Discovery of Patterns in Spatio-Temporal Data Using Clustering Techniques

Amar Mani Aryal
Department of Computer Science
Lamar University
Beaumont, TX USA
e-mail: Aaryal3@lamar.edu

Sujing Wang
Department of Computer Science
Lamar University
Beaumont, TX USA
e-mail: Sujing.Wang@lamar.edu

***Abstract*—Spatial-temporal clustering is very useful unsupervised learning technique and can be used to identify interesting distribution patterns from geo-located data. It is one of the most commonly used data mining techniques in many application domains, e.g. geographic information science, health science, and environmental science. In this paper, we propose a density-based spatial-temporal clustering algorithm for geo-located data points, based on an extension of the SNN (Shared Nearest Neighbor) clustering. The proposed algorithm allows the integration of location, time and other semantic attributes in the clustering process. This algorithm can find clusters of different sizes, shapes, and densities in noisy data. We evaluate the effectiveness of our algorithm through a case study involving a New York City taxi cab pickup data and Maryland crime data. The experimental results show that the proposed algorithm can discover interesting patterns and useful information from spatial-temporal data.**

Keywords-clustering; spatial-temporal clustering; spatial-temporal patterns; shared nearest neighbor clustering

I. INTRODUCTION

Clustering is a popular data-mining technique able to find interesting patterns from complex data. Clustering methods can be categorized into partitioning, hierarchical, density-based, grid-based, and model-based methods. Density-based clustering algorithms are proved to be adequate in the analysis of many types of data. SNN (shared nearest neighbor) [1] is a well-established density-based clustering algorithm, which can find clusters of different sizes, shapes, and densities. SNN has been widely adopted in numerous applications. Spatial-temporal data is one of the fastest growing types of data due to the rapid development of remote sensors, sensor networks, and telecommunication technology and devices. It provides a great opportunity to make such location-based data actionable and insightful with proper analysis techniques. Moreover, such geographic analyses enable a wide range of services such as location-based recommender systems. Traditional studies in data mining focus on discovering clusters from non-spatial and non-temporal data, which are impractical for spatial-temporal data. Pattern discovery from spatial-temporal data is more complex. This paper presents a new density-based clustering algorithm SNN+, which is based on the SNN [1] algorithm. In SNN, the similarity between two points is the number of nearest neighbors that they share. The density associated with a point is the sum of the SNN similarities of a point's k nearest neighbors. Experimental studies prove that our algorithm can effectively cluster spatial-temporal data to identify interesting patterns.

The rest of the paper is organized as follow. Section 2 summarizes related work on spatial-temporal clustering techniques. Section 3 introduces the proposed clustering algorithm in detail. Section 4 evaluates the proposed algorithm with two real data: NYC taxi trips data and Maryland crime data and compares the proposed algorithm with SNN algorithm. Section 5 concludes with a summary of the main findings and proposal of the future work.

II. RELATED WORK

Our work relates to spatial-temporal clustering. Spatial-temporal clustering for point and trajectories has been studied in the literature. Kulldorff [2] introduces basic spatial scan statistics to search spatial-temporal cylinders representing areas where the point objects occur consistently for a significant amount of time; spatial-temporal cylinders are circular regions occurring within a certain time interval. Iyengar [3] extends the basic spatial scan statistics using flexible square pyramid shapes instead of cylinders for spatial-temporal clusters that can either grow or shrink over time and that can also move over time. Wang et al., [4] propose two spatial-temporal clustering algorithms, i.e., ST-GRID and ST-DBSCAN. ST-DBSCAN is an extension of DBSCAN algorithm to perform spatial-temporal clustering by introducing the second parameter of temporal neighborhood radius in addition to the spatial neighborhood radius. The ST-GRID is a grid-based clustering approach which maps the spatial and temporal dimensions into cells. Birant et al., [5] also improve the DBSCAN for spatial-temporal clustering and apply it to discover spatial-temporal distributions of physical seawater characteristics in Turkish seas. A density factor is assigned to each cluster for detecting some noise points when clusters of different densities exist. The density factor of a cluster captures the degree of the density of the cluster. Rinzivillo et al., [6] propose a progressive clustering approach to analyze the trajectories of moving objects supported by visualization and interaction techniques. It progressively applies different distance functions for spatial-temporal data in each step to optimize the outcome of the algorithm. Li et al., [7] introduce the concept of moving micro-cluster to catch some regularities of a moving object. The micro-clusters are kept geographically small at any time. Joshi et al., [8] propose

