AN EFFICIENT PROCESSING OF SPATIO TEMPORAL AGGREGATE QUERIES

A PROJECT REPORT

Submitted by

FARHANA A REHIM (TKM21MCA-2018)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATION

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2023

DECLARATION

I undersigned hereby declare that the project report on **AN EFFICIENT PROCESSING OF SPATIO TEMPORAL AGGREGATE QUERIES**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Vaheetha Salam. This submission represents my ideas in my own words and where ideas or words of others have been included,I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

19-05-2023

FARHANA A REHIM

DEPT. OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING, KOLLAM

2022 - 23

CERTIFICATE

This is to certify that the report entitled **AN EFFICIENT PROCESSING OF SPATIO TEMPORAL AGGREGATE QUERIES** submitted by **FARHANA A REHIM** (TKM21MCA-2018) to the APJ Abdul Kalam Technological University in partial fulfillment of the Masters degree in Computer Applications is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor　　Head of the Department　　External Examiner

Acknowledgement

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, Department of Computer Applications, for providing me with best facilities.

I would like to thank my coordinator and project guide **Prof. Vaheetha Salam**, Department of Computer Applications, who motivated me throughout the project .I would like to thank **Dr. Nishad A** who guided me throughout my work.

I owe my thanks to my advisor **Prof. Natheera Beevi M**, Department of Computer Applications, for her continuous guidance and support.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study. I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

FARHANA A REHIM

ABSTRACT

With recent advances in positioning, telemetry, and telecommunication technologies, the wide availability of devices that produce information about the position of an object in some time, enormous amounts of data about moving objects are being collected and employed by many applications. Moving objects are objects (points) that change their locations (geometric attributes) over time, which requires a higher update frequency. Querying and analyzing this data can give more insightful knowledge on the pattern of the mobility of the object and the interest evinced by visitors in a geographic location.

AN EFFICIENT PROCESSING OF SPATIO TEMPORAL AGGREGATE QUERIES uses an algorithm called SemTraClus which helps in identifying, clustering, and prioritizing semantic regions. Significant locations of a geographical area called “Points of Interest” are extracted using three main methods stay point detection, revisited locations, and intersecting points of different trajectories. These identified regions are clustered using the DBSCAN method and finally, it generates a Weightage Participation value which provides priorities of user interest in different semantic cluster regions. The approach is evaluated through experiments and compared to existing methods. The results show that the proposed approach was able to identify significant locations and prioritize them. In my project many hidden semantic regions have been identified by considering the spatial, temporal, and semantic factors of moving objects, this knowledge can help many application areas like transportation systems for setting up the architectural platform, design of supply chain networks, preparations of travel itinerary of tourist, etc.

Contents

List of Figures

Chapter 1

Introduction

With the proliferation of positioning, telemetry, and telecommunication technologies, coupled with the widespread availability of devices that generate data on the location of an object over time, large volumes of information about moving objects are now being collected and utilized by various applications. Moving objects are those whose position changes over time, requiring a higher frequency of updates. By querying and analyzing this data, valuable insights can be gained into the patterns of mobility of these objects and the level of interest exhibited by visitors in particular geographic locations.

AN EFFICIENT PROCESSING OF SPATIO TEMPORAL AGGREGATE QUERIES uses a new algorithm called SemTraClus that helps identify, cluster, and prioritize semantic regions based on voluminous moving object data. The algorithm uses three main methods to extract significant locations called Points of Interest (POIs): stay point detection, revisited locations, and intersecting points of different trajectories. The identified POIs are then clustered using the DBSCAN method, which groups nearby points into clusters based on their density and assigns a label to each cluster. Finally, the algorithm generates a Weightage of Participation (WoP) value that provides priorities of user interest in different semantic cluster regions.

The algorithm considers both spatial and temporal features simultaneously, which makes it possible to cluster multiple trajectories for the identification of semantic points. This allows for a more detailed introduction to the pattern of mobility of the object and the interest evinced by visitors in a geographic location. The SemTraClus algorithm is significant because it provides a new approach for analyzing and understanding moving object data. By identifying and clustering POIs, the algorithm can help researchers and businesses gain more insights into user behavior, which can be used for a variety of applications such as location-based advertising, urban planning, and traffic management.

1.1 Problem Statement

The drawbacks of currently existing models are:

1. Existing models focus on processing a single trajectory at a time.
2. It is difficult to process entire data due to limitations of system resources in terms of space and time.
3. Moving objects are points that change their locations over time , which requires a higher update frequency, this has caused the database to be flooded with data from various sources.
4. Querying of moving object data is very complex therefore it can be computationally expensive.

To solve the above-stated issues, the proposed method can identify semantic points by clustering multiple trajectories and providing a prioritized list.

1.2 Objectives

1.2.1 Cluster multiple trajectories for identification of semantic points

The first objective of the project is to propose a system that clusters multiple trajectories for the identification of semantic points. The project aims to overcome the limitations of existing systems by considering both spatial and temporal features simultaneously. The proposed algorithm, SemTraClus, uses stay point detection, revisited locations, and intersecting points of different trajectories to identify significant locations, also known as points of interest (POIs), in a geographical area. The algorithm then clusters these POIs using the DBSCAN method, which is a density-based clustering algorithm. By clustering multiple trajectories, the proposed system can identify semantic points with a higher degree of accuracy.

1.2.2 Prioritized list of semantic points

The second objective of the project is to develop a system that provides a prioritized list of semantic points. Once the significant locations are identified and clustered, the proposed system assigns a Weightage of Participation value to each cluster. This value represents the degree of interest that visitors have shown in each cluster. The higher the value, the more popular the cluster is among the visitors. By assigning a Weightage of Participation value to each cluster, the proposed system can provide a prioritized list of semantic points, making it easier for users to access the most popular locations.

1.2.3 Simplify querying and analysis of moving object data

The third objective is to implement a model that simplifies the querying and analysis of moving object data. With the help of SemTraClus, the proposed system can extract and process voluminous amounts of moving object data quickly and efficiently. This means that the system can generate more insightful knowledge on the pattern of mobility of the objects and the interest evinced by visitors in a geographic location. By simplifying the querying and analysis of moving object data, the proposed system can make it easier for users to understand and utilize the data.

1.2.4 Reduce the complexity and time for processing the data

The fourth objective of the project is to propose a system that reduces the complexity and time for processing the data. By using the DBSCAN method for clustering and the Weightage of Participation value for prioritizing, the proposed system can reduce the complexity and time required for processing the data. The system can quickly identify semantic points and provide a prioritized list, making it easier for users to access the most relevant information. By reducing the complexity and time for processing the data, the proposed system can improve the efficiency and effectiveness of analyzing moving object data.

Chapter 2

Literature Survey

A literature survey, also known as a literature review, involves analyzing scholarly sources related to a particular subject. Examining the available literature, provides a comprehensive overview of the state of the field, allowing you to identify relevant theories, approaches, and gaps in the existing body of knowledge. When conducting a literature review from an audit perspective, the main focus is on evaluating the relevant literature. This process covers information that has been published in a specific field of study and sometimes includes information published within a specific time frame.

2.1 Purpose of the Literature Review

1. It gives readers easy access to research on a particular topic by selecting high-quality articles or studies that are relevant, meaningful, important, and valid and summarizing them into one complete report.

2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarize, evaluate, and compare original research in that specific area.

3. It ensures that researchers do not duplicate work that has already been done.

4. It can provide clues as to where future research is heading or recommend areas on which to focus.

5. It highlights the key findings.

6. It identifies inconsistencies, gaps, and contradictions in the literature.
7. It provides a constructive analysis of the methodologies and approaches of other researchers.

2.2 Related Works

The growth of moving object data has been exponential due to the widespread use of GPS-enabled devices and location-based services. These devices and services generate vast amounts of trajectory data, which represents the movement of objects or individuals over time. This data is highly valuable for a wide range of applications such as transportation management, urban planning, emergency response, and location-based advertising, among others. However, managing and analyzing such large-scale trajectory data requires specialized techniques and tools to address issues related to data representation, storage, indexing, clustering, and identification of mobility patterns. As the use of GPS-enabled devices and location-based services continues to grow, the demand for effective techniques for managing and analyzing trajectory data is expected to increase. The SemTraClus algorithm [1] proposes methods for identifying, clustering and prioritizing semantic regions of spatio-temporal trajectories. The algorithm extracts significant locations called Points of Interest using stay point detection, revisited locations, and intersecting points of different trajectories. These identified regions are then clustered using the DBSCAN method and a Weightage of Participation value is generated to provide priorities of user interest in different semantic cluster regions. This is the first work that clusters multiple trajectories for the identification of semantic points, considering spatial and temporal features simultaneously and providing a prioritized location list. The paper also highlights the significance of such a system in simplifying the querying and analysis of moving object data and reducing the complexity and time for processing the data. In connection with the present work, we review works related to semantic trajectories and methods used for clustering these geographical locations.

2.2.1 Moving Object Queries

Alamri et al. [2] propose a comprehensive taxonomy of various types of moving object queries in spatial databases. The authors identify several types of moving object queries, including

Spatial queries, Temporal queries, Spatiotemporal queries, Trajectory queries, and Attribute queries. The paper also discusses several subcategories within each type of query and provides examples of how these queries can be applied in real-world scenarios. The proposed taxonomy can be useful in guiding the development of spatial databases and query languages for moving object data and can help researchers and practitioners to better understand and classify different types of queries. M. A. Hernandez et al. [3] present a model for representing and querying moving objects. The authors argue that there is a need for a unified model that can handle different types of moving objects, such as vehicles, animals, and humans, and can support a range of queries, including spatial, temporal, and spatiotemporal queries. The proposed model is based on a set of primitives, such as point, line, and region, and a set of operators, such as distance, direction, and intersection, that can be used to describe the movement of objects over time. The model also includes a representation of trajectories, which is a sequence of points that describes the path of an object over time. The authors demonstrate the usefulness of their model by presenting several examples of queries that can be performed using their model, such as finding the trajectory of an object that intersects with a region, finding objects that move in a particular direction, and finding the nearest object to a given point at a specific time.

N. K. Dhar et al. [4] provide a comprehensive survey of state-of-the-art in moving object query processing. The authors discuss the challenges associated with processing queries on large-scale and high-speed moving object data and review the existing approaches for various types of moving object queries, such as range, k-NN, and spatial join queries. They also classify the existing techniques based on different criteria, such as data models, indexing structures, query processing strategies, and query languages. The paper concludes with a discussion on open research issues and future directions in the field of moving object query processing. A. K. S. Kumar et al.[5] discusses the importance of moving object databases and proposes new query languages for handling continuous and discrete queries on moving objects. The authors provide a comprehensive survey of existing techniques and propose new algorithms for processing such queries efficiently. They also discuss the challenges associated with handling moving object queries and provide solutions to overcome them. Overall, the paper contributes to the development of effective techniques for handling moving object queries in spatial databases.

2.2.2 Extracting Semantic Points

The extraction of semantic points was first introduced by Spaccapietra et al. [6] which provides a comprehensive analysis of the conceptual aspects of trajectories in the context of data and knowledge engineering. The authors, Spaccapietra et al., present a conceptual view of trajectories based on four key dimensions: spatiotemporal, thematic, contextual, and quality. The spatiotemporal dimension is concerned with the geographical and temporal aspects of trajectories, including the position, velocity, and direction of the moving objects. The thematic dimension captures the domain-specific information associated with the moving objects, such as the type of transportation or the purpose of the movement. The contextual dimension considers the environmental and situational factors that influence the trajectories, such as the weather, traffic, or social events. Finally, the quality dimension deals with the uncertainty and imprecision of trajectory data and the methods to manage and assess the quality of the data. The paper also discusses the main challenges and research opportunities related to trajectories, such as data modeling, indexing and querying, data fusion and integration, and knowledge discovery. The authors propose a conceptual framework for trajectories based on the four dimensions and outline a set of research questions and directions for future work.

