Detecting Points of Interest in a City from Taxi GPS with Adaptive DBSCAN

Uriwan Angkhawey
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand
uriwan.a@student.chula.ac.th

Veera Muangsin
CU Big Data Analytics and IoT Center (CUBIC)
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand
veera.m@chula.ac.th

***Abstract*—Points of Interest (POIs) are popular places that create activities and transportation demands in a city. As cities constantly evolve, keeping an up-to-date map of POIs is difficult. However, by observing digital footprints of people, including taxi pick-up and drop-off locations, we can detect hotspots of activities that indicate POI locations by using techniques such as point clustering. DBSCAN is a popular density-based clustering algorithm for geographic data points. However, the effectiveness of the algorithm relies on determining appropriate parameters that match the distribution pattern of data points. That is hard because pick-up and drop-off locations are distributed differently in different areas. This paper proposes a method to automatically determine the parameters of DBSCAN according to point distribution in each dataset. The algorithm has been applied to pick-up and drop-off locations of taxis in Bangkok. The experimental results show that the proposed method can effectively discovered POIs in areas with different distribution patterns.**

Keywords—Taxi GPS Data, DBSCAN, Clustering, Point of Interest

I. INTRODUCTION

Points of Interest (POIs) are landmarks and popular places such as transit stations, shopping malls, travel destinations, government and business offices. They are centers of activities and transportation demands. In short, POIs are places that should be put on a map. As cities constantly evolve, making an up-to-date map of POIs is difficult. However, by observing digital footprints of people, we can detect hotspots of activities that indicate POI locations. Nowadays, taxis are equipped with GPS (Global Positioning System) device for real-time vehicle tracking. Taxi GPS data containing Latitude, Longitude, Speed, Date and Time, and Occupancy Status is widely used in transportation and human mobility research [1]. Taxi trips data is a digital footprint data that reveals origins, destinations, routes and time that are related to road networks, land use and geodemographic of the city. Therefore, it can be used to detect POIs that correspond to real activities of people. Analysis of POIs is useful in many fields, including Urban planning, Crime analysis, Epidemiological surveillance planning, and Marketing planning.

Detecting POIs from taxi GPS data can be done by finding clusters of pick-up and drop-off locations that are surrounding POIs. There are several clustering algorithms but DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular algorithm that has the ability to find arbitrary shaped clusters that exclude outlier points (noise) [2]. However, the effectiveness of the algorithm relies on determining appropriate parameters, namely, epsilon that is the distance between neighbor points in a cluster, and *MinPts* that is the minimum number of points to form a cluster. Both parameters should match the distribution pattern of data points. In case of the taxi dataset, determining the parameters is hard because pick-up and drop-off locations are distributed differently in different areas.

Therefore, in this paper we propose a method to automatically determine both parameters for DBSCAN clustering algorithm by considering the density distribution of each dataset. The method has been applied to taxi pick-up/drop-off locations in Bangkok to detect Points of Interest.

The remainder of this paper is organized as follows. Section 2 briefly introduces related works on DBSCAN algorithm to describe various methods to select parameters. Section 3 presents the proposed method to determine the parameters. Section 4 provides the experimental results of the proposed method along with its comparison with DMDBSCAN. Finally, Section 5 draws conclusions and plans of future research.

II. RELATED WORKS

The Density-Based Spatial Clustering of Application with Noise (DBSCAN) algorithm [2] is one of the most popular clustering algorithms. DBSCAN focuses on the density of points by defining a cluster as a group of points that have their neighbors within a specified distance, and each cluster must have core points with neighbors of at least a specified number. Points that are not included in any cluster are considered as noise or outliers. The distinguishing advantages of DBSCAN are the ability to find arbitrarily shaped clusters and noises exclusion. From the definition, DBSCAN requires two parameters. Firstly, the distance between neighbor points in a cluster, i.e. *Eps* (epsilon). Secondly, the minimum number of neighbor points around core points to form a cluster, i.e. *MinPts*. The algorithm starts with defining core points which have the number of points greater than or equal to *MinPts* by setting every point as unmasked. Point sets as marked and assigns cluster when any point within the radius of the *Eps* is reachable from core points. Then, the algorithm surveys every point in *Eps* and repeats to find the next core point to create a cluster until any point cannot allocate to the cluster. After processing all points, a point is not assigned to a cluster will be

978-1-5386-7805-3/18/$31.00 ©2018 IEEE

considered as noise. DBSCAN can be grouped efficiently with a single density data set. However, there are issues with different densities.

