, pgARM consumes more time, but emerges as the strongest tool to detect tightly associated groups in the data we examined, as can be observed in Figures 12 and 13 for the clustering coefficient and modularity score. Unlike hgARM, it uses path information and does not do any approximation for cluster discovery.

As a result, we conclude that our proposed idea to solve the problem of attractive region mining by transforming the spatial data sets as graph data sets perform well. We obtain clusters of descent quality in a reasonable amount of time. GridDBScan is not a scalable algorithm. However, with the proposed variants, the ARM problem gets into a new shape and is accelerated through using LSH.

During the problem formulation and implementation, We faced several challenges during the study. LSH's memory requirements, which scale with dataset size, necessitate future work considering memory optimization techniques like adjusting the number of bands or utilizing parallel/distributed platforms. Additionally, transforming the problem domain from Euclidean space to graphs for efficient solutions proved challenging, mainly when modeling real-world datasets based on coordinates. This required careful formulation of coordinate-to-graph data transformation using LSH. Designing suitable hashing functions for LSH implementation was another difficulty. Finally, selecting appropriate evaluation measures (e.g. Davies–Bouldin) for comparative analysis between graph-based and non-graph-based approaches presented its complexities.

## 5. Conclusion and future directions

In this paper, we presented a novel solution to the problem of attractive region mining (ARM) for taxi services, shifting the focus from spatial data alone to trajectory-based insights. By transforming spatial data into trajectory graphs, we achieved more meaningful clusters of attractive regions. Our method demonstrated superior time efficiency, clustering coefficient, and modularity scores with real-world datasets. This work advances our understanding of trajectory-based clustering and has significant implications for optimizing taxi services, driver efficiency, and broader intelligent transportation systems.

Future research could explore applications of this approach in managing public bike sharing systems and other urban mobility services. Our work presents several exciting avenues for future exploration, including optimizing performance through parallel execution platforms, automating LSH parameter determination for efficiency, developing methods to analyze weighted trajectories (considering path costs), and integrating time series analysis to uncover additional insights from trajectory data. These extensions have the potential to further enhance the performance, scalability, and impact of our approach.

## Notes

1. <https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/>.
2. <https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i> respectively.
3. <https://snap.stanford.edu/data/roadNet-CA.html>.
4. <https://snap.stanford.edu/data/roadNet-TX.html>.

## Acknowledgments

The authors wish to express their appreciation to all the stakeholders in this investigation, including the Islamic University of Madinah, FAST-NU, and Birmingham City University. The code of the proposed strategy will be provided on request. In this collaborative research effort, the authors made substantial contributions, each playing a pivotal role in the development and execution of the study. Kifayat Ullah Khan (K.U.K.) and Muhammad Toqeer (M.T.) contributed extensively to conceptualizing the research, defining the overarching methodology, and providing critical insights into trajectory analysis techniques. Muhammad Toqeer (M.T.) played a vital role in the implementation

phase, transforming spatial data sets into trajectory graphs and devising algorithms for cluster identification. His expertise in data processing and computational analysis significantly improved the technical aspects of the study. Waqas Nawaz (W.N.) focused on the comprehensive literature review, ensuring the study's contextual grounding in existing research while contributing to the interpretation of results. Additionally, W.N. actively participated in the discussion and synthesis of findings, contributing to the overall coherence and depth of the article. Through their combined efforts, the authors collaboratively shaped and executed the research, resulting in the submission of the manuscript for publication.

## Data availability

The experiments were carried out on four publicly available data sets, namely T-Drive, ECML/PKDD, and Road Networks of California and Texas.

- **T-Drive:** <https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/>
- **ECML/PKDD:** <https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data>
- **Road Networks of California:** <https://snap.stanford.edu/data/roadNet-CA.html>
- **Road Networks of Texas:** <https://snap.stanford.edu/data/roadNet-TX.html>

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This research is supported by the Deanship of Scientific Research of the Islamic University of Madinah, KSA under the research groups (first) project no. 956.