Bogorny et al. [7] introduce a conceptual data model called ”CONSTAnT” for semantic trajectories of moving objects. The authors highlight the importance of understanding the semantics of the trajectories of moving objects, which can provide insights into various applications such as traffic analysis, tourism, and healthcare. The CONSTAnT model extends the existing models in the literature by incorporating semantic information along with the spatiotemporal information of the trajectories. The paper first introduces the basic concepts of the CONSTAnT model, such as trajectory, trajectory point, and semantic information. The authors then describe the various components of the model, including the trajectory component, semantic component, and annotation component. The trajectory component contains the spatio-temporal information of the trajectory, such as the location, time, and speed of the moving object. The semantic component captures the semantic information associated with the trajectory, such as the type of transportation used, the purpose of the trip, and the activity being performed. The annotation component provides additional information about the trajectory, such as the source of the data and the quality of the data. The authors also discuss the relationships between the various components of the model, such as the relationship between trajectory points and semantic information. They also demonstrate the use of the CONSTAnT model in various application domains, such as transportation analysis and tourism. Alvares et al. [8] present a model that enriches raw trajectories of moving objects with semantic geographical information. The authors argue that trajectories provide useful information on the movements of objects, but to better understand the data, it is essential to add semantic information related to the geographical context. The proposed model consists of three layers: the trajectory layer, the semantic layer, and the conceptual layer. The trajectory layer contains the raw data collected by sensors such as GPS, which captures the positions of the moving object over time. The semantic layer enriches this data with additional information such as the names of places visited, the type of transportation used, and the speed of the object. Finally, the conceptual layer is a higher-level abstraction that defines concepts related to moving objects, such as the purpose of the trip or the behavior of the object. The model uses a set of semantic rules to enrich the trajectories with semantic information. For example, a rule can specify that if a moving object stays for more than one hour in a specific area, it is considered to have visited a place of interest. The semantic layer also includes a set of spatial operators to support spatial queries, such as finding the trajectory of a moving object that passes through a specific point of interest.

Methods like DB-SMoT [9] cluster spatiotemporal data points based on their directionality. The authors argue that the direction of motion is an important feature in spatiotemporal data analysis, particularly in the context of moving objects, as it can provide insights into the behavior of these objects. They propose a clustering algorithm that takes into account the direction of motion, as well as the spatial and temporal features of the data points, to identify clusters of similar trajectories. The DB-SMoT algorithm is based on the Sequential Minimal Optimization (SMO) algorithm, which is a popular algorithm for training Support Vector Machines (SVMs). In DB-SMoT, SMO is used to find clusters of spatiotemporal points that have similar directionality, as well as similar spatial and temporal characteristics. The authors evaluate the performance of DB-SMoT using synthetic and real-world datasets and compare its performance with other clustering algorithms, including DBSCAN and ST-DBSCAN. The results show that DB-SMoT outperforms these algorithms in terms of both accuracy and efficiency, particularly in datasets with a high degree of directionality. Sajimon Abraham and Lal [10] suggest a method for measuring the similarity of network-constrained trajectories of moving objects using sequence alignment techniques. The authors propose a new metric called the Spatio-Temporal Sequence Alignment (STSA) distance, which uses a dynamic programming algorithm to align two sequences of spatial locations and times. The method is evaluated using both synthetic and real-world data, and the results show that it outperforms other similarity measures in terms of accuracy and efficiency. They also present an encoding technique for managing road network information. The paper ”Moving Object Queries”, provides a comprehensive survey of the state-of-the-art in moving object query processing. The authors discuss the challenges associated with processing queries on large-scale and high-speed moving object data and review the existing approaches for various types of moving object queries, such as range, k-NN, and spatial join queries. They also classify the existing techniques based on different criteria, such as data models, indexing structures, query processing strategies, and query languages. The paper concludes with a discussion on open research issues and future directions in the field of moving object query processing. Overall, the paper provides a useful reference for researchers and practitioners working on moving object databases and query processing.

2.2.3 Clustering of moving object data

Clustering moving object data is a process of grouping similar trajectories or objects based on their spatial and/or temporal characteristics. Clustering is an important task in moving object database management, as it can help to identify patterns and trends in the movement behavior of objects. Several clustering techniques can be used for moving object data, including density-based clustering, grid-based clustering, partition-based clustering, and model-based clustering. Kisilevich et al. [11] provide an overview of various spatiotemporal clustering methods for different types of data. The chapter starts by defining the concept of spatiotemporal clustering and its significance in various fields such as transportation, ecology, and crime analysis. The authors then discuss various spatiotemporal clustering techniques such as density-based, grid-based, and partition-based methods. They provide a detailed explanation of each method along with its advantages and disadvantages. Additionally, the authors also discuss techniques for evaluating the quality of spatiotemporal clusters. The chapter also covers some advanced topics such as clustering uncertain spatiotemporal data, handling high-dimensional data, and clustering data streams. Finally, the authors conclude by discussing some real-world applications of spatiotemporal clustering, such as traffic analysis, disease outbreak detection, and crime pattern recognition.

Density-based methods identify objects in dense regions and classify them according to the distance function which is more suitable for spatial clustering. There are various density-based clustering such as DBSCAN [12] that is designed to discover clusters of arbitrary shape in large spatial databases that may contain noise. The algorithm is based on the idea that a cluster is a dense region of points, separated by regions of lower point density. It works by defining two parameters: the radius , and the minimum number of points required to form a dense region, minPts. Points that are within distance of each other are considered to be in the same neighborhood, and if a point has at least minPts points in its neighborhood, it is considered to be a core point. Points that are not core but are within distance of a core point are considered to be in the same cluster as the core point. Points that are not core points and are not within distance of any core point are considered to be noise. The next one is ST-DBSCAN [13] is a clustering algorithm designed to handle spatial-temporal data, which is data that has both spatial and temporal attributes. The algorithm is an extension of the well-known DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, which is commonly used for spatial data clustering. The main advantage of the ST-DBSCAN algorithm is that it can handle data with varying densities and irregularly shaped clusters in both the spatial and temporal dimensions. The algorithm works by first partitioning the data into spatial neighborhoods based on a distance threshold, and then examining the temporal behavior of each neighborhood to determine if it forms a cluster. The ST-DBSCAN algorithm requires two input parameters: the spatial and temporal distance thresholds. These parameters determine how close two data points need to be to be considered part of the same cluster. The algorithm also uses a set of core points and border points to define clusters.

In OPTICS algorithm [14] is its ability to identify clusters of varying density and shape. The algorithm produces a reachability plot that shows the density of points at different distances from each other, allowing for more flexible cluster identification. The OPTICS algorithm works by creating a hierarchical ordering of the data points based on their connectivity and density. It starts by selecting an arbitrary point and identifying its neighbors within a certain radius. The algorithm then examines the density of each point and expands the cluster if the point has a sufficient number of neighbors. The ordering of the points is determined by their reachability distance, which is the minimum distance needed to reach a point with higher density.OPTICS has several advantages over other clustering algorithms. It can handle noisy data and does not require a priori knowledge of the number of clusters. It also provides a more detailed visualization of the cluster structure than other algorithms.

Many algorithms are proposed based on DBSCAN like VDBSCAN [15] which can handle data with varying densities and cluster shapes. This is achieved by introducing a parameter that controls the minimum and maximum density of clusters, allowing for more flexible cluster identification. The VDBSCAN algorithm works by first identifying core points, which are points with a minimum number of neighbors within a certain radius. The algorithm then examines the density of each point and expands the cluster if it has a sufficient number of neighbors within the density range defined by the parameter. Points that do not belong to any cluster are considered noise. The performance of VDBSCAN was evaluated using several real-world datasets and it was found to be effective in identifying clusters of varying densities and shapes. The algorithm is also computationally efficient and can handle large datasets.MDBSCAN [16] is another dynamic method for discovering density-varied clusters, which is a clustering algorithm used for analyzing spatial data. The main advantage of this method is its ability to handle datasets with varying densities and shapes, as well as its adaptability to changing densities over time. The algorithm is based on the concept of density-based clustering, where a cluster is defined as a region of high density surrounded by areas of lower density. The dynamic method works by first initializing the algorithm with a set of core points and a set of candidate points. The algorithm then computes the distances between the candidate points and the core points and assigns each candidate point to the closest core point. The core points are then updated based on the density of the points in their vicinity, and the process is repeated until the algorithm converges.

Chapter 3

Methodology

The recent advancement in positioning, telemetry, and telecommunication technologies has led to the generation of voluminous amounts of moving object data. Querying and analyzing this data can give more insightful knowledge on the pattern of the mobility of the object and the interest evinced by visitors in a geographic location. In this paper, a new algorithm called SemTraClus is introduced which helps in identifying, clustering, and prioritizing semantic regions.

The proposed model is broken into 3 sections

1. Identify ”Points of Interest” using stay points detection, revisited points, and intersecting points.

2. Clusters the points using the DBScan method

3. Assign Weightage of Participation to get a prioritized list of locations.

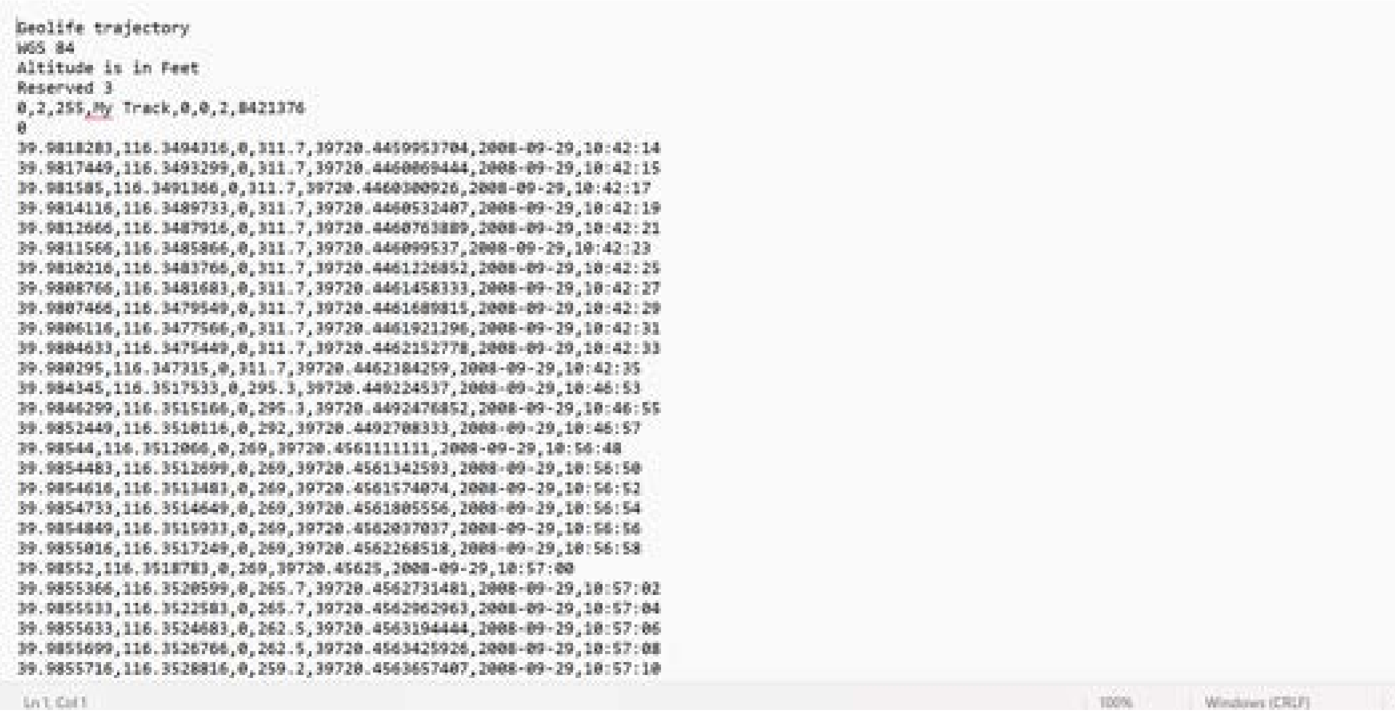
3.1 Dataset

3.1.1 Microsoft Geolife Trajectory Dataset

This GPS trajectory dataset was collected in the (Microsoft Research Asia) Geolife project by 178 users over four years (from April 2007 to October 2011). This dataset contains 17,621 trajectories with a total distance of 1,251,654 kilometers and a total duration of 48,203 hours. This dataset recorded a broad range of user's outdoor movements in Beijing which include going to work, shopping, hiking, etc. In this trajectory file, every single folder of this dataset stores a user's GPS log files, which were converted to PLT format.