DBSCAN is quite sensitive to the parameter choices. In [7], the authors suggest using small *MinPts* by default and larger *MinPts* for large and noisy datasets, and choosing *Eps* depending on the distances between points in clusters. Too large *Eps* will produce too large clusters. Too small *Eps* will produce small clusters and too much noise. As a consequence, DBSCAN does not work well when a dataset has varied densities in different parts because a single pair of *Eps* and *MinPts* cannot fit all clusters. Therefore, a number of modified algorithms have been proposed to solve this problem.

The Varied Density-Based Spatial Clustering of Application with Noise (VDBSCAN) algorithm [3] is based on fixing the *MinPts* and determining multiple *Eps* values that match different densities. Firstly, the distance from a point to its k^{th} nearest neighbor (called *k-dist*) is calculated for all points. The *k-dist* values are sorted in ascending order and plotted. If the dataset has highly varying densities, there will be sharp changes in the plot. The values of *k-dist* at the sharp changes will be used as *Eps*. Then the DBSCAN algorithm is repeated in rounds with the fixed *MinPts* and determined *Eps* values in ascending order. The points that are not assigned to a cluster in the previous round will be passed to the subsequence round. VDBSCAN can find the clusters with different densities but it does not perform so well with high dimensional datasets.

The Dynamic Method of Discovery Density Varied Clusters (DMDBSAN) algorithm [4] automatically defines a suitable value of *Eps*. It is also based on *k-dist* plot. The plotted graph shows a sharp slope corresponds to *Eps* for each density level. It is used as a parameter to finds clusters by DBSCAN. *Eps* value will be varied by the number of *MinPts* but does not change immediately. DMDBSCAN is strong in estimates the number of clusters and It can find the clusters with different densities.

Determination of Optimal Epsilon (*Eps*) Value of DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra [6] applies DBSCAN to optimal *Eps* value that is determined automatically by DMDBSCAN technique. They perform clustering in spatial data onto single-density to detect hotspot areas in the highest densities in Sumatra.

The AutoEpsDBSCAN algorithm [5] provides the method that differs from VDBSCAN but has the ability to discover clusters in varied density data by automatically selecting both *Eps* and *MinPts*. However, the algorithm still needs the user enters the number of point, i.e. k. Initially, the *Eps* value derives from calculating the average of distances between all k and its nearest neighbors. The k-distance plot distance in ascending order, which similar to a k-dist graph of VDBSCAN. The graph shows a separate layer in the multi-density dataset. Next step, select the various *Eps* values in layers and define *MinPts* in the difference *Eps* values by computing the average of the number of points in *Eps*.

The Grid and KD-tree for DBSCAN algorithm (GD-DBSCAN) [8] focuses on improving the most time consuming part of DBSCAN, namely the neighborhood search function. This algorithm is based on partitioning the entire area into grid partitions. Therefore, searching for neighbor points within *Eps* distance will involve only the points in the same partition and the surrounding partitions. Points in every partition are organized as kd-tree structure to accelerate searching. In addition, searching in multiple partitions can be done in parallel. This algorithm was applied to discover taxi pick-ups hotspots in Shanghai, China. However, this work did not deal with varying densities.

The P-DBSCAN algorithm [9] is a density-based clustering algorithm for exploring attractive areas using collections of geo-tagged photos. The idea behind this algorithm is that in an attractive area, there should be many distinct people taking photos instead of a few people taking lots of photos. Therefore, this algorithm assigns an owner to each point. This algorithm also handle different densities by splitting a cluster if different areas of the cluster have large differences in densities.