978-1-5090-6238-6/17/$31.00 ©2017 IEEE

spatial-temporal clustering algorithm for polygons, STPC. STPC extends DBSCAN to cluster polygons by redefining the neighborhood of a polygon as the union of its spatial neighborhood and temporal neighborhood. The temporal aspect is constant or reduced to a fixed interval or time instance when calculating spatial neighbors of a polygon. Moreover, the spatial dimension is instead held to a constant space when calculating temporal neighbors of a polygon. Therefore, STPC clustering algorithm only clusters polygons that do not change their locations, sizes, and shapes over time. Only the non-spatial attributes or properties might change with time. Our proposed clustering algorithm SNN+ extends the well-established density-based Shared Nearest Neighbor (SNN) [1] clustering algorithm. Advantages of SNN+ include its capability to find clusters of different shapes, sizes, and densities in high dimensional data and its tolerance to noises. In addition, SNN+ does not require the number of clusters to be determined in advance. SNN+ can cluster points, trajectories, and polygons.

III. SNN+ CLUSTERING ALGORITHM

The SNN algorithm is a density-based clustering algorithm [1]. We extend SNN for spatial-temporal data by defining the overall spatial-temporal distance between a pair of events. A spatial-temporal event p in dataset D is associated with time t when it occurred, location vector (lo, la) where it occurred, and a set of other attributes $\{a\}$. The overall spatial-temporal distance between a pair of events p and q in dataset D is defined as follows:

$$d_{st}(p,q) = w_s d_s(p,q) + w_t d_t(p,q) + w_a d_a(p,q) \qquad (1)$$

where function $d_s(p,q)$ is any function that can measure the spatial distance between p and q on a sphere from their longitudes and latitudes, e.g. the Haversine formula. Function $d_t(p,q)$ is any function that can measure the temporal distance between p and q taking into account the behavior of time (hours, days, years, season, etc.). Function $d_a(p,q)$ can be any function that suits the attribute domain. w_s, w_t, and w_a are weight factors associated with spatial, temporal, and semantic attributes between p and q ($w_s+w_t+w_a=1$). In general, the similarity between a pair of spatial-temporal events p and q, denoted by *similarity (p,q)*, is the number of k nearest neighbors that they share:

$$similarity(p,q) = |KNN(p) \cap KNN(q)| \qquad (2)$$

The density of p is defined as the sum of the similarities between p and its' k nearest neighbors as follows:

$$density(p) = \sum_{i=1}^{k} similarity(p,q) \qquad (3)$$

where q is the i-th nearest neighbor of p.

All events in dataset D that have a density greater than *MinPs* are labeled as "core". *MinPs* is a user specified parameter.

$$CoreP(D) = \{p \in D \mid density(p) \geq MinPs\} \qquad (4)$$

If two core events are within a radius, *Eps*, of each other, then they are placed in the same cluster. Clusters are then formed by computing the transitive closure of data points that can be reached from an unprocessed core point using their respective nearest neighbors; this process continues until all core points have been assigned to a cluster. The remaining points that are not within a radius of *Eps* of any core points are classified as outliers and not included in any clusters. The pseudocode for SNN+ is listed in Figure 1.

SNN+ (D, k, *MinPs*, *Eps*)
Input: dataset $D = \{p_1, p_2, \ldots, pn\}$, the number of the nearest spatial-temporal neighbors k, the core point threshold *MinPs*, the similarity threshold *Eps*.
Output: set of generated clusters C_i.