PLT format:

1. Line 1…6 are useless in this dataset, and can be ignored. Points are described in the following lines, one for each line.
2. Field 1: Latitude in decimal degrees.
3. Field 2: Longitude in decimal degrees.
4. Field 3: All set to 0 for this dataset.
5. Field 4: Altitude in feet (-777 if not valid).
6. Field 5: Date - number of days (with a fractional part) that have passed since 12/30/1899.
7. Field 6: Date as a string.
8. Field 7: Time as a string.



```
Geolife trajectory
WGS 84
Altitude is in Feet
Reserved 3
0,2,255,My Track,0,0,2,8421376
0
39.9818283,116.3494316,0,311.7,39720.4459953704,2008-09-29,10:42:14
39.9817449,116.3493299,0,311.7,39720.4460069444,2008-09-29,10:42:15
39.981585,116.3491366,0,311.7,39720.4460300926,2008-09-29,10:42:17
39.9814116,116.3489733,0,311.7,39720.4460532407,2008-09-29,10:42:19
39.9812666,116.3487916,0,311.7,39720.4460763889,2008-09-29,10:42:21
39.9811566,116.3485866,0,311.7,39720.446099537,2008-09-29,10:42:23
39.9810216,116.3483766,0,311.7,39720.4461226852,2008-09-29,10:42:25
39.9808766,116.3481683,0,311.7,39720.4461458333,2008-09-29,10:42:27
39.9807466,116.3479549,0,311.7,39720.4461689815,2008-09-29,10:42:29
39.9806116,116.3477566,0,311.7,39720.4461921296,2008-09-29,10:42:31
39.9804633,116.3475449,0,311.7,39720.4462152778,2008-09-29,10:42:33
39.980295,116.347315,0,311.7,39720.4462384259,2008-09-29,10:42:35
39.984345,116.3517533,0,295.3,39720.449224537,2008-09-29,10:46:53
39.9846299,116.3515166,0,295.3,39720.4492476852,2008-09-29,10:46:55
39.9852449,116.3510116,0,292,39720.4492708333,2008-09-29,10:46:57
39.98544,116.3512066,0,269,39720.4561111111,2008-09-29,10:56:48
39.9854483,116.3512699,0,269,39720.4561342593,2008-09-29,10:56:50
39.9854616,116.3513483,0,269,39720.4561574074,2008-09-29,10:56:52
39.9854733,116.3514649,0,269,39720.4561805556,2008-09-29,10:56:54
39.9854849,116.3515933,0,269,39720.4562037037,2008-09-29,10:56:56
39.9855016,116.3517249,0,269,39720.4562268518,2008-09-29,10:56:58
39.98552,116.3518783,0,269,39720.45625,2008-09-29,10:57:00
39.9855366,116.3520599,0,265.7,39720.4562731481,2008-09-29,10:57:02
39.9855533,116.3522583,0,265.7,39720.4562962963,2008-09-29,10:57:04
39.9855633,116.3524683,0,262.5,39720.4563194444,2008-09-29,10:57:06
39.9855699,116.3526766,0,262.5,39720.4563425926,2008-09-29,10:57:08
39.9855716,116.3528816,0,259.2,39720.4563657407,2008-09-29,10:57:10
```

Figure 3.1: Microsoft Geolife Trajectory Dataset

3.2 Data Preperation

1. Analyzing the dataset: To analyze the dataset, the first step is to identify the various values in the dataset and select the column which is necessary for our algorithm. For

better understanding plot the trajectories.

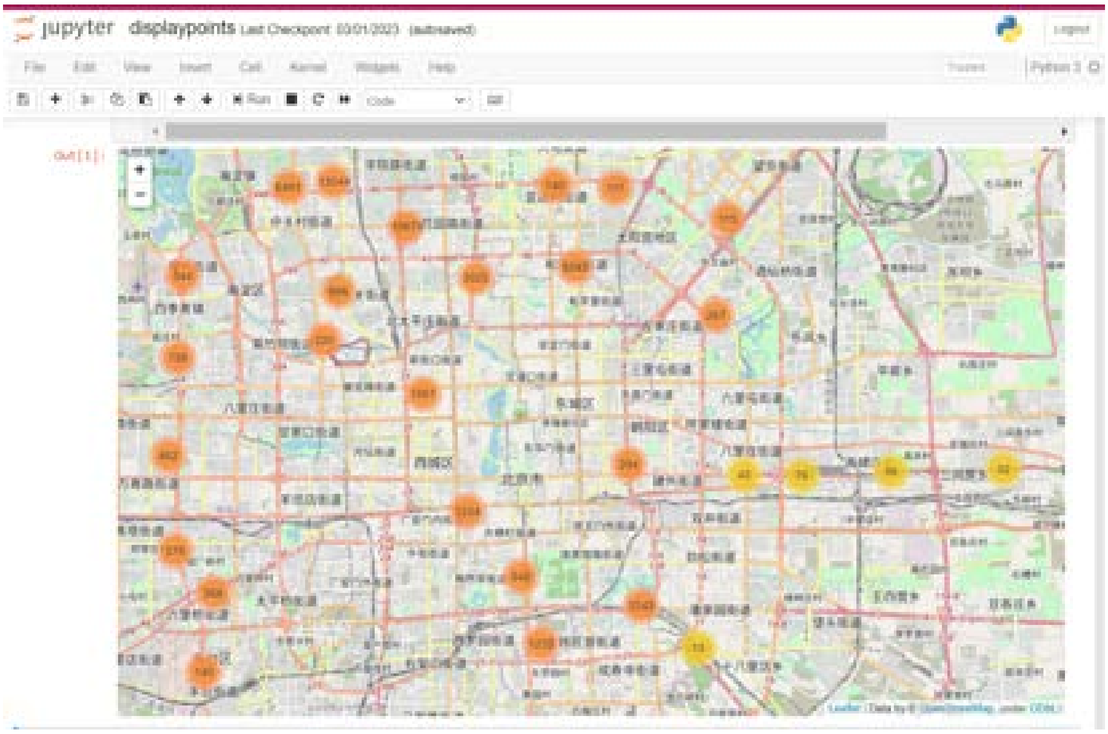

jupyter displaypoints Last Checkpoint 03/01/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Out[1]:
+
-

Figure 3.2: Trajectories

2. Removing unwanted data: The first 6 lines of the Geolife Trajectory Dataset is unwanted so we skip those lines and remove duplicate files, missing files, etc that may affect the accuracy of the analysis.

3. Identifying files with similar sizes: Geolife Trajectory Dataset consists of 181 folders (users) with multiple trajectories. Since it is hard to process the whole data, we identify files with similar sizes and group them.

4. Identifying files with stay points: Every.PLT file doesn't contain stay points, therefore we identify files with stay points using GPS analysis and use those files for further analysis.

3.3 SemTraClus Algorithm

The majority of the existing techniques for semantic point identification are focused on single trajectories. But if we choose a geographical area, we can observe that more meaningful places can be mined by accounting for different movement tracks collectively. We are bridging this gap in research with the introduction of SemTraClus.

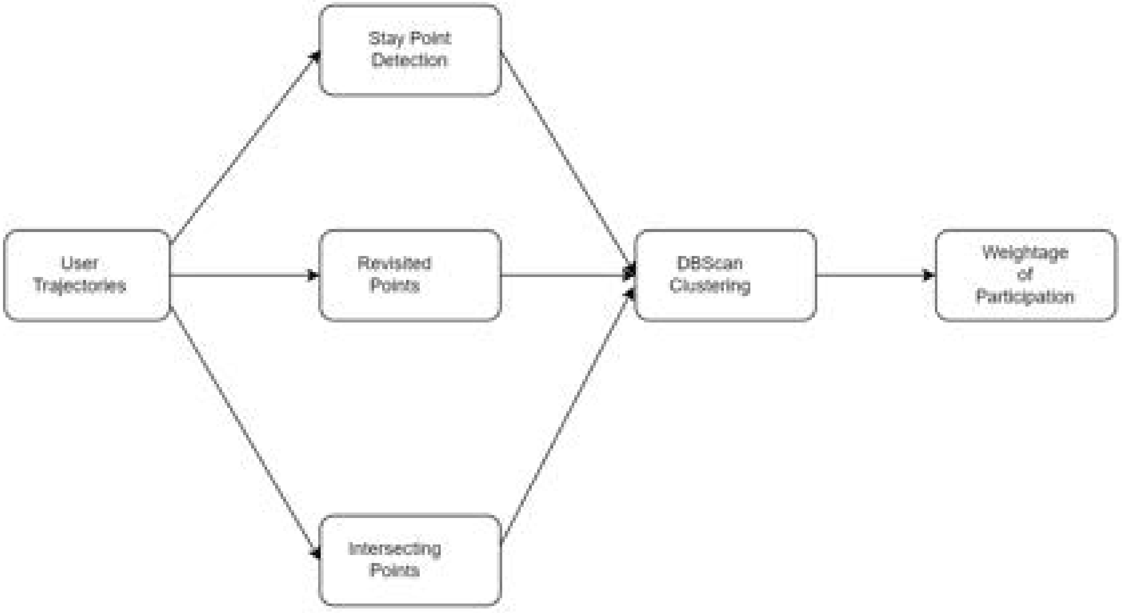


Figure 3.3: Architecture

The first phase of the algorithm excavates meaningful locations using stay points detection, revisited locations, and intersecting points of different trajectories.

3.3.1 Stay Point Detection

Stay point detection is a process of identifying specific locations where a moving object has spent a considerable amount of time being stationary. These locations are referred to as stay points or points of interest (POIs). The stay points are identified by analyzing the trajectory data of the moving object, and they represent significant locations that are meaningful for further analysis.

Stay point detection algorithms usually consider two factors to identify stay points - spatial proximity and temporal duration. In other words, a stay point is detected when a moving object stays at a location for a minimum amount of time and does not move beyond a certain distance from that location. The duration and distance thresholds can be defined based on the characteristics of the trajectory data and the application requirements.

Different algorithms are used to check if the distance between consecutive trajectory points is below a certain threshold and the total time stayed in that point is greater than a fixed duration, if so they are considered stay points.

The algorithms used are :

1. Euclidean Distance - It is a measure of the distance between two points in a two or higher-dimensional space. The Euclidean distance between two points (p1, p2) and (q1, q2) in a two-dimensional plane is calculated using the following formula:

$$d = sqrt((q1 - p1)^2 + (q2 - p2)^2)$$

2. Haversine Formula - The Haversine formula is a mathematical equation used to calculate the great-circle distance between two points on the surface of a sphere, such as the Earth. The Haversine formula is expressed as:

$$d = 2r * arcsin(sqrt(sin^2((lat2 - lat1)/2) + cos(lat1) * cos(lat2) * sin^2((lon2 - lon1)/2)))$$

3. Manhattan Distance - Manhattan distance is a distance metric used to calculate the distance between two points in a grid-like structure, such as a city block or a chessboard. The Manhattan distance between two points (x1, y1) and (x2, y2) is calculated by taking the absolute difference between their x-coordinates and their y-coordinates, and summing those differences:

$$d = |x2 - x1| + |y2 - y1|$$

After comparing the three algorithms, the Manhattan distance provided the greatest accuracy level.

3.3.2 Revisited Points

Revisited points are the locations in a trajectory where an object has visited more than once. These points are detected by comparing the stay points of a single trajectory and identifying whether any of the stay points within a particular radius have been revisited. To identify revisited points, the first step is to calculate stay points using a staying point detection algorithm. Once the stay points are identified, the next step is to check if any of the stay points have been revisited. This is done by using loops to compare the distance between each pair of stay points. The distance is calculated using the haversine formula, which takes into account the curvature of the Earth's surface. If the distance between two stay points is less than a minimum radius (which could be set based on the precision of the dataset), then the two points are considered revisited.

By identifying revisited points, it is possible to detect patterns in the movement of an object, such as frequently visited locations or routes taken repeatedly. This information can be useful in a variety of applications, such as urban planning or transportation analysis. Revisited points are the locations in a trajectory where an object has visited more than once. These points are detected by comparing the stay points of a single trajectory and identifying whether any of the stay points within a particular radius have been revisited.

To identify revisited points, the first step is to calculate stay points using a staying point detection algorithm. Once the stay points are identified, the next step is to check if any of the stay points have been revisited. This is done by using loops to compare the distance between each pair of stay points. The distance is calculated using the haversine formula, which takes into account the curvature of the Earth's surface. If the distance between two stay points is less than a minimum radius (which could be set based on the precision of the dataset), then the two points are considered revisited. By identifying revisited points, it is possible to detect patterns in the movement of an object, such as frequently visited locations or routes taken repeatedly. This information can be useful in a variety of applications, such as urban planning or transportation analysis.