Detecting Hotspots from Taxi Trajectory Data Using Spatial Cluster Analysis [10] the clustering research that using Decision Tree technique to detect hotspot by using taxi transportation in Wuhan on holidays, weekdays and weekend show that the analysis and comparison between hotspot distribution at the particular time in weekend and holidays are quite similar. However, the research found that weekday, weekend, and holidays have resulted in the distribution of data that impact to hotspot appearance.

III. METHODOLOGY

By visually observing pick-up and drop-off locations on the map, we can easily find that most points are on the roadsides but some points form dense clusters around POIs. These clusters are vary in terms of size (diameter) and density. In order to identify these clusters automatically, we need an algorithm that is able to adapt to the differences. Moreover, most POIs are relatively small and can be closely located. We prefer an algorithm that is good at separating small clusters rather than gathering into large clusters.

The proposed method automatically determines both *Eps* and *MinPts* for DBSCAN corresponding to the density of data in the area. Appropriate *Eps* values define from the range of minimum distance that calculates a distance between points and all of its neighbors by Haversine function. This function determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Moreover, this approach calculates the optimal parameters with the Sturge's rule which statistical packages are widely used to produce reasonable histograms. The method can be described as follows.

1) For each point, calculate the distance in meters to all other points by using Haversine function.

2) Determine the lowest range of distances from step 1 based on Sturge's rule.

$$\text{Lowest range} = \frac{\max \text{distance value} - \min \text{distance value}}{1 + 3.322 \log N} \quad (1)$$

where 1 + 3.322 log N is the Sturges's rule to define the class intervals (i.e. of the bins) by 1 + 3.222 is 1 + (1/ log10 (2)) and N is the number of member in the distance set.

3) Choose distance collection in the lowest range and repeat the formula from step 2 to divide the entire range of values into a series of intervals. Then, plot a line graph by number of points and its neighbors in each range.

4) Select *Eps* value from each peak of the line graph where the graph change.

5) Find *MinPts* for each *Eps* level by counting points within *Eps* radius that selected from step 3 to create a histogram by choosing bin sizes with Sturge's rule.

6) Find the bin of the histogram that has the most frequency. The size of the bin is used as *MinPts*.

7) Assign both *MinPts* and *Eps* values to DBSCAN algorithm to cluster data. Points are not clustered by the previous parameters. It will be assigned to a cluster in the next parameters.

8) Repeat from step 4.

IV. Experiment and Results

A. Exploration data

MinPts and *Eps* from the proposed methods are assigned to DBSCAN to clusters area in Bangkok with using taxi pick-up and drop-off in one day which has maximum number of points from the dataset of 9 months (February – September 2016).

The area is classified into two levels depending on density distribution of a designated area, single-density distribution and multi-density distributions. The study is as follows.

- Low population and single-density distribution shows the amount of taxi pick-up and drop-off approximately 530 — 737 points, the highest density on a Friday in July.
- Medium population and single-density distribution shows the amount of taxi pick-up and drop-off approximately 3,206— 3,989 points, the highest density on a Saturday in March.
- High population and multi-density distribution shows the amount of taxi pick-up and drop-off approximately 4,640 — 5,751 points, the highest density on a Saturday in March.