## ORCID

Waqas Nawaz  <http://orcid.org/0000-0002-9989-6163>

## References

- Cai, Guochen, Chihiro Hio, Luke Bermingham, Kyungmi Lee, and Ickjai Lee. 2014. "Sequential Pattern Mining of Geo-Tagged Photos with An Arbitrary Regions-Of-interest Detection Method." *Expert Systems with Applications* 41 (7): 3514–3526. <https://doi.org/10.1016/j.eswa.2013.10.057>.
- Cai, Hua, Xiaowei Zhan, Ji Zhu, Xiaoping Jia, Anthony S. F. Chiu, and Ming Xu. 2016. "Understanding Taxi Travel Patterns." *Physica A: Statistical Mechanics and Its applications* 457:590–597. <https://doi.org/10.1016/j.physa.2016.03.047>.
- Chavhan, Suresh, Deepak Gupta, Chandana Nagaraju, A. Rammohan, Ashish Khanna, and Joel J. P. C. Rodrigues. 2021. "An Efficient Context-Aware Vehicle Incidents Route Service Management for Intelligent Transport System." *IEEE Systems Journal* 16 (1): 487–498. <https://doi.org/10.1109/JSYST.2021.3066776>.
- Chen, Xinqiang, Zichuang Wang, Qiaozhi Hua, Wen-Long Shang, Qiang Luo, and Keping Yu. 2022. "AI-empowered Speed Extraction Via Port-Like Videos for Vehicular Trajectory Analysis." *IEEE Transactions on Intelligent Transportation Systems* 24 (4): 4541–4552. <https://doi.org/10.1109/TITS.2022.3167650>.
- Cheng, Tao, James Haworth, Berk Anbaroglu, Garavig Tanaksaranond, and Jiaqiu Wang. 2021. "Spatio-Temporal Data Mining." In *Handbook of Regional Science*, edited by M. M. Fischer and P. Nijkamp. Berlin, Heidelberg: Springer.
- Deng, Yajuan, Meiyi Li, Qing Tang, Renjie He, and Xianbiao Hu. 2020. "Heterogenous Trip Distance-Based Route Choice Behavior Analysis Using Real-World Large-Scale Taxi Trajectory Data." *Journal of Advanced Transportation* 2020:1–16.
- De Souza, Allan M., Roberto S. Yokoyama, Guilherme Maia, Antonio Loureiro, and Leandro Villas. 2016. "Real-Time Path Planning to Prevent Traffic Jam Through an Intelligent Transportation System." In *2016 IEEE Symposium on Computers and Communication (ISCC)*, 726–731.
- Fu, Xin, Chengyao Xu, Yuteng Liu, Chi-Hua Chen, F. J. Hwang, and Jianwei Wang. 2022. "Spatial Heterogeneity and Migration Characteristics of Traffic Congestion—A Quantitative Identification Method Based on Taxi Trajectory