3.3.3 Intersecting Points

Intersecting points are locations where multiple trajectories cross or intersect. To identify intersecting points, the SemTraClus algorithm considers multiple trajectories and their stay points. The stay points of each trajectory are identified separately using the Stay Point Detection method explained earlier.

Once the stay points are identified for all the trajectories, the SemTraClus algorithm checks the distances between stay points of different trajectories to determine if they intersect. The distance between two stay points is calculated using the Haversine formula, which is a mathematical formula used to calculate the shortest distance between two points on a sphere. If the distance between two stay points is less than a certain threshold distance, then these points are considered intersecting points. The threshold distance is a parameter that can be adjusted based on the user's needs. By considering multiple trajectories simultaneously, the SemTraClus algorithm can identify intersecting points more accurately and efficiently than other stay point detection methods.

Algorithm 1 Sematic Location Extraction

```
ProcedureGetSemanticLocations($\tau$)
foreach TR $\in \tau$ do
    $SP(x, y, duration)$ ← call CalculateStayTime(TR.List) method to calculate the stay time
    $RP(x, y, revCnt)$ ← call GetRevisitLocations(TR.List) method to calculate the revisited points
    $IP(x, y, intersectCnt)$ ← call GetIntersectLocations(TR.List) method to calculate the intersecting points
end for
$temp$ ← MEAN($SP.duration$)
$spat$ ← MIN(MODE($RP.revCnt, IP.intersectCnt$))
foreach trajectory point in $SP$ do
    if $SP$.duration $\geq$ $temp$ then
        Add trajectory point in $SP$ to SemanticPointList
    end if
end for
foreach trajectory point in $RP$ do
    if $RP$.revCnt $\geq$ $spat$ then
        Add trajectory point in $RP$ to SemanticPointList
    end if
end for
foreach trajectory point in $IP$ do
    if $IP$.intersectCnt $\geq$ $spat$ then
        Add trajectory point in $IP$ to SemanticPointList
    end if
end for
DEDUPLICATE(SemanticPointList)
return SemanticPointList
```

3.3.4 DBSCAN - Density Based Spatial Clustering of Applications with Noise

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular density-based clustering algorithm used in machine learning and data mining. The algorithm aims to identify clusters of points that are densely packed together and separated by areas of lower point density. The algorithm uses two parameters, eps, and MinPts, to define the clusters.

The eps parameter defines the maximum distance between two points that are considered to be in the same cluster. The MinPts parameter defines the minimum number of points that must be present in a cluster for it to be considered a valid cluster. Points are considered as staying points if they are in a dense region and not on the border of a cluster.

DBSCAN also identifies outliers, or noise points, which are data points that do not belong to any cluster. These points are often located in areas of low density, far away from any cluster. In our case, the algorithm was used to cluster the stay points from 3 tables using DBSCAN with eps=0.5 and minpts=4. The clustered groups and noises were represented using different colors, making it easy to visualize and interpret the results. The total number of clusters identified by the algorithm was 5.

The algorithm works by selecting an arbitrary point and checking its neighborhood to see if it has at least MinPts points within a distance of eps. If it does, the point is considered a core point, and a new cluster is created. The algorithm then expands the cluster by adding neighboring points until all points in the cluster have been identified. If a point is not a core point but is within the eps distance of a core point, it is considered a border point and added to the cluster. Finally, points that are not core or border points are considered noise points.

DBSCAN is a popular clustering algorithm because it does not require prior knowledge of the number of clusters, and it can handle clusters of different shapes and sizes. However, it is sensitive to the choice of eps and MinPts parameters and may not work well with datasets of varying densities.

Silhouette Score

Silhouette score is a measure of the quality of clustering results, and it helps to evaluate how well the data points fit into their respective clusters. The score is calculated for each data point and ranges from -1 to 1, where a score of 1 indicates good clustering and a score of -1 indicates poor clustering.

To calculate the silhouette score for a data point i, we need to consider two distances:

The average distance between point i and all other points in its cluster is denoted as a(i).

The average distance between point i and all other points in the nearest neighboring cluster, is denoted as b(i).

The silhouette score for point i is then calculated as:

$$s(i) = (b(i) - a(i))/max(a(i), b(i))$$

If a(i) is much smaller than b(i), then s(i) will be close to 1, indicating a good clustering.If a(i) is much larger than b(i), then s(i) will be close to -1, indicating poor clustering.If a(i) and b(i) are similar, then s(i) will be close to 0, indicating that point i is equally similar to both its cluster and its nearest neighboring cluster.

The overall silhouette score for the entire clustering can be calculated as the average of the silhouette scores for all data points. A higher silhouette score indicates better clustering. The silhouette score got for the clustering is 0.9897658. In summary, the silhouette score is a useful metric to evaluate the quality of clustering results and can help to determine the optimal number of clusters for a given dataset.

Algorithm 2 SemTraClus

```
Input: $\tau$ - set of raw trajectories, $WtArray$ - weights assigned for different criteria, $MinPts$ - minimum points required to form a cluster, $Eps$ - radius of the cluster
Output: $SemLocLst$ - a list of semantic locations with user priorities
// Semantic Location extraction
1. Preprocess the raw trajectories
2. $semanticPointList \leftarrow$ call GetSemanticLocations($\tau$) method to retreive the semantic points.
3. for each $semLocationSet$ do
4.     $sematicClusterList \leftarrow$ apply DBSCAN method by passing $semLocationSet, MinPts, Eps$ // Cluster with respect to given Eps and MinPts
5. end for
6. $WPList \leftarrow$ CalculateWP($sematicClusterList$)
7. $SemLocLst \leftarrow$ ConvertToSemanticLocations($WPList, WtArray$)
9. return $SemLocLst$
```

3.3.5 Weightage of Participation

The Weighted Priority (WP) metric is a measure introduced to determine the priority value of semantic regions in a geographical area. It takes into account three criteria, namely the number of individual locations in a semantic region, the total stay time of different clusters in a region, and the number of users who passed through a semantic region.

The WP of each cluster is calculated using the following equation:

$$WPcluster = W1 * Ai_{cluster} + W2 * Bi_{cluster} + W3 * Ci_{cluster}$$

Where W1, W2, and W3 are the weights assigned to the respective criteria. The value of weights vary depending on which criteria we give more priority.

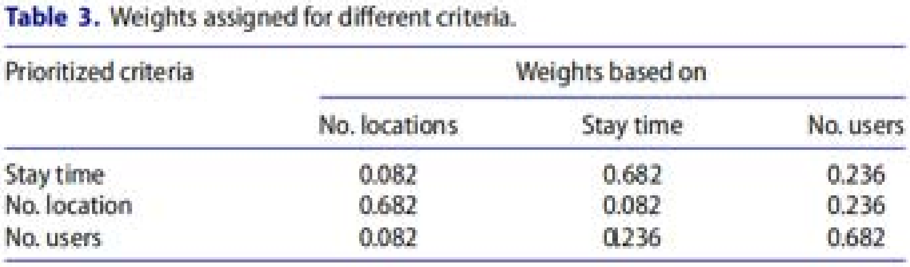


Table 3. Weights assigned for different criteria.

Prioritized criteria	Weights based on		
	No. locations	Stay time	No. users
Stay time	0.082	0.682	0.236
No. location	0.682	0.082	0.236
No. users	0.082	0.236	0.682

Figure 3.4: Weights

Ai represents the number of semantic points in each cluster divided by the total number of semantic points in all clusters. It indicates the fraction of semantic points that belong to a particular cluster.

Bi represents the total duration in each cluster divided by the total duration of all clusters. It indicates the fraction of total stay time that belongs to a particular cluster.

Ci represents the number of users in each cluster divided by the total number of users. It indicates the fraction of users that passed through a particular cluster.

The WP values for all clusters are then ranked in descending order to obtain a list of semantic regions based on their priority values. This prioritization helps in identifying the most important semantic regions in a given area and can be used in various applications such as targeted advertising, urban planning, and tourism.

Algorithm 3 WP calculation and priority assignment

```
CalculateWP ($sematicClusterList, Wt$)
/* CALCULATING WEIGHTAGE OF PARTICIPATION */
for each $Cluster$ in $sematicClusterList$ do
    for each $User$ in $Cluster$do
        calculate $Ai[cluster][user]$, $Bi[cluster][user]$ as per equation 3
    end for
end for
// WP of cluster
$SemLocLstCluster = Ai_{cluster} * Wt[0] + Bi_{cluster} * Wt[1] + Ci_{cluster} * Wt[2]$
$WPList \leftarrow TrajLstTrajectory, SemLocLstCluster$
return $WPList$
```

3.4 Software Requirements and Specifications

The software requirements for the project include:

1. Python
2. Anaconda
3. Jupyter Notebook
4. SpatioTemporal Database

3.4.1 Python

Python being an object-oriented programming language is ideally modeled for fast prototyping of complicated applications. It has interfaces to several OS system calls and libraries and is protractile to C or C++. The Python programming language is utilized by many large companies, including NASA, Google, YouTube, BitTorrent, etc. Python programming is extensively used in artificial intelligence, natural language processing, neural networks, and other cutting-edge computer science disciplines. Python is a potent language that can be used to create GUIs, create online applications, and create games. Python reading and writing are quite different from reading and writing standard English statements. Python programs must first be processed by machines since they are not written in a machine-readable language. This indicates that each time a program is executed, its interpreter reads the program's code and translates it into byte code that can be read by a computer. The quality of Python is excellent throughout. In Python, all classes, data types, functions, and methods are treated equally. Programming languages are developed to meet the needs of users and programmers for an effective tool to construct programs that have an influence on people's lives, way of life, economy, and society. By boosting productivity, improving communication, and boosting power, they help improve life. Here, Python version 3.8.5 is used.

3.4.2 Anaconda

For scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), Anaconda is a free and open-source version of the Python and R programming languages that attempt to streamline package management and deployment. Conda, a package management system, controls package versioning. For Windows, Linux, and MacOS, the Anaconda distribution provides data science packages. The Anaconda distribution includes the Virtual Environment Manager and Conda package management in addition to more than 1,500 packages. Additionally, Anaconda Navigator, a graphical user interface, is provided as a substitute for CLI, the command line interface. The management of package dependencies is a big difficulty for Python data science and the main distinction is between conda and the pip package manager. For this reason, conda was created.

3.4.3 Jupyter Notebook

The Jupyter Notebook App is a server-client program that enables the web browser-based editing and execution of notebook papers. The Jupyter Notebook App may be used locally, without an internet connection, or it can be deployed on a remote server and viewed online.

When a Jupyter Notebook App is opened, the component that is shown immediately is the Notebook Dashboard. The Notebook Dashboard is primarily used to manage the running kernels and open notebook papers (visualize and shutdown) The ”computational engine” that runs the code in a Notebook document is called a notebook kernel. Python code is run via the Python kernel. The corresponding kernel is started instantly when a The notebook document is opened. The kernel does the calculation and generates the results when the notebook is run (either cell-by-cell or through the menu Cell -¿ Run All). The kernel may use a lot of CPU and RAM, depending on the kind of calculations

3.4.4 SpatioTemporal Database

Spatio-temporal databases, also known as Moving Object Databases (MOD), are specialized databases that can handle spatiotemporal data, i.e., data that includes both location and time information. They are used to store and manage data related to moving objects, such as vehicles or people. These databases are designed to handle large amounts of data and can efficiently process queries related to the location and movement of objects over time.

The spatiotemporal data is typically represented in the form of points, lines, regions, or volumes. Points are used to represent the location of an object at a specific time, for example, a school, a building, or a person's location. Lines represent linear features, such as streets, rivers, and roads, and can be used to track the movement of objects along those features. Regions are composed of a set of points or lines that define a specific area, such as the boundary of a city or a forest. Volumes represent 3D space, and can be used to track the movement of objects in 3D, such as airplanes or drones.

MODs are designed to handle the massive amounts of data that can be generated by the tracking of moving objects over time. They use indexing techniques to optimize the storage and retrieval of data, such as R-trees and k-d trees. They also use partitioning techniques to distribute the data across multiple servers, which allows for parallel processing and reduces the response time for queries. Compression techniques can also be used to reduce the amount of data that needs to be stored, while still maintaining the accuracy of the spatiotemporal data.The use of spatiotemporal databases has become increasingly important in many applications, such as traffic management, logistics, and surveillance. In these applications, the ability to track the location and movement of objects over time can provide valuable insights into patterns and trends, which can be used to optimize operations and improve decision-making.