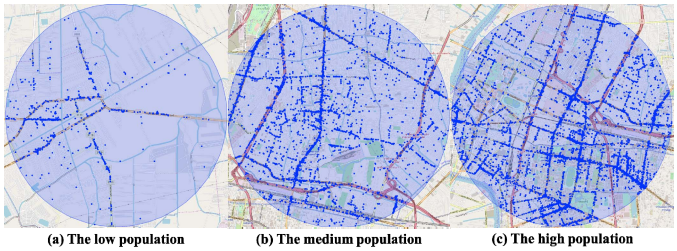


Fig. 1. The areas with different density-distribution

B. Determining Eps and MinPts Parameters

Fig. 2 and Fig. 3, present *Eps* and *MinPts* that are determined by the proposed method. *Eps* is selected from the values at the peaks in the line graph. The values of *Eps* are 0.0005, 0.0001727, 0.002218, 0.0027 and 0.003191 respectively. *MinPts* is the maximum value of frequency in the histogram for each *Eps* as Fig 3. The experimental result compares the clustering by parameters between the proposed method and DMDBSCAN which is defined *Eps* by the distance of the 3rd Nearest Neighbors is shown in Table I and Fig. 4.

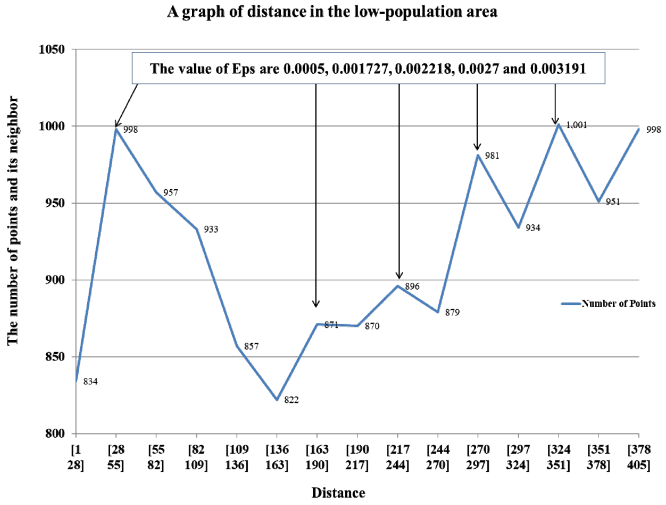
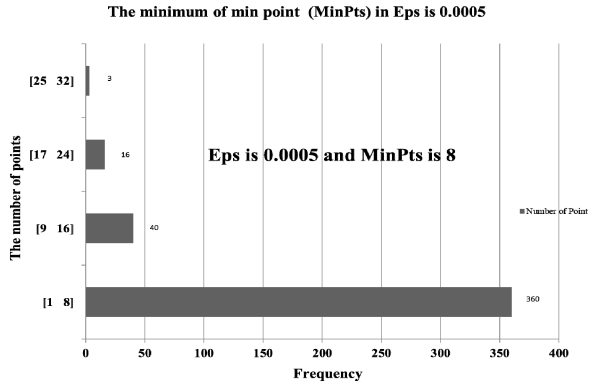
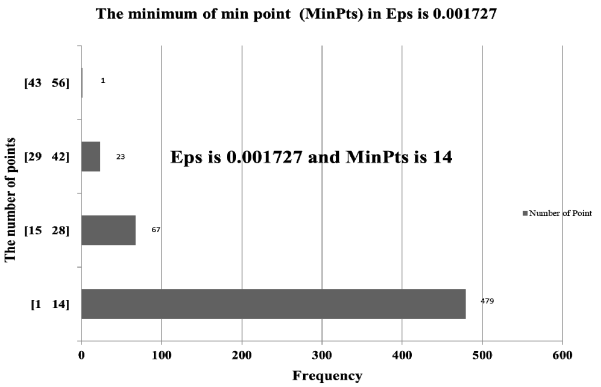


