DBSCAN OPTIMIZATION FOR IMPROVING MARINE TRAJECTORY CLUSTERING AND ANOMALY DETECTION

X. Han, C. Armenakis, M. Jadidi

Geomatics Engineering, Dept. of Earth and Space Science and Engineering, Lassonde School of Engineering,
York University, Toronto, Canada – (han978, armenc, mjadidi)@yorku.ca

Commission IV, WG IV/7

KEY WORDS: DBSCAN, Trajectory Clustering, Mahalanobis Metric, Machine Learning, Marine Transportation

ABSTRACT:

Today maritime transportation represents 90% of international trade volume and there are more than 50,000 vessels sailing the ocean every day. Therefore, reducing maritime transportation security risks by systematically modelling and surveillance should be of high priority in the maritime domain. By statistics, majority of maritime accidents are caused by human error due to fatigue or misjudgment. Auto-vessels equipped with autonomous and semi-autonomous systems can reduce the reliance on human's intervention, thus make maritime navigation safer. This paper presents a clustering method for route planning and trajectory anomalies detection, which are the essential part of auto-vessel system design and development. In this paper, we present the development of an enhanced density-based spatial clustering (DBSCAN) method that can be applied on historical or real-time Automatic Identification System (AIS) data, so that vessel routes can be modelled, and the trajectories' anomalies can be detected. The proposed methodology is based on developing an optimized trajectory clustering approach in two stages. Firstly, to increase the attribute dimension of the vessel's positioning data, therefore other characteristics such as velocity and direction are considered in the clustering process along with geospatial information. Secondly, the DBSCAN clustering model has been enhanced by introducing the Mahalanobis Distance metric considering the correlations of the position cluster points aiming to make the identification process more accurate as well as reducing the computational cost.

1. INTRODUCTION

Today maritime transportation represents 90% of international trade volume and there are more than 50,000 vessels sailing the ocean every day. Therefore, reducing maritime transportation security risks by systematically modelling and surveillance should be of high priority in the maritime domain. By statistics, between 75% and 96% of maritime accidents are caused by human error due to fatigue or misjudgment (Merkel, 2019). Auto-vessels equipped with autonomous and semi-autonomous systems can reduce the reliance on human's intervention, thus make our oceans and maritime navigation safer. Besides navigation safety, auto-vessels also contribute to surveying efficiency, cost saving, environmental protection, etc. (Marr, 2019). Despite the security concern and hurdles of resolution about the regulation, auto-vessels still face fewer barriers to adoption comparing to unmanned vehicles driving on the road (Merkel, 2019). In contrast to air traffic control applications, auto-vessels faces fewer technical challenges since only two-dimensional space is involved, which reduces the trajectory domain complexity (Vespe et al., 2012). Thus auto-vessels should be the most promising automatous vehicles implemented in the near future. In December 2018, Rolls-Royce and Finferries have demonstrated world's first fully autonomous ferry (Rolls, 2018). But the ships were only deployed on simple inland where waters are calm, the route is simple, and there isn't much traffic. Indeed, there is still a long way to go in design and development of auto-vessel related research, including route planning and trajectory anomalies detection, situational awareness and intelligent responses toward changing environments. This paper focuses on the first element, route clustering and trajectory anomalies detection, by proposing an algorithm for generating high precise modelling of the vessels' trajectories and detecting vessels trajectories anomalies such as unexpected stops, deviations from regulated routes, or inconsistent speed.

The reliable open-sourced data sources for generating nautical routes are historical and real-time Automatic Identification System (AIS) data (Silveira et al., 2013). AIS is an automatic tracking system to identify and locate vessels by exchanging data with other nearby ships, AIS base stations, and satellites. According to Safety of Life at Sea (SOLAS) convention, ships of 300 gross tonnage and upwards in international voyages, 500 and upwards for cargoes not in international waters, and passenger vessels are obliged to be fitted with AIS equipment, making AIS data abundant globally. Furthermore, AIS becomes a worldwide data standard and therefore this coherent source of information can be suitable for global marine transportation traffic modelling and analysis. Though, in this paper we use open-sourced AIS data as the main data source for the proposed algorithm testing. Given the large amount of AIS data, this is more feasible to adopt unsupervised learning in modeling and anomaly detection processes with a high degree of automation.