The spatio temporal data is stored in the form of line,point and polygon:

- A point is the simplest spatial object in a spatiotemporal database, and it is used to represent a specific location or point of interest. In a moving object database, a point could represent the location of a moving object at a specific time. For example, the point could represent the location of a vehicle or a person at a particular time.

- A line is a spatial object that is used to represent linear features such as roads, rivers, and streets. In a spatiotemporal database, a line could be used to represent the path of a moving object over time. For example, the line could represent the route taken by a vehicle or a person as they move through a city.

- A polygon is a spatial object that is used to represent an area or region. In a spatiotemporal database, a polygon could be used to represent the boundaries of a city, forest, or any other geographical region. A polygon is composed of bounding arcs and label points that define the boundary of the area.

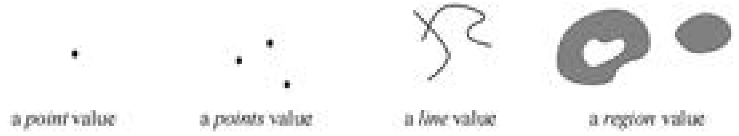


Figure 3.5: SpatioTemporal Datatypes

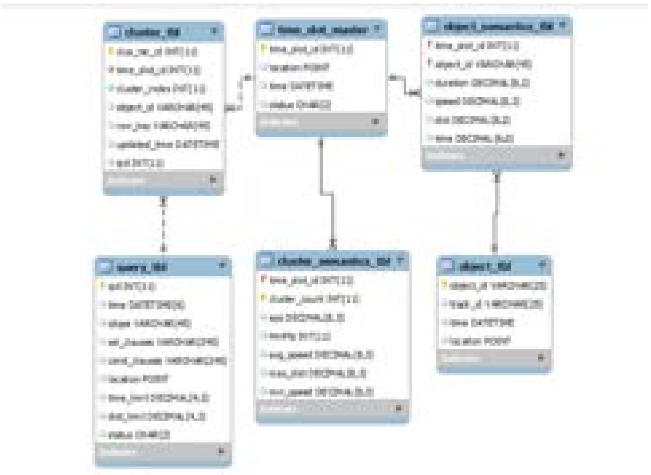


Fig. 2. The data model

Figure 3.6: Database Design

3.4.5 Hardware and experimental environment

The hardware used for this experiment includes Windows 11 Home 64-bit OS, x64-based processor, Intel(R) Core (TM) i7-8565U CPU @ 1.80GHz, 1992 Mhz, 4 Core(s), 8 Logical Processor(s), 16 GB RAM.

The experimental environment was prepared by using Python 3.10 programming language. Machine learning and deep learning libraries like - NumPy, Pandas, Matplotlib, folium, and Seaborn were also used.

Chapter 4

RESULT AND DISCUSSION

The primary quality control method used in software development is testing. Following the coding stage, testing purposes are served by running the accessible computer programs. Testing must find flaws made during the earlier phase as well as those introduced during development. So, the purpose of testing is to find program requirements, design, or coding flaws.

- A program is tested by being run to identify any errors.
- A excellent test case has the highest chance of spotting an error that hasn't been identified yet.
- A test that finds an error that hasn't been found yet is successful.

Our objective is to develop tests that systematically uncover many sorts of issues with minimal time and effort. Testing indicates that software functionalities appear to operate as expected and that performance criteria appear to have been met. The information acquired during testing is an excellent predictor of program reliability and a partial indicator of software quality as a whole. Testing has one drawback, however: it can only demonstrate the presence of software defects, not their absence.

4.1 Testing and its types used

The main task following software development is to determine whether the experimental results and the actual results agree. Testing is the process in question. It is employed to ensure that the created system is free from errors. Testing's primary purpose is to find errors and missing operations by running the software. Additionally, it makes sure that the developer satisfies all of the project's goals. Testing's objective is to determine is to identify defects in the developed software as well as ways to increase its correctness, usability, and efficiency. It seeks to gauge a software program's performance, functionality, and specification. The developed program is put through tests, and the outcomes are compared to the required documentation. Debugging is carried out when there are too many faults that have happened. After debugging, the software is once more tested to make sure there are no errors. Unit testing, integration testing, and system testing are the main testing methodologies used in this project.

- In unit testing, testing is done on each distinct piece of software. It ensures that the software's many components all function as intended. Each section of the algorithm was tested with various datasets to assure its accuracy.

- In integration testing, the integrated distinct components are examined to see whether or not the intended purpose was accomplished. It helps us find any problems that might appear after the units are combined. The algorithm is mainly divided into three sections that is identifying points of interest using three methods, clustering and weightage of participation. Once the points of interest are identified they are applied in clustering and checked if clusters were able to be formed based on the conditions.

- The entire piece of software is evaluated during system testing to make sure it meets all the requirements. Datasets are applied to the algorithm and the results are checked to find out if these locations are actually of geographical importance.

4.2 Output Screens and Results

1. Mapping Trajectories:

   The Geolife Trajectory Dataset consists of trajectories from 181 users, each containing a collection of spatial points representing the movement patterns of individuals over time. To visualize and analyze these trajectories, the Folium library is utilized, which provides a powerful tool for creating interactive and customizable maps. Additionally, Folium allows for interactivity, allowing users to zoom in and out, pan across the map, and click on individual points or trajectories to retrieve more information or perform further analysis. This interactive aspect enhances the exploratory capabilities of the geospatial data, enabling users to gain insights and understand the patterns and characteristics of the trajectories more effectively. With the combination of the Geolife Trajectory Dataset and the Folium library, researchers and analysts can not only visualize the trajectories but also conduct advanced analyses, such as clustering trajectories, identifying significant locations, or detecting anomalies in movement patterns.

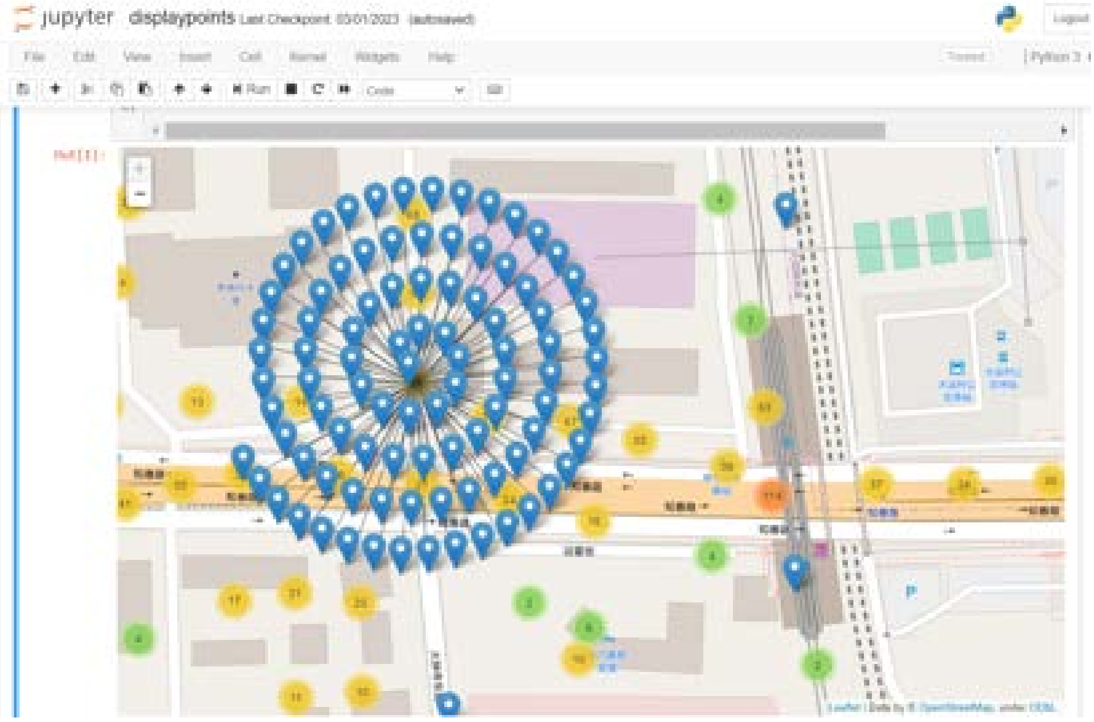
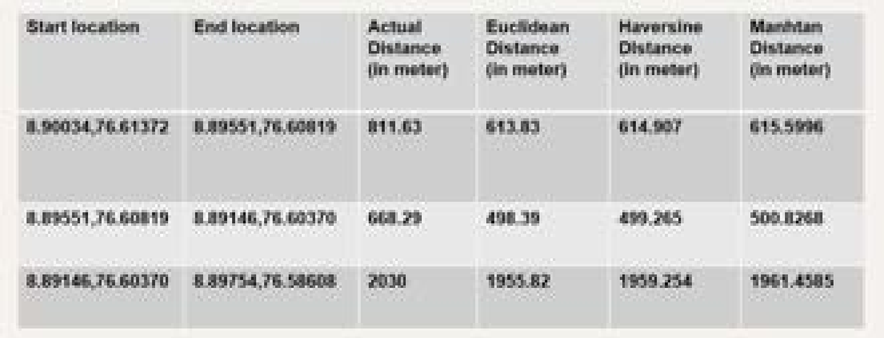


Figure 4.1: Trajectories

2. Distance formula comparison

   To assess the accuracy of different distance formulas in the context of the dataset, the Manhattan, Euclidean, and Haversine distance formulas are compared. These distance formulas provide a means to quantify the spatial proximity or dissimilarity between points in the dataset. By applying these distance formulas to the dataset, the distances calculated using each formula are compared with the actual ground truth distances. This comparison allows for the assessment of which distance formula provides a more accurate representation of the spatial relationships between the points in the dataset.



Start location	End location	Actual Distance (in meter)	Euclidean Distance (in meter)	Haversine Distance (in meter)	Manhtan Distance (in meter)
8.90034,76.61372	8.89551,76.60819	811.63	613.83	614.907	615.5996
8.89551,76.60819	8.89146,76.60370	668.29	498.39	499.265	500.8268
8.89146,76.60370	8.89754,76.58608	2030	1955.82	1959.254	1961.4585

Figure 4.2: Distance Formula

3. Stay Point Detection

The identification of stay points within the dataset of user trajectories involves setting time and distance thresholds to determine significant stationary locations. Once these stay points are identified, the Folium library is employed to map and visualize them interactively and dynamically. To identify stay points, a time threshold is defined to determine the minimum duration a user must spend within a certain radius to be considered stationary. This threshold helps filter out transient stops and focus on meaningful locations where users spend a significant amount of time. Additionally, a distance threshold is set to specify the maximum distance a user can travel within a given timeframe to still be considered stationary. This threshold accounts for minor movements within a location and ensures that stay points capture true stationary periods.

Using Folium, the identified stay points can be plotted on a map, visualizing their spatial distribution. Each stay point is represented by a marker, allowing for easy identification and exploration of the significant locations within the dataset

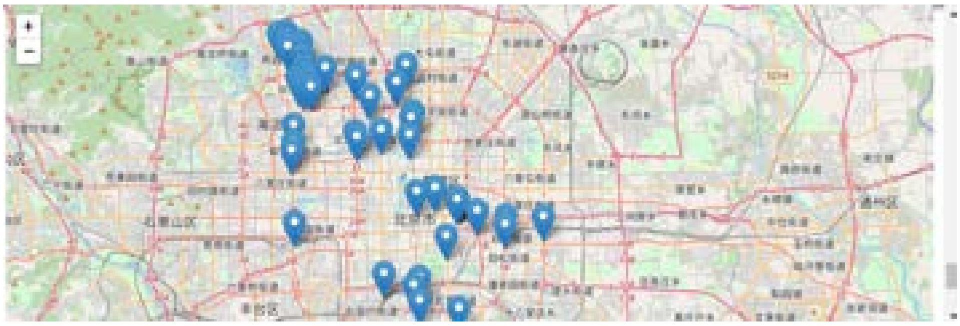


Figure 4.3: Stay Point

4. Revisited Points

   To check if stay points are revisited, several methods can be employed. One common approach is to compare the coordinates of each stay point with the coordinates of all other stay points for the same user. By setting an appropriate distance threshold, we can account for slight variations in the geographic coordinates due to factors like GPS accuracy or user mobility patterns. If the distance between two stay points is within this threshold, it suggests that the user has visited the same or a very close location on different occasions. Mapping the revisited locations using Folium allows for a visual representation of these points on an interactive map. By plotting the revisited locations along with the stay points, researchers can easily distinguish between one-time visits and repeated visits.