Fig. 2. *Eps* from the proposed method





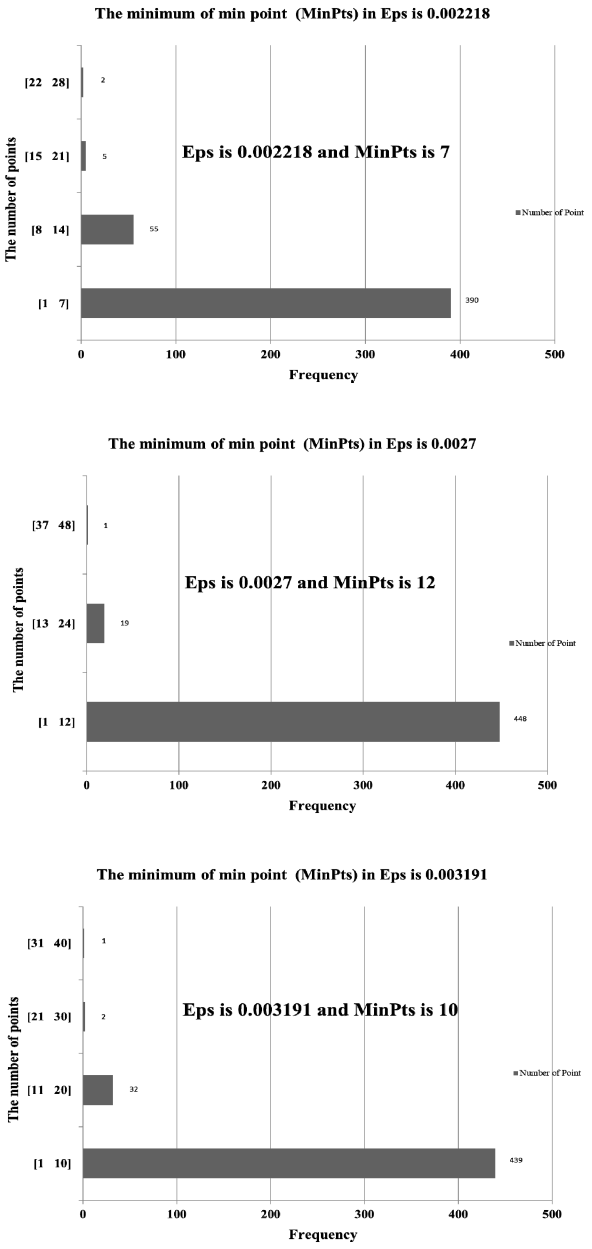
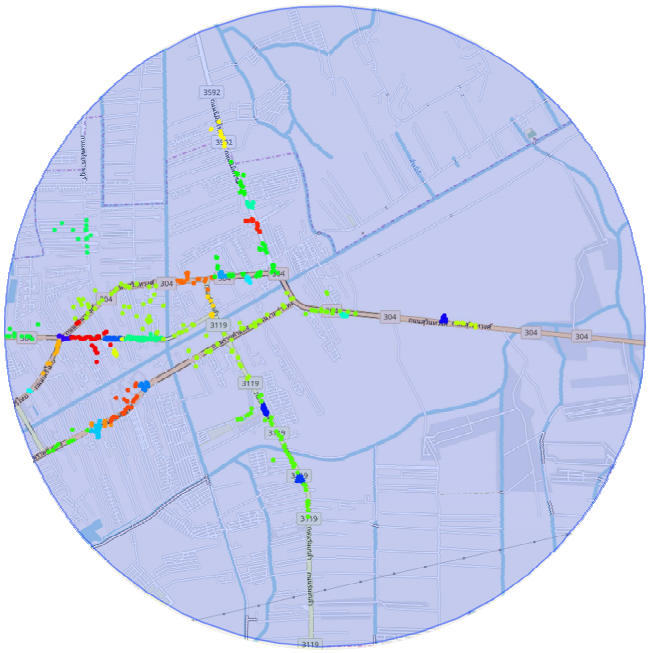