1. Compute spatial-temporal distance matrix of D;
2. Mark every p in D 'unprocessed';
3. **For** every pair of events p and q in D
4. compute similarity (p,q);
5. **end for**
6. **for** every p in D
7. compute density (p);
8. **if** density (p) is greater than *MinPs* **then**
9. mark p as 'core';
10. **end if**
11. **end for**
12. **for** every core p in D
13. **If** p is marked 'unprocessed' **then**
14. form a cluster C_i of points that can be reached from p following the entries of the respective NN-lists of core points;
15. **end if**
16. mark all points in $\boldsymbol{C_i}$ as 'processed';
17. **end for**
18. **return** set of generated clusters C_i;

Figure 1. Pseudocode for SNN+.

SNN+ requires several user-defined parameters that have significant impacts on clustering results. These user-defined parameters need to be changed and adapted according to the data being clustered:

- k: the number of the nearest spatial-temporal neighbors. It is the most important parameter as it determines the granularity of the clusters. In general, if k is too small, SNN+ will tend to find many small clusters and a lot of outliers. On the other hand, if k is too large, SNN+ will tend to find only a few large clusters.
- *MinPs*: the core point threshed. It is the minimum number of the shared neighbors required for core points. It allows the user to control how many points above the similarity threshold are needed to qualify as a core point. *MinPs* should be smaller than k.
- *Eps*: the density threshold. It is used as the criteria to define the SNN density of each point. *Eps* should be smaller than k as well.

SNN+ follows the structure of SNN. The time complexity of SNN+ is $O(n^2)$ without the use of an indexing structure, where n is the number of events in the dataset D. If an indexing structure such as a k-d tree or an R* is used, the time complexity will be reduced to $O(nlog(n))$. The space complexity is $O(kn)$ since only the k nearest spatial and temporal neighbors need to be stored; while the nearest neighbors can be computed once and used repeatedly for different runs with different parameter values.

IV. Case Study

A. Dataset Description

The TLC Trip Record Data [9] were collected by technology providers authorized under the Taxicab and Livery Passenger Enhancement Programs. There are over 1.1 billion taxi trips from January 2009 through June 2016. Each individual trip record contains precise location coordinates for where the trip started and ended, timestamps for when the trip started and ended, and several other variables including fare amount, payment method, and distance traveled. The crime data is derived from reported crimes classified according to Maryland criminal code and documented by approved police incident reports including raw data of all founded crimes reported after July 2013. Each individual crime record contains precise location coordinate for where the crime occurred, timestamps for when it happened, and several other variables including class description, police district name, and agency.

B. TLC Trip Record Data

In this section, we consider an application of our SNN+ clustering technique to TLC Trip Record data [9] and a brief comparison with the existing SNN algorithm. We analyzed taxi pickups from 6 AM to 7 AM on January 8, 2014 and attempted to find spatial-temporal clusters in each 20 minutes of interval.

The value of k is chosen based on the silhouette value [10]. The silhouette value measures the similarity of a point to other points in the same cluster, when compared to points in other clusters. The silhouette value for a cluster is obtained by computing the mean of the silhouette values of all the points within that cluster. We selected the value of k for which the obtained clusters have highest silhouette values. e.g. k = 20 for taxi data.

We used the Euclidean distance to compute the spatial similarity and the temporal distance function described in (5) to compute the temporal distance $d_t(p,q)$ between events p and q.

$$d_t(p,q) = \begin{cases} 0, & p \text{ and } q \in \text{the same time interval} \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

The spatial and temporal distances are normalized and a weighted sum of the distances is computed with (w_s=0.5, w_t=0.5). We obtained 22 clusters with k=20 and the clusters were categorized based on the mean value of time.

Fig. 2, Fig. 3 and Fig. 4 show the dense spatial-temporal clusters that have been identified. These clusters represent the hotspots for taxi pickup in each 20-minute time interval and are located around various train and bus terminals. If we observe closely, we see some patterns: The clusters near the Port Authority Bus Terminal and 34 St- Penn Station continue in all the three time intervals which suggests that these areas are crowded with people getting off the train and bus early in the morning and looking for taxi rides. The cluster near the Grand Central Terminal continues to appear in the first two time intervals. However, this cluster does not appear in the interval 6:40 AM to 7:00 AM which means that fewer people were looking for taxi rides near the Grand Central Terminal. We infer that it takes relatively less time to get a taxi ride in this area within this time window.