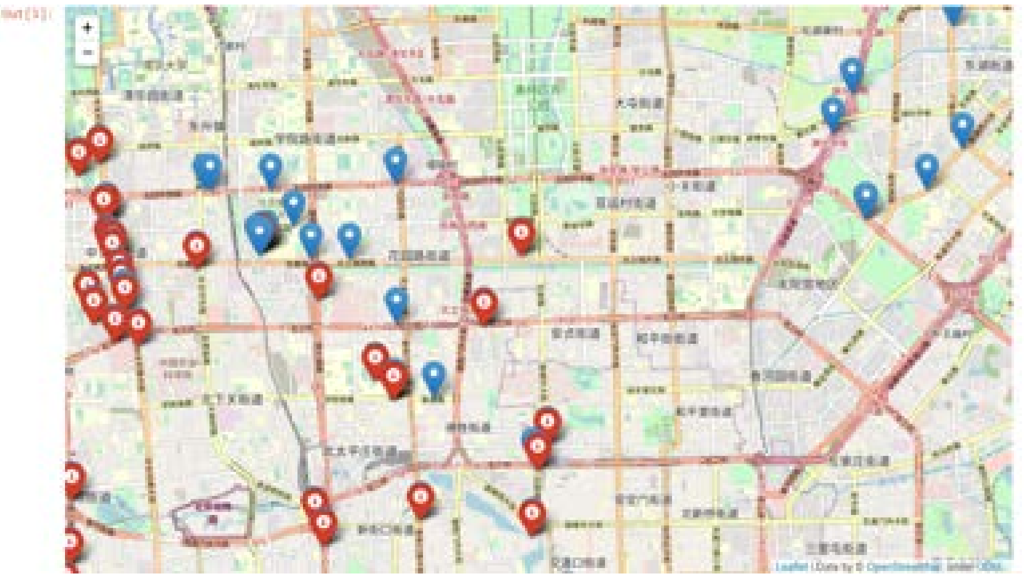


Figure 4.4: Revisited Points

5. Intersecting Points

   After identifying the stay points for different users, the next step is to determine if these stay points intersect or overlap with each other. Identifying intersecting points can provide valuable insights into areas where multiple users spend time simultaneously, indicating potential points of interest or areas of social interaction. To check for intersection among stay points, a spatial analysis approach is applied. This involves comparing the geographic coordinates of each stay point for one user with the coordinates of all stay points for other users. If there is a spatial overlap within a certain distance threshold, it suggests that multiple users have stay points near each other. Once intersecting points are identified, they can be mapped using Folium.

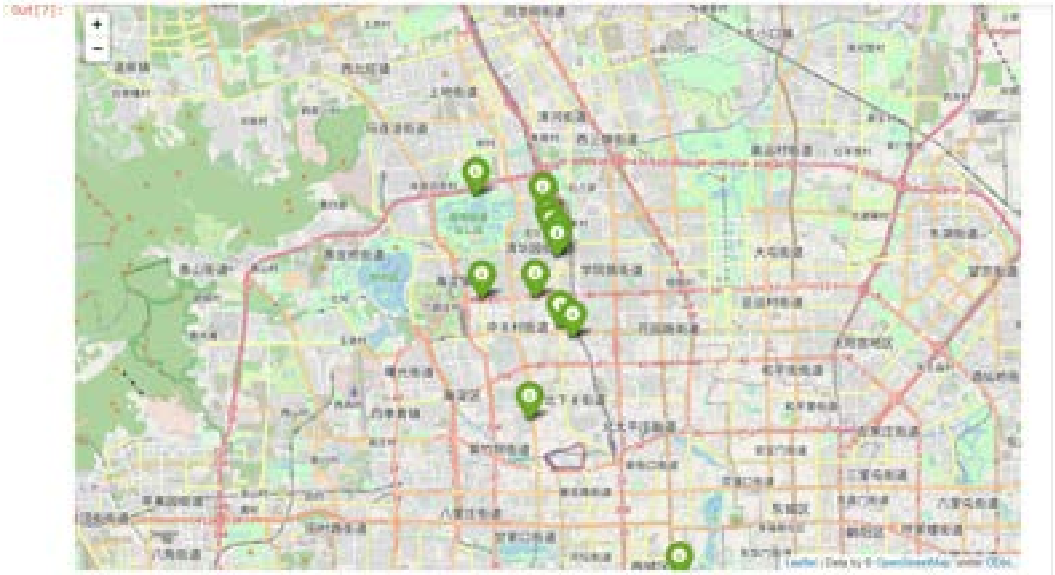


Figure 4.5: Intersecting Points

6. DBSCAN Clustering

   To further analyze and group the identified points, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is utilized. DBSCAN is a popular density-based clustering algorithm that groups points based on their density and proximity to each other. By applying the DBSCAN algorithm to the identified points, the stay points are clustered based on their spatial density and proximity. The algorithm takes into account both the spatial and temporal characteristics of the points, allowing for the discovery of meaningful groups within the dataset. The resulting clusters can vary in size, shape, and density based on the underlying distribution of the stay points. Each cluster represents a distinct group of points that are densely connected, indicating similar spatial patterns or user behaviors.

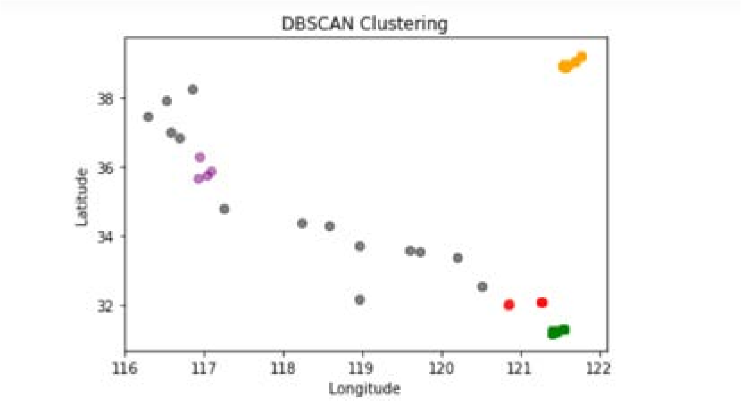
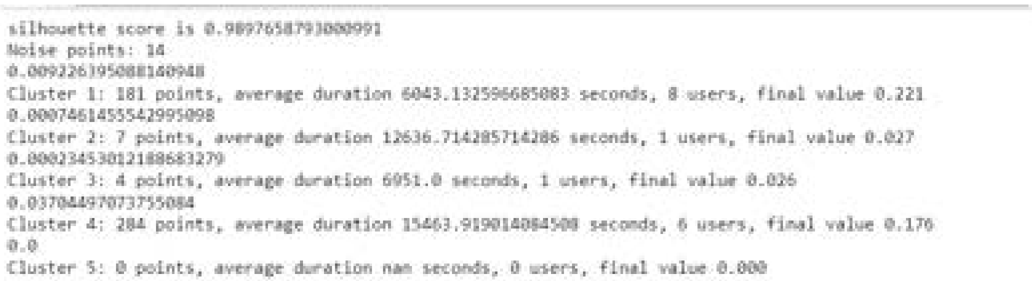


Figure 4.6: DBSCAN

7. Weightage of Participation and Silhouette score

   After clustering the identified points using DBSCAN, further analysis is performed to evaluate the quality and similarity of the resulting clusters. Two important measures used in this analysis are the Weightage of Participation and the Silhouette Score.The Weightage of Participation is a metric calculated for each cluster, indicating the level of importance or relevance of the cluster based on user preferences or interests. This metric provides a weighted value that reflects the degree of user engagement or significance within each cluster. The calculation of the Weightage of Participation can consider various factors such as user frequency, duration of stay, or specific user preferences.

   In addition to the Weightage of Participation, the Silhouette Score is calculated to assess the internal cohesion and similarity of points within each cluster. The Silhouette Score measures how close each point is to the other points within its cluster compared to points in neighboring clusters. It provides an estimate of the compactness and separation of the clusters. A higher Silhouette Score indicates that points within a cluster are similar to each other and well-separated from points in other clusters. This suggests that clustering is effective in capturing meaningful patterns and grouping similar points.



```
silhouette score is 0.9897658793000991
Noise points: 14
0.009226395088140948
Cluster 1: 181 points, average duration 6043.132596685083 seconds, 8 users, final value 0.221
0.0007461455542995098
Cluster 2: 7 points, average duration 12636.714285714286 seconds, 1 users, final value 0.027
0.00023453012188683279
Cluster 3: 4 points, average duration 6951.0 seconds, 1 users, final value 0.026
0.03704497073755084
Cluster 4: 284 points, average duration 15463.919014084508 seconds, 6 users, final value 0.176
0.0
Cluster 5: 0 points, average duration nan seconds, 0 users, final value 0.000
```

Figure 4.7: Weightage of Participation and Silhouette score

8. Cluster with highest WP Score

   After calculating the Weightage of Participation (WP) for each cluster, the cluster with the highest WP value represents the geographical location with the most demand or importance according to user preferences. To showcase this cluster on a map, the Folium library can be utilized. First, the cluster with the highest WP value is identified based on the calculated WP values for each cluster. This cluster is considered the focal point of interest due to its high user demand or relevance. Next, the geographic coordinates of the stay points within this cluster are extracted. These coordinates represent the spatial locations that contribute to the high WP value. Using Folium, a map can be generated where the identified cluster is visualized. The Folium library allows for the customization of map styles, zoom levels, and markers, providing an interactive and visually appealing representation.

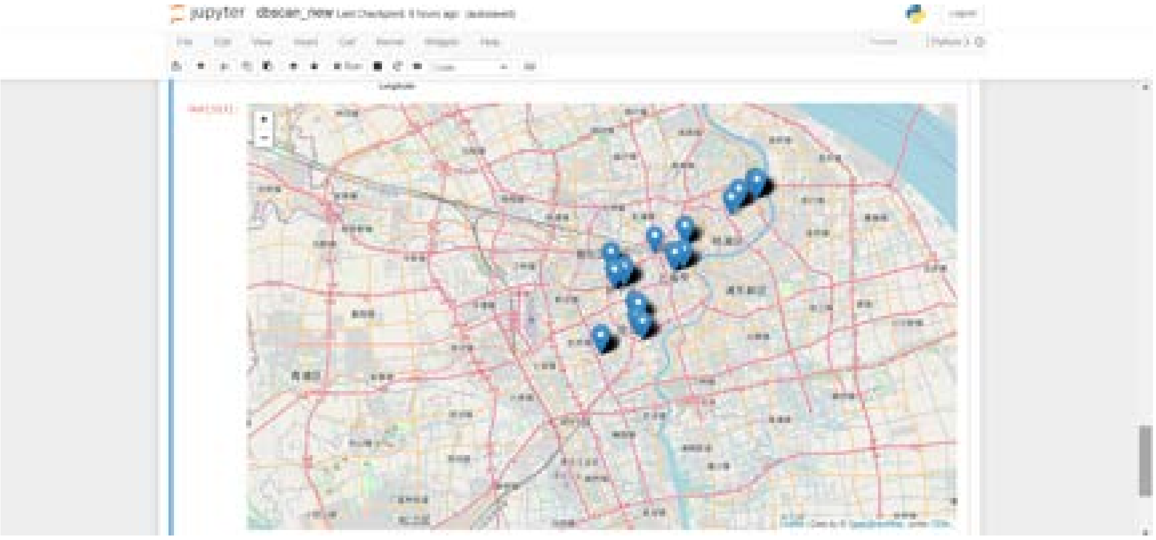


Figure 4.8: Cluster with highest WP score

9. Comparison between stay point and SemTraClus

   When comparing the number of points of interest (POIs) retrieved using a simple stay point detection method and SemTraClus, there can be significant differences in the results. In a simple stay-point detection method, the identification of POIs is usually based on basic criteria such as a minimum time threshold and a radius around each point. This method may not consider semantic factors or the clustering of stay points. On the other hand, SemTraClus incorporates additional factors such as semantic information, spatial clustering, and user-identifying POIs using stay point detection, revisited points, and, intersecting points of different trajectories. The difference in the number of POIs identified is plotted using a bar graph.