In this paper, Density-Based Spatial Clustering of Applications with Noise (DBSCAN, Ester et al., 1996) is proposed to be used as the foundation of the marine trajectory modelling. DBSCAN, an unsupervised method, is now available in many clustering libraries and widely used in many real-world applications (Hall et al., 2009). As DBSCAN is relying on a density-based notion of clusters, this consider to be an effective method to discover clusters of arbitrary shapes as well as identifying outliers (Ester et al., 1996). Thus, DBSCAN demonstrates huge potentials to be applied on marine trajectory clustering (Liu, 2015). However, applying the traditional DBSCAN clustering method has huge shortcoming with unevenly distributed authentic AIS data, this makes unreliable method to be applied on marine trajectory

https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-455-2020

clustering without optimization. The traditional DBSCAN method requires two input parameters, *MinPts* and ε, and the user needs to determine appropriate values for them. But in real life, this is very difficult to find the optimal parameters when the data and scale cannot be well understood. Furthermore, as the traditional DBSCAN is based on Euclidean distance metric, this sometimes cannot handle data with complex shape and distribution. Thus, novel distance metrics need to be proposed to optimize the DBSCAN performance.

In this paper, we present an enhanced DBSCAN clustering method that can be applied on historical or real-time AIS data, therefore the vessel routes can be modelled, and the trajectories' anomalies can be detected. In section 3, we firstly described the data source and the synthetic data for algorithm testing, followed by two stages of our proposed methodology. Firstly, to increase the dimensions of the vessel's positioning data, additional attributes such as velocity and direction are considered in the clustering process along with the geospatial information. Secondly, the DBSCAN clustering method is enhanced by introducing the Mahalanobis Distance metric, taking into account the correlations of the position cluster points aiming to make the identification process more accurate as well as reducing the computational cost. Results of the enhanced DBSCAN applied on AIS data are discussed in Section 4, whereas conclusions and future work directions presented in Section 5.

2. RELATED WORKS

This section discusses the development and state-of-the-art marine trajectory clustering methods that are widely used. Considering the critical role of trajectory data mining in modern intelligent systems for surveillance security, abnormal behaviours detection, crowd behaviours analysis and traffic control, trajectory clustering has attracted growing attention (Bian et al., 2018). Existing trajectory clustering methods can be grouped into three categories: supervised, unsupervised and semi-supervised algorithms (Mikhail et al., 2009).

Supervised algorithms aim at training a model which is able to determines the labels of testing data after learning labelled training data. Therefore, supervised algorithms perform tasks based on understanding of "ground truth", thus the accuracy is usually high. The most commonly used algorithm is Nearest Neighbour algorithm. The k-Nearest Neighbour algorithm (k-NN), for example, are finding a voting system to determine the category of a new entity and all data are kept in the same feature space. In trajectory clustering, the distances from an inquiry trajectory to all labelled trajectory data are computed, and the label of the inquiry trajectory is voted by its k nearest neighbours (Gao et al., 2007). Support Vector Machine (SVM) is trained to generate the hypervolume, which can separate the outliers from the valid trajectories (Piciarelli et al., 2008). But SVM as a binary classifier, have difficulties group trajectories into sub-clusters. Neural Network is another widely used supervised algorithm. The network is constructed by a number of layers of connected neurons, each of which is represented by a regression model. Since in most cases Neural Network is used for data classification, neural networks cluster trajectories through classifying observations into the pre-defined or pre-labelled clusters (Cho and Chen, 2014). Usually supervised algorithms have high accuracy but requires huge human efforts to prepare the training data and need massive data to train it.