Table I displays the standard deviation and mean of spatial and temporal attributes for the taxi pickup data. Time mean is computed by taking the mean of the minutes since the hour is constant for all the data points. The low values of standard deviation as shown in Table I. suggest that the clusters obtained are dense in the spatial domain.

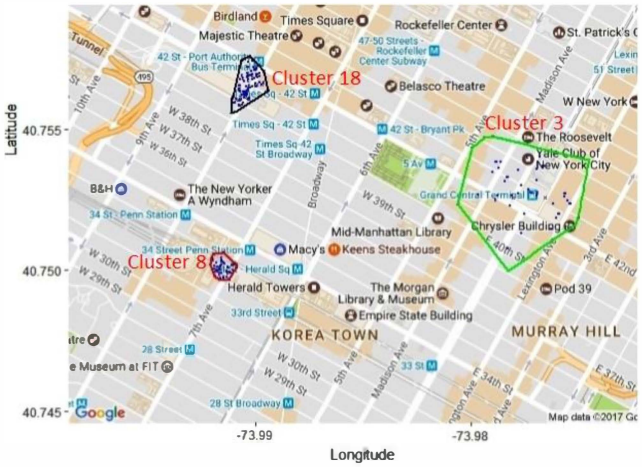

Figure 2. Clusters 18, 8, 3 identified during 6:00 AM to 6:20 AM, Jan 8 2014, in black, red, and green contour respectively.

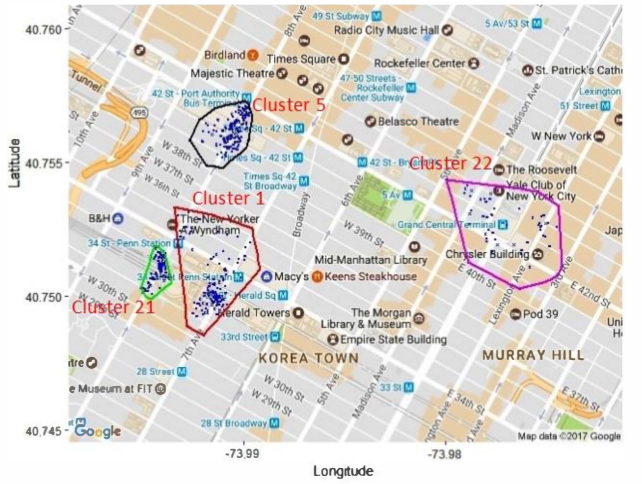

Figure 3. Clusters 5, 1, 21, 22 identified during 6:20 AM to 6:40 AM Jan 8 2014, in black, red, green, and magenta contour respectively.

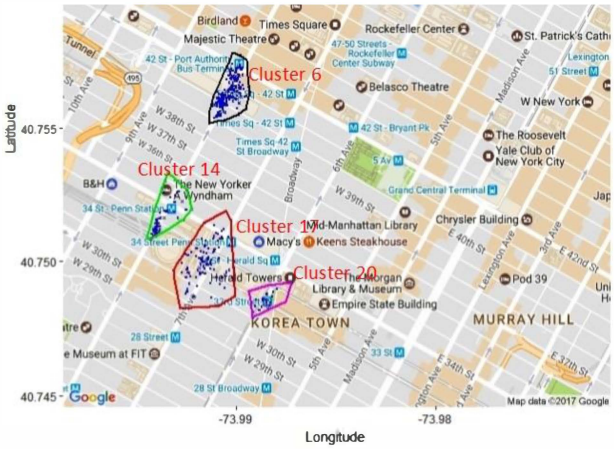

Figure 4. Clusters 6, 17, 14, 20 identified during 6:40 AM to 7:00 AM, Jan 8 2014, in black, red, green, and magenta contour respectively.