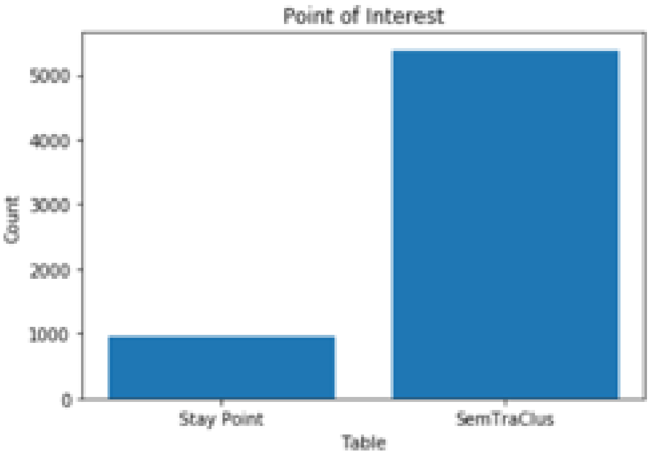


Figure 4.9: Comparison of Stay Point Detection and SemTraClus

10. Weightage of Participation (Location)

    Calculate the Weightage of Participation (WP) for each cluster by giving location criteria more priority, the next step is to visualize these values to identify the cluster with the highest participation and prioritize locations accordingly. One effective way to represent the WP values is by using a bar graph. In the bar graph, each cluster is represented by a bar, and the height of the bar corresponds to its WP value. The x-axis of the graph represents the cluster labels or identifiers, while the y-axis represents the WP values. The bar graph provides a clear visual comparison of the participation values for each cluster.

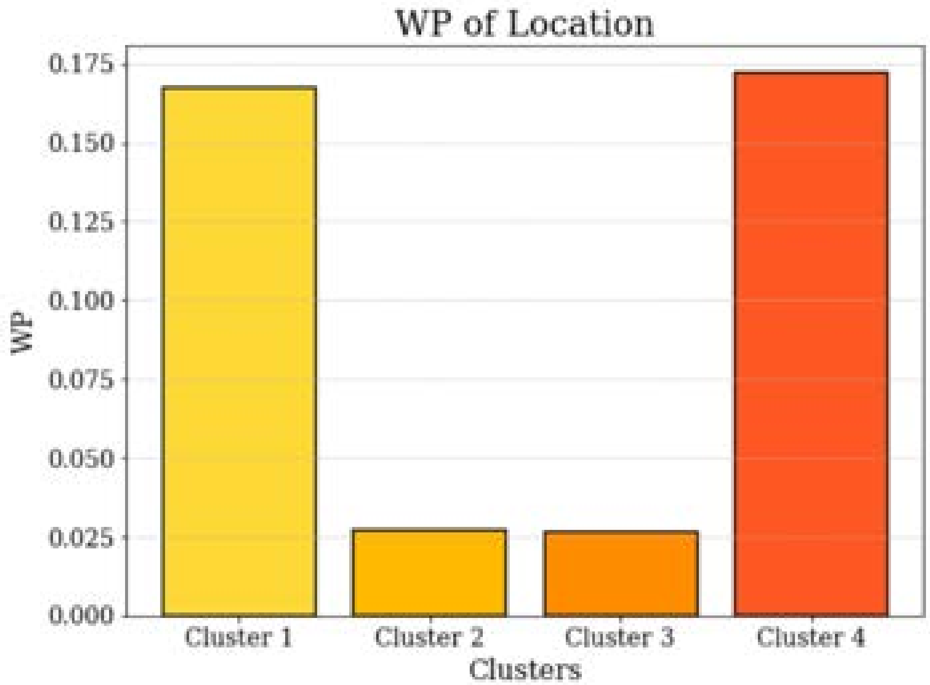


Figure 4.10: WP of clusters (location criteria)

11. Weightage of Participation (Duration)

    Calculate the Weightage of Participation (WP) for each cluster by giving duration criteria more priority, the next step is to visualize these values to identify the cluster with the highest participation and prioritize locations accordingly. One effective way to represent the WP values is by using a bar graph. In the bar graph, each cluster is represented by a bar, and the height of the bar corresponds to its WP value. The x-axis of the graph represents the cluster labels or identifiers, while the y-axis represents the WP values. The bar graph provides a clear visual comparison of the participation values for each cluster.

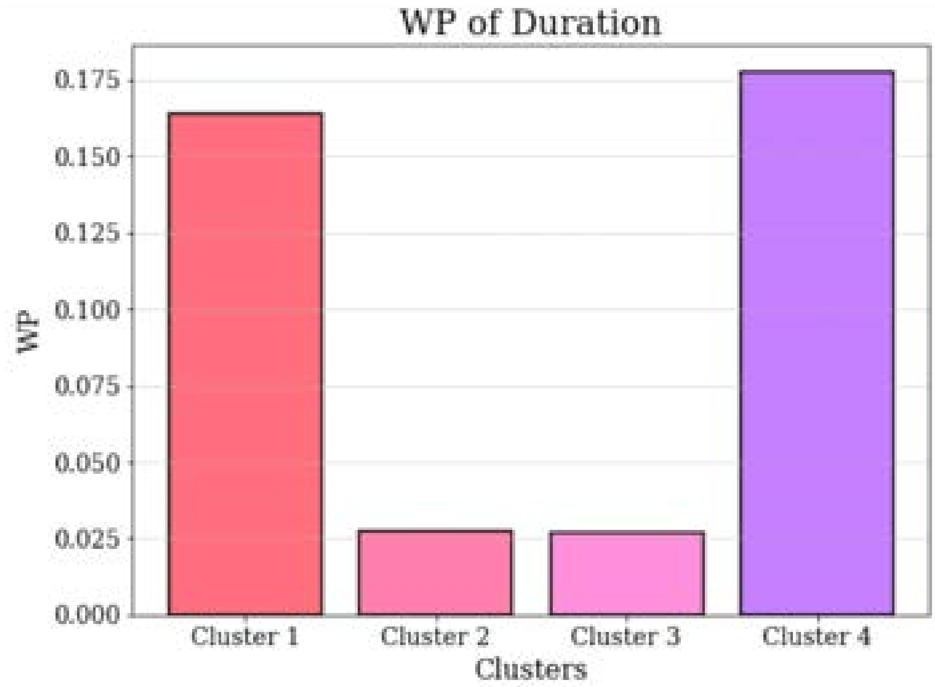


Figure 4.11: WP of clusters (Duration criteria)

12. Weightage of Participation (Users)

    Calculate the Weightage of Participation (WP) for each cluster by giving users criteria more priority, the next step is to visualize these values to identify the cluster with the highest participation and prioritize locations accordingly. One effective way to represent the WP values is by using a bar graph. In the bar graph, each cluster is represented by a bar, and the height of the bar corresponds to its WP value. The x-axis of the graph represents the cluster labels or identifiers, while the y-axis represents the WP values. The bar graph provides a clear visual comparison of the participation values for each cluster.

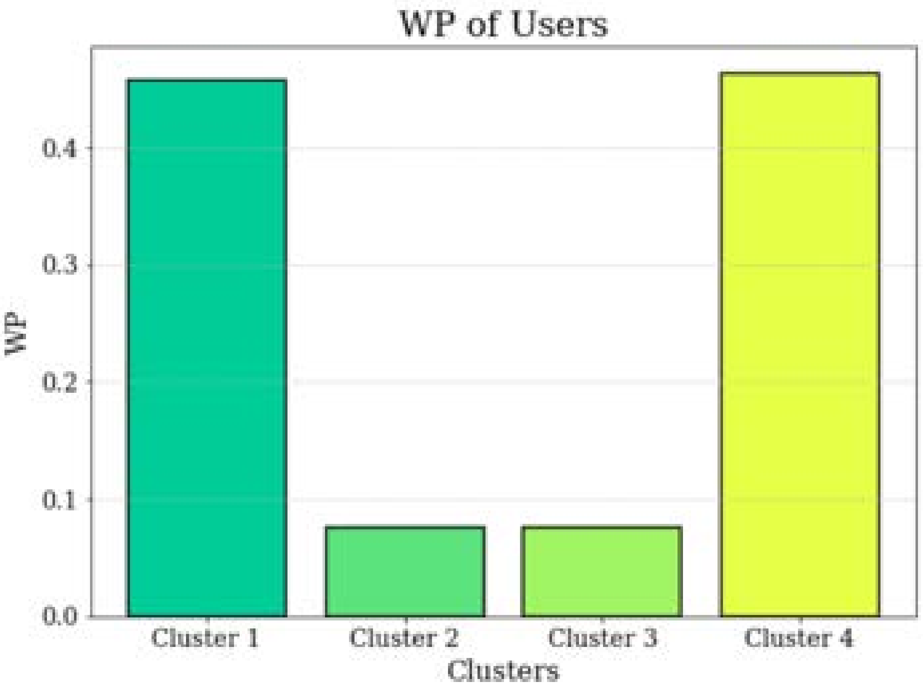


Figure 4.12: WP of clusters (User criteria)

Chapter 5

CONCLUSION

Clustering and querying of moving object data are gaining more momentum as an area of research due to the generation of huge spatial-temporal data from a wide range of location sensing devices. Analyzing and querying this data can provide insightful knowledge on the mobility patterns of objects and the interest shown by visitors in geographic locations. This paper introduces a new algorithm called SemTraClus that identifies, clusters, and prioritizes semantic regions. The algorithm has been tested on real-world data sets, and the results show that SemTraClus outperforms other state-of-the-art clustering algorithms in terms of accuracy, efficiency, and scalability. The identified semantic regions can provide valuable insights into the mobility patterns and interests of individuals in geographic locations.

The ability to analyze and understand the movement patterns and interests of individuals can provide significant benefits to businesses and researchers. For example, businesses can use this information to offer personalized and targeted experiences to their customers based on their interests and preferences. Researchers can use this information to study the behavior and preferences of individuals in specific geographic locations, such as tourist attractions, shopping centers, or urban environments.

In conclusion, the SemTraClus algorithm provides a valuable tool for clustering and querying spatiotemporal data, enabling businesses and researchers to gain deeper insights into the mobility patterns and interests of individuals in geographic locations. This can lead to better decision-making and personalized experiences for users.

5.1 Future Enhancement

The system is designed in such a way that addition of new modules can be done without much difficulty. In order to make the system as versatile and user-friendly as possible, the advanced characteristics of this technology were taken into consideration.Some of the features that can be added in future are:

1. Activity Recognition: You could use machine learning techniques to classify the type of activity (e.g., walking, driving, working, etc.) that users engage in while they are at each stay point. This information could be used to provide more personalized recommendations or targeted advertising.

2. Time-Based Analysis: You could analyze patterns of behavior over time to identify trends or changes in user behavior. For example, you could identify seasonal changes in the popularity of certain stay points or revisited points.

3. Sentiment Analysis: You could use natural language processing techniques to analyze social media posts or other user-generated content related to stay points or important areas. This could provide insight into how users feel about these locations, and help you identify areas for improvement or potential marketing opportunities.

REFERENCES

[1] Nishad, A. and Abraham, S., 2021. SemTraClus: an algorithm for clustering and prioritizing semantic regions of spatio-temporal trajectories. International Journal of Computers and Applications, 43(8), pp.841-850.

[2] Alamri, S., Taniar, D. and Safar, M., 2014. A taxonomy for moving object queries in spatial databases. Future Generation Computer Systems, 37, pp.232-242

[3] M. A. Hernandez et al.: A Foundation for Representing and Querying Moving Objects :IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 6, pp. 1493-1517, 2003.

[4] ”Moving Object Queries: A Survey” by N. K. Dhar and D. K. Nehab, IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 7, pp. 936-957, July 2007.

[5] Continuous and Discrete Moving Object Queries in Spatial Databases” by A. K. S. Kumar, P. S. S. Kumar, and J. L. Harmon, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 39, no. 2, pp. 267-280, March 2009.

[6] Spaccapietra S, Parent C, Damiani ML, et al. A conceptual view on trajectories. Data Knowl Eng. 2008;65(1):126–146.