Fig. 3. *MinPts* in each *Eps* level of the proposed method

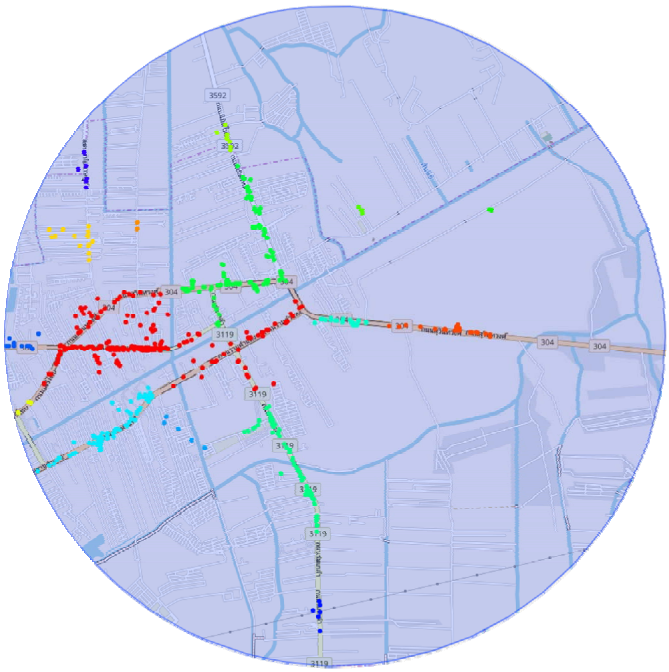
Table I and Fig. 4, present the clustering results by the proposed method and DMDBSCAN algorithm on 3 areas. The proposed method generates 13 clusters of the low-population distribution area, 23 clusters in the medium-population distribution area and 105 clusters on the high-population distribution area. While the DMDBSCAN method generates 16 clusters of the low-population distribution area, 7 clusters in the medium-population distribution area, and 53 clusters in the high-population distribution area. The experiment shows that (a), (c), and (e) can separate smaller clusters better than (b), (d), and (f). When the density is increased, the proposed method used in this experiment can detect hotspots in the area correctly as displayed in (c) and (e). While the DMDBSCAN hardly group the points as in (d) or roughly group the points in (f). Therefore, this can be concluded that the proposed method can cluster data points with different densities completely. So, we can identify the area that can be focused.

TABLE I. THE PARAMETERS AND RESULTS FROM DBSCAN CLUSTERING

Area	Method	Parameter and Results				
		Total Points	*Eps*	*Min Pts*	*Cluster*	*Noise*
Low population	The proposed method	737	0.0005	8	13	488
		488	0.001727	14	5	389
		389	0.002218	7	13	124
		124	0.0027	12	0	124
		124	0.003191	10	0	124
	DMDBSCAN	737	0.164703	3	16	81
Medium population	The proposed method	3,989	0.000682	19	23	2,891
		2,891	0.0027	40	17	1,722
	DMDBSCAN	3,989	0.131256	3	7	32
High population	The proposed method	5,751	0.000155	5	99	4,727
		4,727	0.000309	6	105	3,784
		3,784	0.000464	6	104	2,934
		2,934	0.000618	7	39	2,634
		2,634	0.000773	7	50	2,182
		2,182	0.000927	8	11	2,082
		2,082	0.001082	8	27	1,841
		1,841	0.001236	8	32	1,537
		1,537	0.001382	7	53	1,077
		1,077	0.001536	9	0	1,077
		1,077	0.001691	10	0	1,077
		1,077	0.001845	11	0	1,077
		1,077	0.002	10	9	975
		975	0.002155	10	6	904
		904	0.002309	11	1	893
		893	0.002455	10	16	685
		685	0.002609	10	7	601
		601	0.002764	9	12	488
		488	0.002918	10	0	488
	DMDBSCAN	5,751	0.07893	3	53	122