TABLE I. VARIATION MEASUREMENT OF CLUSTERS IN TAXI PICKUP DATA

Cluster No.	Latitude SD (σ_{lat})	Longitude SD (σ_{long})	Latitude Mean (μ_{lat})	Longitude Mean (μ_{long})	Days Mean (μ_{days})
22	0.0147	0.0184	39.1601	-77.1916	14.67
23	0.0114	0.0150	39.1761	-77.2641	12.55
15	0.0091	0.0076	39.0062	-76.9875	17.20
5	0.0176	0.0163	39.1588	-77.1896	49.12
13	0.0180	0.0123	39.1746	-77.2612	46.21
10	0.0134	0.0112	39.0701	-76.9534	51.19
8	0.0300	0.0172	39.0188	-77.0202	77.80
16	0.0166	0.0145	39.1699	-77.1874	75.89
2	0.0157	0.0118	39.1549	-77.0771	75.43

C. Crime Data

In this section, we consider an application of our SNN+ clustering technique to Maryland Crime Data [11]. We analyzed the crime data of April, May, and June 2016. We used the Euclidean distance to compute the spatial similarity and the temporal distance function given by (6).

$$d_t(p,q) = \begin{cases} 0, & p \text{ and } q \in \text{the same month} \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

where $d_t(p,q)$ is the temporal distance between two events p and q.

We normalized the spatial and temporal distance and computed the weighted sum of distance using weights (w_s=w_t=0.5). We used natural numbers from 1 through 91 to denote the days in the three-month period. For e.g. April corresponds to the interval [1, 30], May corresponds to the interval [31, 61], and June corresponds to the interval [62, 91]. We obtained 27 clusters with k=30 (k was chosen by the same method as in taxi data) and categorized the clusters based on the mean value of days.

TABLE II. VARIATION MEASUREMENT OF CLUSTERS IN CRIME DATA

Cluster No.	Latitude SD (σ_{lat})	Longitude SD (σ_{long})	Latitude Mean (μ_{lat})	Longitude Mean (μ_{long})	Time Mean (μ_{time})
18	0.00044	0.00036	40.7566	-73.9904	12.04
8	0.00022	0.00031	40.7501	-73.9916	09.98
3	0.0010	0.0014	40.7528	-73.9774	10.67
5	0.00053	0.00058	40.7562	-73.9905	29.96
1	0.0010	0.0007	40.7505	-73.9915	30.25
21	0.00048	0.00032	40.7510	-73.9942	34.05
22	0.0010	0.0016	40.7527	-73.9769	29.89
6	0.00054	0.00043	40.7564	-73.9904	49.98
17	0.00073	0.00069	40.7066	-74.0132	54.06
14	0.00053	0.00048	40.7515	-73.9938	51.07
20	0.00031	0.00046	40.7486	-73.9885	50.14

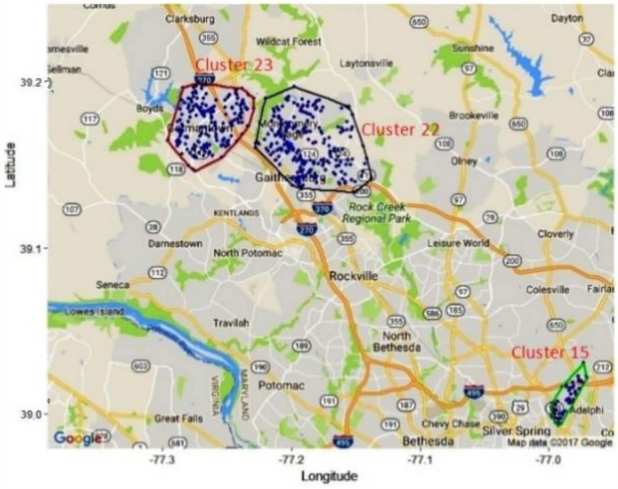

Figure 5. Clusters 22, 23, 15 identified during April 2016 in black, red, and green contour respectively.