[7] Bogorny V, Renso C, Aquino AR, et al. Constant – a conceptual datamodel for semantic trajectories of moving objects. Trans GIS. 2014;18(1): 66–88.

[8] Alvares LO, Bogorny V, Kuijpers B, et al. A model for enriching trajectories with semantic geographical information. In: Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems. ACM; 2007. p. 22

[9] Rocha, J.A.M., Times, V.C., Oliveira, G., Alvares, L.O. and Bogorny, V., 2010, July. DB-SMoT: A direction-based spatio-temporal clustering method. In 2010 5th IEEE international conference intelligent systems (pp. 114-119). IEEE.

[10] Abraham S, Lal PS. Spatio-temporal similarity of network-constrained moving object trajectories using sequence alignment of travel locations. Transp Res Part C: Emerg Technol. 2012;23:109–123.

[11] Kisilevich S, Mansmann F, Nanni M. Spatio-temporal clustering. In: Data mining and knowledge discovery handbook. Springer, Boston, MA, 2009.855–874.

[12] Ester M, Kriegel H-P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96. 1996. p. 226–231.

[13] irant D, Kut A. ST-DBSCAN: an algorithm for clustering spatial–temporal data. Data Knowl Eng. 2007;60(1):208–221.

[14] Ankerst M, Breunig MM, Kriegel H-P, et al. OPTICS: ordering points to identify the clustering structure. In: ACM Sigmod Record, vol. 28. ACM;1999. p. 49–60.

[15] Liu P, Zhou D, Wu N. VDBSCAN: varied density based spatial clustering of applications with noise. In: 2007 International Conference on Service Systems and Service Management. IEEE; 2007. p. 1–4

[16] Elbatta MT, Ashour WM. A dynamic method for discovering density varied clusters. Int J Signal Process Image Process Pattern Recogn. 2013;6(1):123–134.

APPENDIX

Screenshots

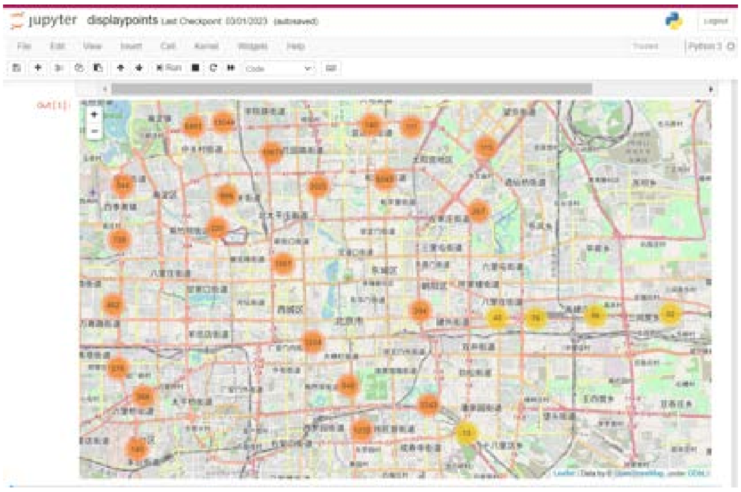


Figure A.1: Folium Map

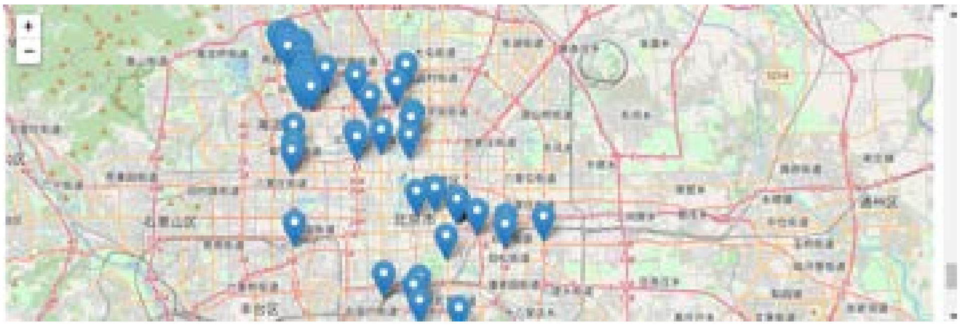


Figure A.2: Stay Point

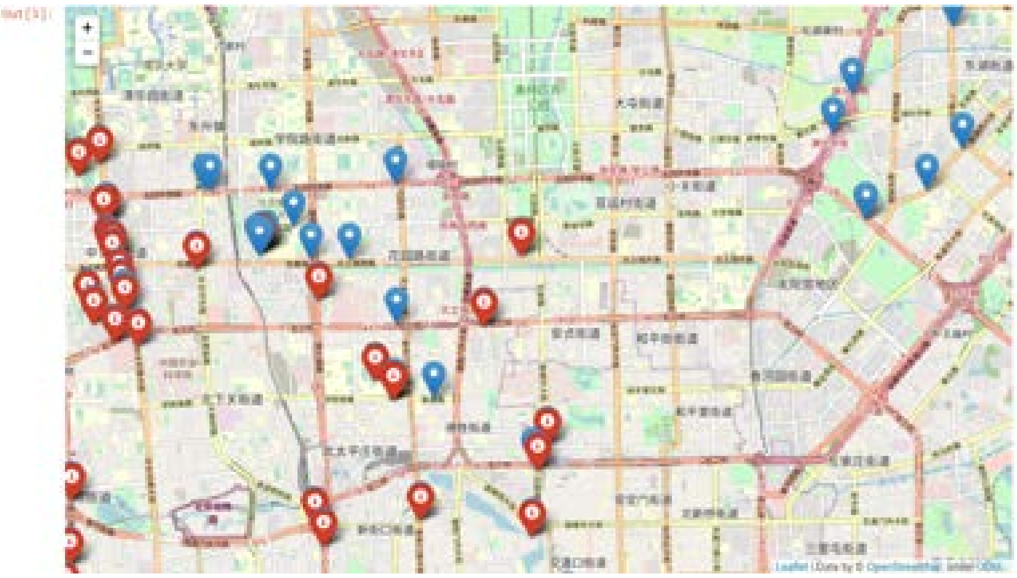


Figure A.3: Revisited Points

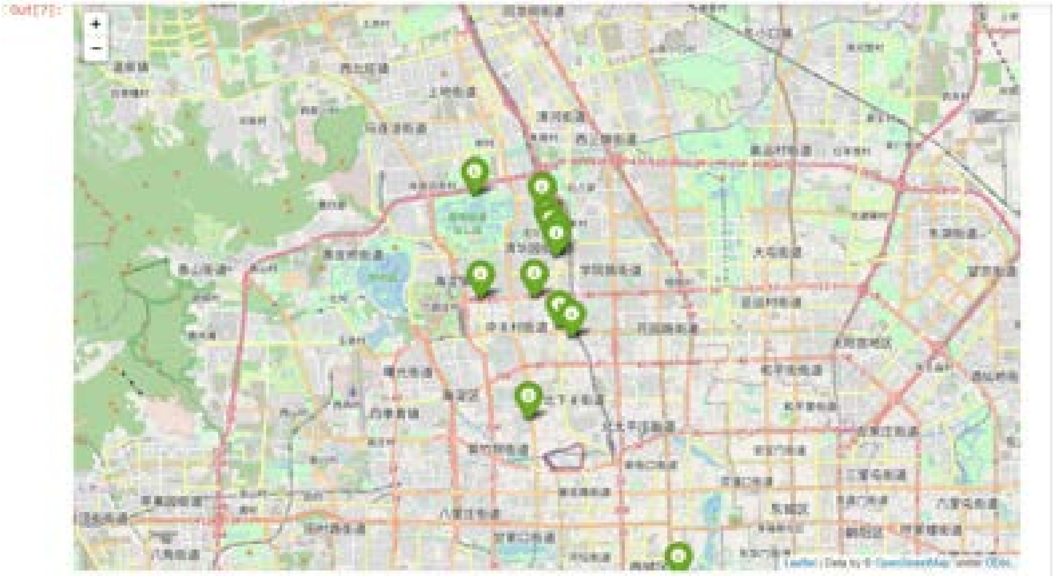


Figure A.4: Intersecting Points

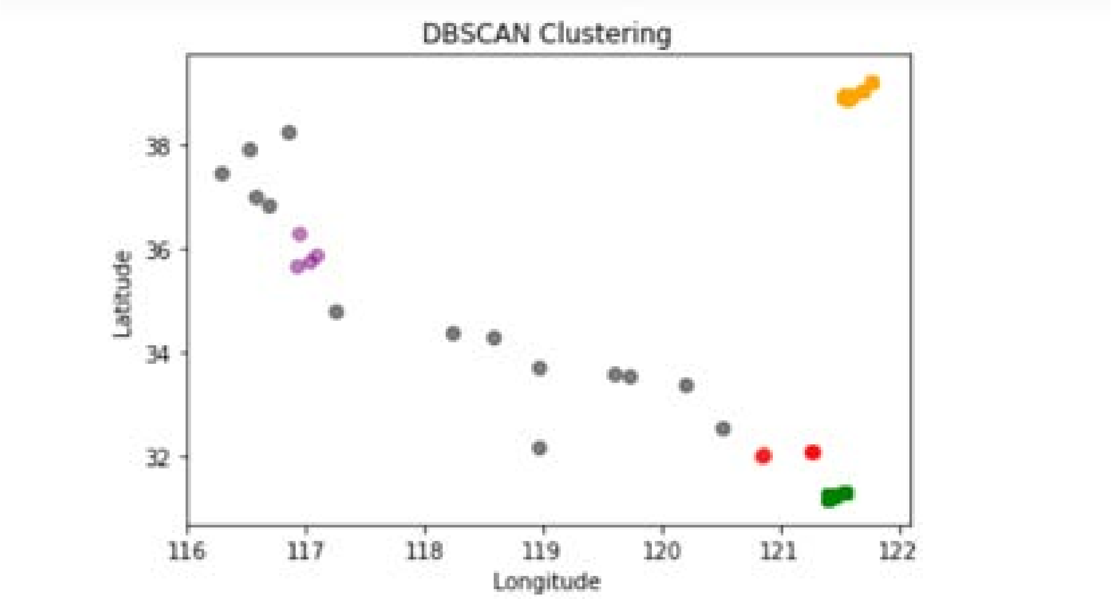


Figure A.5: DBSCAN Clusters

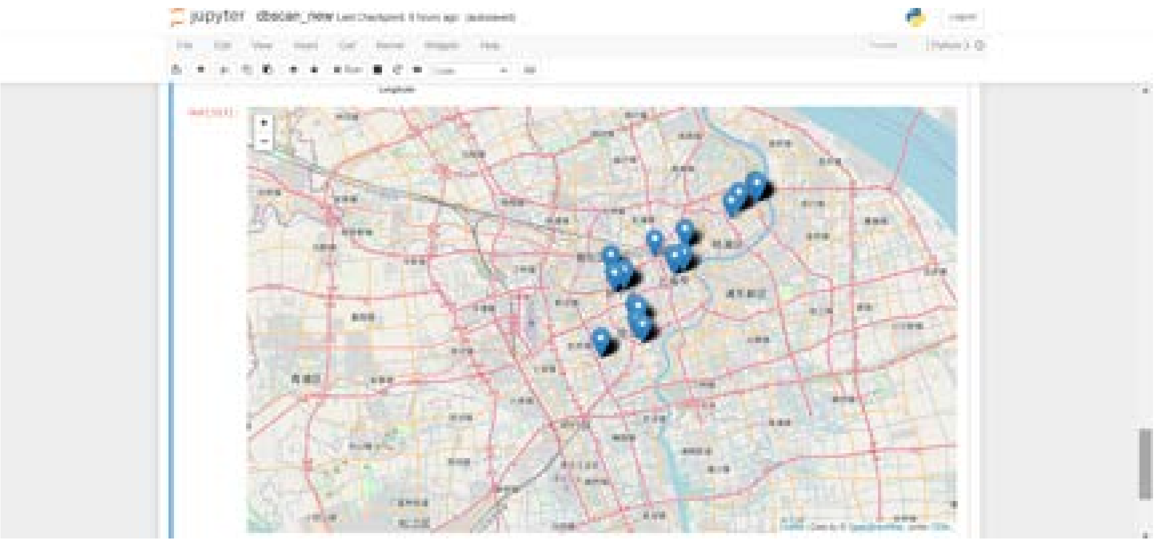


Figure A.6: Cluster with highest priority