Unsupervised algorithms infer a function to describe internal relationships between unlabelled data. Unlike the supervised learning which requires massive labelled training data, unsupervised learning is a type of self-organized learning that helps find previously unknown patterns in data set without pre-existing labels. Hidden structures and unknown similarities can be found by unsupervised algorithms. The hierarchical clustering model is a tree-structured model that considers more attributes at each level (Li et al., 2006). Spectral Clustering models represent trajectory data as an affinity matrix, then compute internal relationships by analysing these affinity matrices (Xiang and Gong, 2008). Densely clustering models classify trajectories by considering the spatial information calculated by distance metrics. The close points are very likely to be clustered into same group. Widely used DBSCAN and k-means are inspired by this idea. k-means methods divide data into k clusters but are difficult to be implemented in authentic data since the k value will never be well-known on real-world problems (Galluccio et al., 2012). DBSCAN cluster points together which are "density reachable" (Ester et al., 1996). Unsupervised algorithms spare human burdens from preparing massive training data but usually need optimization before implementation and also have the disadvantages of high computation cost and heavy memory load.

Semi-supervised algorithms fall between unsupervised algorithms and supervised algorithms. The algorithms only require a small amount of labeled data to train the model then conduct the cluster tasks while updating the model with unlabeled data. So, comparing to the supervised learning, semi-supervised algorithms need much smaller human burdens on preparing training data. Comparing to the unsupervised ones, semi-supervised models usually have better performance regard to accuracy. Some semi-supervised algorithms are invented starting from unsupervised or supervised algorithms. For example, small amount of pre-defined clusters can be prepared by the humans and the new observations are clustered to update the classifier automatically (Gurung et al., 2014). In this way, semi-supervised algorithms may combine the advantages of both supervised and unsupervised algorithms and result in more efficient methods.

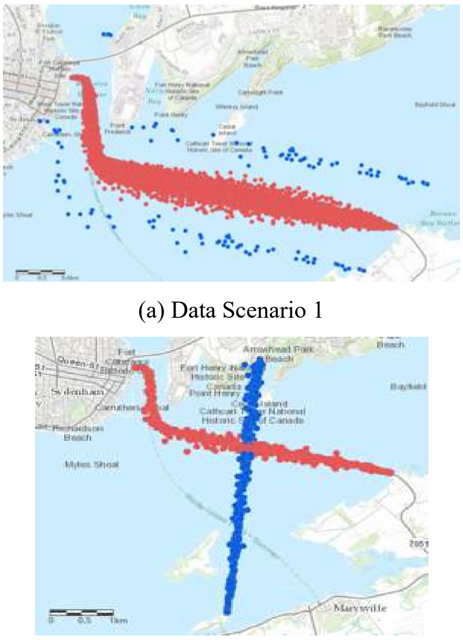
This research project is inspired by the semi-supervised algorithms and proposed to be applied to the trajectory clustering in real-world problems. This research project starts from optimizing an unsupervised algorithm, DBSCAN, then modify it into a semi-supervised model. The foundation of the models can be generated from data preparation by human experts or semantic data, then unlabelled historical observations will be sent to the model to update the model. Then the model can predict new data records which cluster it belongs to and whether it is an outlier.

3. METHODOLOGY

3.1 Data Preparation

AIS data is point-based data showing vessels' information at a specific time, including location coordinates, speed, heading, and vessel type. The AIS raw data we selected is open-source and hosted at *MarineCadastre.gov*. The AIS Data, available from 2009 to 2017 in CSV format, covers most of the places globally. The Earth's sphere has been divided into 20 zones, and each zone has 12 CSV files to record the data month by month. The raw data's total size is around 800GB before uncompressed. Due to the huge size of the data available, it would be inapplicable to use without filtering and re-selecting. In this paper, we selected Wolfe Island Ferry data (Lake Ontario, Canada) at January 2017 for algorithm testing. MongoDB was utilized for database management, for the efficiency on using built-in solutions to facilitate data manipulations such as geospatial indexing and advanced geospatial queries.

In order to test the performance of the clustering algorithm, some more artificial data are generated as additional supportive datasets. For instance, the optimized DBSCAN algorithm should be able to identify outliers and noises from the main trajectories. Also, the algorithm should distinguish different trajectories from intersections. So, two synthetic datasets are created based on the Lake Ontario small dataset for testing two mentioned scenarios. The first scenario was designed to test the performance on outlier detection where 150 noisy points were randomly generated around the main trajectory (Fig. 1a), while the second one tested whether the clustering algorithm is able distinguish intersections and identify them as separate clusters where 2000 points were rotated by 90 degrees (Fig. 1b). In both Figures 1a and 1b, the real data is shown in red colour and the synthetic outliers and crossing data are shown in blue.