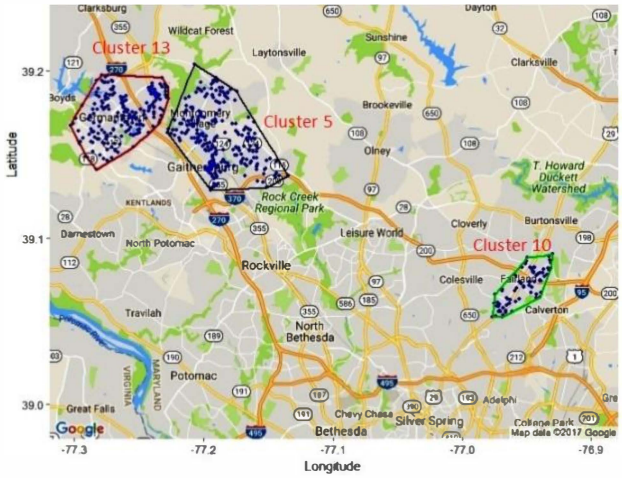

Figure 6. Clusters 5, 13, 10 identified during May 2016 in black, red, and green contour respectively.

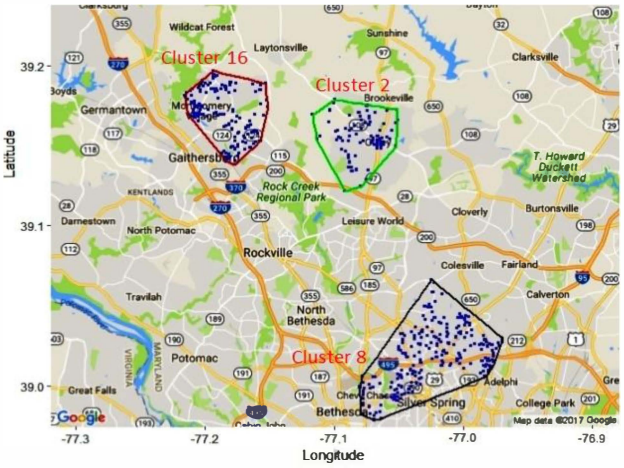

Figure 7. Clusters 8, 16, 2 identified during June 2016 in black, red, and green contour respectively.

D. Comparision between SNN+ and SNN

In this section, we compare the performance of SNN+ with SNN. We applied SNN [1] algorithm in the same TCLP trip record data (6 AM to 7 AM on January 8, 2014) to find the clusters of taxi pickups in New York. Fig. 8 displays three such clusters out of the 34 total clusters identified with k=20. It is evident from Table III that the time means of the clusters are 33.601, 35.790, and 32.085, which are very close to the mean value of time for the entire dataset which is 34.952. Thus, SNN can identify clusters that are dense in the spatial domain, but not in the temporal domain. However, SNN+ can identify clusters that are dense in space, time, and other semantic attributes.

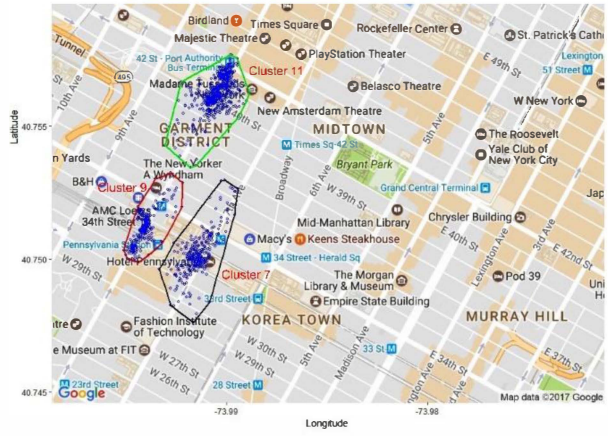

Figure 8. Cluster 7, 9, 11 obtained using SNN clustering algorithm in taxi data.