- Data." *Physica A: Statistical Mechanics and Its Applications* 588:126482. <https://doi.org/10.1016/j.physa.2021.126482>.
- Gaffney, Scott, and Padhraic Smyth. 1999. "Trajectory Clustering with Mixtures of Regression Models." In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 63–72.
- Gao, Xu, and Fusheng Yu. 2017. "Trajectory Clustering Using a New Distance Based on Minimum Convex Hull." In *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSACIS)*, 1–6.
- Guan, Bo, Liangxu Liu, and Jinyang Chen. 2013. "Using Relative Distance and Hausdorff Distance to Mine Trajectory Clusters." *TELKOMNIKA Indonesian Journal of Electrical Engineering* 11 (1): 115–122. <https://doi.org/10.11591/telkomnika.v11i1.1877>.
- Hamdi, Ali, Khaled Shaban, Abdelkarim Erradi, Amr Mohamed, Shakila Khan Rumi, and Flora D. Salim. 2022. "Spatiotemporal Data Mining: A Survey on Challenges and Open Problems." *Artificial Intelligence Review* 55 (2): 1441–1488. <https://doi.org/10.1007/s10462-021-09994-y>.
- Hou, Minghui, Natalie Cruz, Chris R. Glass, and Sherrie Lee. 2021. "Transnational Postgraduates: Navigating Academic Trajectories in the Globalized University." *International Studies in Sociology of Education* 30 (3): 306–324. <https://doi.org/10.1080/09620214.2020.1853590>.
- Huang, Liping, Yongjian Yang, Hechang Chen, Yunke Zhang, Zijia Wang, and Lifang He. 2022. "Context-Aware Road Travel Time Estimation by Coupled Tensor Decomposition Based on Trajectory Data." *Knowledge-Based Systems* 245:108596. <https://doi.org/10.1016/j.knosys.2022.108596>.
- Kharrat, Ahmed, Iulian Sandu Popa, Karine Zeitouni, and Sami Faiz. 2008. "Clustering Algorithm for Network Constraint Trajectories." In *Headway in Spatial Data Handling: Lecture Notes in Geoinformation and Cartography*, edited by A. Ruas and C. Gold, 631–647. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-540-68566-1>.
- Lan, Shiyong, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. "Dstagnn: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting." In *International Conference on Machine Learning*, 11906–11917.
- Lee, Jae-Gil, Jiawei Han, and Kyu-Young Whang. 2007. "Trajectory Clustering: A Partition-And-Group Framework." In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, 593–604.
- Li, Zhenhui, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. "Mining Periodic Behaviors for Moving Objects." In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1099–1108.
- Li, Huanhuan, Jasmine Siu Lee Lam, Zaili Yang, Jingxian Liu, Ryan Wen Liu, Maohan Liang, and Yan Li. 2022. "Unsupervised Hierarchical Methodology of Maritime Traffic Pattern Extraction for Knowledge Discovery." *Transportation Research Part C: Emerging Technologies* 143:103856. <https://doi.org/10.1016/j.trc.2022.103856>.
- Li, Ye, and Hongxiang Ren. 2022. "Visual Analysis of Vessel Behaviour Based on Trajectory Data: A Case Study of the Yangtze River Estuary." *ISPRS International Journal of Geo-Information* 11 (4): 244. <https://doi.org/10.3390/ijgi11040244>.
- Li, Mingqian, Panrong Tong, Mo Li, Zhongming Jin, Jianqiang Huang, and Xian-Sheng Hua. 2021. "Traffic Flow Prediction with Vehicle Trajectories." In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35. 294–302.
- Li, Shiqiang, Weize Wang, Jiawei Shan, Heng Qi, Yanming Shen, and Baocai Yin. 2019. "An Effective Spatio-Temporal Query Framework for Massive Trajectory Data in Urban Computing." In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, 586–593.
- Li, Wu, Shengchuan Zhao, Jingwen Ma, Otto Anker Nielsen, and Yu Jiang. 2023. "Book-Ahead Ride-Hailing Trip and Its Determinants: Findings From Large-Scale Trip Records in China." *Transportation Research Part A: Policy and Practice* 178:103875.
- Liu, Dongchang, Shih-Fen Cheng, and Yiping Yang. 2015. "Density Peaks Clustering Approach for Discovering demand Hot Spots in City-Scale Taxi Fleet Dataset." In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 1831–1836.
- Liu, Jin, Xiao Yu, Zheng Xu, Kim-Kwang Raymond Choo, Liang Hong, and Xiaohui Cui. 2017. "A Cloud-Based Taxi Trace Mining Framework for Smart City." *Software: Practice and Experience* 47 (8): 1081–1094.
- Mao, Feng, Minhe Ji, and Ting Liu. 2016. "Mining Spatiotemporal Patterns of Urban Dwellers From Taxi Trajectory Data." *Frontiers of Earth Science* 10 (2): 205–221. <https://doi.org/10.1007/s11707-015-0525-4>.
- Moulavi, Davoud, Pablo A. Jaskowiak, Ricardo J. G. B Campello, Arthur Zimek, and Jörg Sander. 2014. "Density-Based Clustering Validation." In *Proceedings of the 2014 SIAM International Conference on Data Mining*, 839–847.
- Nikitopoulos, Panagiotis, Aris-Iakovos Paraskevopoulos, Christos Doukeridis, Nikos Pelekis, and Yannis Theodoridis. 2018. "Hot Spot Analysis Over Big Trajectory Data." In *2018 IEEE International Conference on Big Data (Big Data)*, 761–770.
- Ohadi, Negar, Ali Kamandi, Mahmood Shabankhah, Seyed Mohsen Fatemi, Seyed Mohsen Hosseini, and Alireza Mahmoudi. 2020. "Sw-DBScan: A Grid-Based DBScan Algorithm for Large Datasets." In *2020 6th International Conference on Web Research (ICWR)*, 139–145.