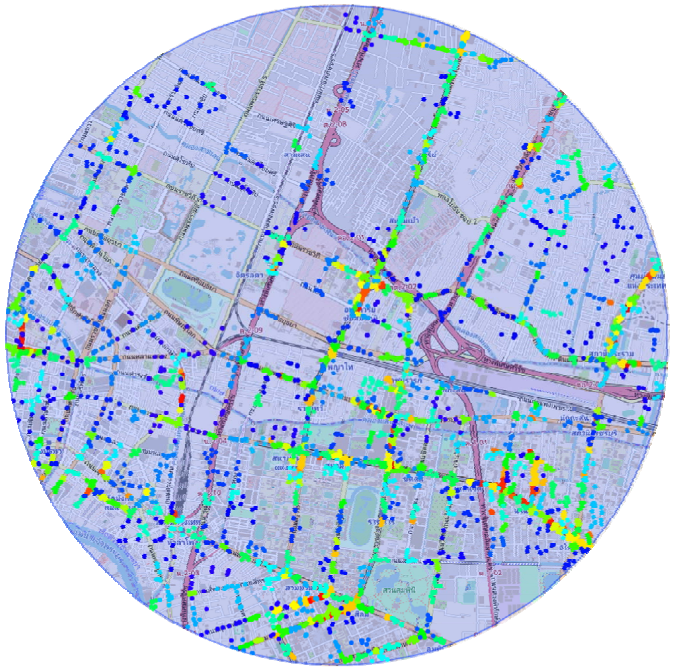
(a) The low-population-clustering result with parameters from the proposed method



(b) The low-population-clustering result with parameters from the DMDBSCAN

(d) The medium-population-clustering result with parameters from the DMDBSCAN

(c) The medium-population-clustering result with parameters from the proposed method



(e) The high-population-clustering result with parameters from the proposed method

(f) The high-population-clustering result with parameters from the DMDBSCAN

Fig. 4: The comparison clustering results between the proposed method and DMDBSCAN

Fig. 5. POI places to appear in clusters

Fig. 5 shows the POIs that appear on an online map of the area of study including department store, market, bank, restaurant, hospital, government office, school, etc. When compared to the clustering results, the detected POIs can be closely associated to the real POIs.

V. Conclusion

This paper proposes the methods to automatically determine the necessary parameters for DBSCAN algorithm by considering the distribution of point density and apply it to pick-up/drop-off locations of taxis in Bangkok. The analysis results show that experimental parameters affected a shape and size of a cluster and the proposed method can determine parameters of DBSCAN for clustering area and has ability to cluster data on both the single-density distribution area and multi-density distribution area. We have verified the cluster area results and found POIs, such as markets, university, restaurants, and shopping malls, etc. In the future, we will apply this method to other areas to detect new POIs to analyze the direction of urban development.

Acknowledgment

This research is supported by Chulalongkorn Academic Advancement into 2nd Century Project, Chulalongkorn University.

References

[1] Z. Zheng, S. Rasouli, and H. Timmermans, Evaluating the Accuracy of GPS-based Taxi Trajectory Records, Procedia Environmental Sciences, vol. 22, pp. 186-198, .2014.

[2] Ester M, Kriegel HP, Sander J, and Xu X , A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), pp. 226-231, 1996.

[3] Liu, Peng , Zhou Dong , and Wu, Naijun, VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise, Proc. of 2007 International Conference on Service Systems and Service Management (ICSSM2007).

[4] Elbatta, Mohammed T. H., and Wesam M. Ashour, A Dynamic Method for Discovering Density Varied Clusters, 2013.

[5] Naik, M. Ganapathi and Kedar Sawant. AutoEpsDBSCAN : DBSCAN with Eps Automatic for Large Dataset, International Journal on Advanced Computer Theory and Engineering, 2013.

[6] Nadia Rahmah, and Imas Sukaesih Sitanggang, Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra, IOP Conference Series: Earth and Environmental Science, 2016.

[7] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu., DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. ACM Transactions on Database Systems, 42, 2017.

[8] L. Zhang, C. Chen, Y. Wang and X. Guan, Exploiting Taxi Demand Hotspots Based on Vehicular Big Data Analytics, 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, pp. 1-5, 2016.

[9] Kisilevich, Slava, Florian Mansmann and Daniel A. Keim, P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos, COM.Geo, 2010.

[10] Po Zhao, Ke Qin, Q Zhou, C. K. Liu, Y. X. Chen, Detecting Hotspots from Taxi Trajectory Data Using Spatial Cluster Analysis, 2015.