TABLE III. VARIATION MEASUREMENT OF SELECTED CLUSTERS IN TAXI DATA USING SNN

Cluster No	Longitude SD (σlong)	Latitude SD (σlat)	Latitude Mean (μlat)	Longitude Mean (μlong)	Time Mean (μtime)
7	0.00064	0.00078	40.7502	-73.9914	33.601
9	0.00052	0.00064	40.7512	-73.9941	35.790
11	0.000636	0.00065	40.7563	-73.9905	32.085

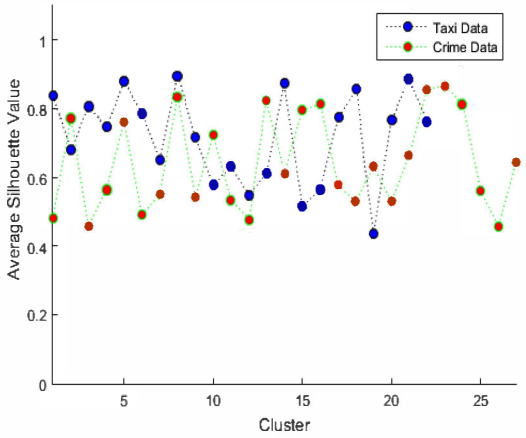

Figure 9. Average Silhouette value of the clusters obtained using SNN+ clustering algorithm in taxi and crime data.

E. Quality of Clusters

We now attempt to measure the quality of clusters obtained by our clustering algorithm. We used the standard deviation and mean of the spatial attributes along with the average Silhouette [10] value of each cluster to examine the quality of the clusters. The low standard deviation values for latitude and longitude as depicted in table I and II and the high average Silhouette value of the clusters as shown in Fig. 8 suggest that our clustering algorithm can generate high quality clusters.

V. CONCLUSION AND FUTURE WORK

In this paper we introduced a new density-based spatial-temporal clustering algorithm that can find clusters of different shapes, sizes, and densities. The algorithm can automatically determine the number of clusters. The clustering results achieved with two real spatial-temporal data, e.g. NYC Taxi Data [9] and Maryland Crime Data [10], are very promising as spatial-temporal data were effectively clustered identifying relevant patterns.

Our ongoing work focuses on developing a spatial-temporal data mining framework, supported by a scalable computing infrastructure for big spatial-temporal data by utilizing high-performance computing resources, such as Hadoop-GIS.

REFERENCES

[1] L. Ertöz , M. Steinbach , and V. Kumar, "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data", in the Second SIAM International Conference on Data Mining, San Francisco, CA, 1-3 May 2003.

[2] M. Kulldor, "A Spatial Scan Statistic", Communications in statistics: Theory and Methods, Vol.26, pp.481-1496, 1997.

[3] V.S. Iyengar, "On Detecting Space-time Clusters", In:Proceedings of the 10th ACM SIGMOD, Seattle,Washington, USA, 22-25 August, 2004.

[4] M. Wang, A. Wang, and A. Li, Mining," Spatial-temporal Clusters from Geodatabases", Lecture Notes in Computer Science, Vol. 4093, pp. 263-270, 2006.

[5] D. Birant and A. Kut, "ST-DBSCAN: an Algorithm forClustering Spatial-temporal Data", Data& KnowledgeEngineering, Vol. 60, pp. 208-221, 2007.

[6] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti,N. Andrienko, and G. Andrienko, "Visually DrivenAnalysis of Movemen Data by Progressive Clustering,Information Visualization", Vol. 27, pp. 225-239, 2008.

[7] Y. Li, J. Han, and J. Yang, "Clustering Moving Objects", In: the 10th ACM GISMOD conference, Seattle,Washington, USA. 22-25 August, 2004.

[8] D. Joshi, A. Samal, and L.K. Soh, "Spatial-temporalPolygonal Clustering with Space and Time as First class Citizens, Geoinformatica", Vol. 17, pp.387-412,2013.

[9] NYC taxicab data: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml, last accessed Jan. 2017.

[10] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", Journal of computational and applied mathematics, 1987.

[11] Maryland Crime Data: https://catalog.data.gov/dataset/crime, last accessed Jan. 2017.