- O'Hagan, Adrian, Thomas Brendan Murphy, Isobel Claire Gormley, Paul D. McNicholas, and Dimitris Karlis. 2016. "Clustering with the Multivariate Normal Inverse Gaussian Distribution." *Computational Statistics & Data Analysis* 93:18–30. <https://doi.org/10.1016/j.csda.2014.09.006>.
- Olayode, Isaac Oyeyemi, Bo Du, Lagouge Kwanda Tartibu, and Frimpong Justice Alex. 2023. "Traffic Flow Modelling of Long and Short Trucks Using a Hybrid Artificial Neural Network Optimized by Particle Swarm Optimization." *International Journal of Transportation Science and Technology* 1–19. <https://doi.org/10.1016/j.ijst.2023.04.004>.
- Pachni-Tsitiridou, Olga, and Konstantinos Fouskas. 2019. "Location-Aware Technologies: How They Affect Customer Experience." In *Strategic Innovative Marketing and Tourism: 7th ICSIMAT, Athenian Riviera, Greece*, Vol. 2018. 1199–1206.
- Paulsen, Mads, Thomas Kjær Rasmussen, and Otto Anker Nielsen. 2021. "Impacts of Real-Time Information Levels in Public Transport: A Large-Scale Case Study Using An Adaptive Passenger Path Choice Model." *Transportation Research Part A: Policy and Practice* 148:155–182.
- Qi, Hong, and Panpan Liu. 2018. "Mining Taxi Pick-Up Hotspots Based on Spatial Clustering." In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 1711–1717.
- Saptawati, Gusti Ayu Putri. 2017. "Spatio-Temporal Mining to Identify Potential Traffic Congestion Based on Transportation Mode." In *2017 International Conference on Data and Software Engineering (ICoDSE)*, 1–6.
- Takimoto, Yoshiaki, Kento Sugiura, and Yoshiharu Ishikawa. 2017. "Extraction of Frequent Patterns Based on Users' Interests from Semantic Trajectories with Photographs." In *Proceedings of the 21st International Database Engineering & Applications Symposium*, 219–227.
- Tran, Duy Hoang, Pieter Leyman, and Patrick De Causmaecker. 2022. "Adaptive Passenger-Finding Recommendation System for Taxi Drivers with Load Balancing Problem." *Computers & Industrial Engineering* 169:108187. <https://doi.org/10.1016/j.cie.2022.108187>.
- Wang, Zheng, Kun Fu, and Jieping Ye. 2018. "Learning to Estimate the Travel Time." In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 858–866.
- Wang, Yang, Zhengyang Zhou, Kai Liu, Xike Xie, and Wenhua Li. 2020. "Large-Scale Intelligent Taxicab Scheduling: A Distributed and Future-Aware Approach." *IEEE Transactions on Vehicular Technology* 69 (8): 8176–8191. <https://doi.org/10.1109/TVT.2020.2999999>.
- Zhang, Peng, Jun Zheng, Hailun Lin, Chen Liu, Zhuofeng Zhao, and Chao Li. 2023. "Vehicle Trajectory Data Mining for Artificial Intelligence and Real-Time Traffic Information Extraction." *IEEE Transactions on Intelligent Transportation Systems* 24 (11): 13088–13098. <https://doi.org/10.1109/TITS.2022.3178182>.
- Zheng, Linjiang, Dong Xia, Xin Zhao, Longyou Tan, Hang Li, Li Chen, and Weining Liu. 2018. "Spatial-temporal Travel Pattern Mining Using Massive Taxi Trajectory Data." *Physica A: Statistical Mechanics and Its Applications* 501:24–41. <https://doi.org/10.1016/j.physa.2018.02.064>.