(a) Data Scenario 1

(b) Data Scenario 2

Figure 1. Tested datasets at Wolfe Island Ferry data collected January 2017

3.2 Enhanced DBSCAN Model Development

DBSCAN is a widely-used density-based clustering algorithm: given a set of points in some space, it groups points that are closely packed together (points with many nearby neighbours), marking as outliers that lie alone in low-density regions (nearest neighbours are too far away) (Ester et al., 1996). DBSCAN requires the user to define two terms: ε and *minPts*. Term ε is a parameter specifying the radius of a neighbourhood with respect to some point. Term *minPts* is a parameter that determines if a point is a core point. Under the definition of DBSCAN clustering, the points are classified into core points, non-core points but (density) reachable points and outliers. If at least *minPts* points are within distance ε, this point is defined as a core point, and the points within this distance will be in the same cluster with the core point. DBSCAN clustering will iterate from point to point, calculate the distances among points, and to identify the point category. The points which are not reachable from any core points are outliers or noise points. The core points surrounding with other reachable points make them a cluster, when the core points are reachable from each other will join the clusters together to form a larger cluster. Because DBSCAN algorithm clusters points by density reachability, complex shapes from trajectories can be handled well by DBSCAN. Thus, DBSCAN has the potentials to dealing with geospatial data clustering.

To solve the aforementioned challenges, an optimized DBSCAN on trajectory data is developed in this paper. First, the dimension of the geospatial data is increased so that other attributes such as velocity and direction are considered in the clustering process besides just geospatial information. Second, the DBCAN clustering model has been modified with the Mahalanobis Distance metric, taking account of correlations between each point and the whole cluster to make the identification process more accurate, and also reduce the computational cost.

3.2.1 High Dimensional Geospatial Data and Data Normalization

The traditional densely based clustering works with two-dimensional data. Latitude and longitude are the only attributes to be considered. Based on spatial density the 2D points will be clustered together. Increasing the dimensions of the data can change the concept of "density reachability" and enhance the clustering model abilities to find more complex unknown similarities between the data. Each data record is extended into a five-dimensional vector, as shown at Eq. (1) taking into account Speed over Ground (SOG), Course over Ground (COG) and Heading.

$$\boldsymbol{x} = [latitude, longitude, SOG, COG, Heading]^T \quad (1)$$

The data are normalized between [-1, 1] as required by most of the machine learning techniques including DBSCAN. After normalization, all six attributes share the same mean value, the same variance value and the same weight when clustering.

3.2.2 DBSCAN using Mahalanobis Distance Matrix

As mentioned in the Section 3.2, traditional DBSCAN clustering will iterate from point to point, calculate the distances among points, and to identify core points and then clustering the surrounding points together. Thus, the traditional DBSCAN has two main shortcomings: 1) high computation costs and 2) only local characteristics are considered when identifying the cluster. Using the Mahalanobis Distance metric will resolve the aforementioned challenges by increasing the cost efficiency and considering the correlation between the point within the entire cluster. The Mahalanobis metric describes the distance between one point to a group of points. The Mahalanobis distance DM($\mathbf{x}$) from a point data, $\mathbf{x}$, to a cluster with mean, $\boldsymbol{\mu}$, and covariance matrix, $\mathbf{S}$, are defined by Eqs. (2), (3) and (4) respectively.

$$D_M(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{S}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})^T} \quad (2)$$

$$\boldsymbol{\mu} = \left[\mu_{latitude}, \mu_{longitude}, \mu_{SOG}, \mu_{COG}, \mu_{Heading}\right]^T \quad (3)$$

$$\boldsymbol{S_{ij}} = cov(x_i, x_j) = \langle (x_i - \mu_i)(x_j - \mu_j) \rangle \quad (4)$$

The proposed algorithm requires some efforts to generate pre-defined clusters and the user input ε term. Then each point will be iterated and Mahalanobis distance to each pre-defined clustered will be calculated. The distance will be compared with the user input ε term. If the Mahalanobis distance is smaller than ε, the point can be identified to belong to the cluster and then update the cluster parameters. If the Mahalanobis distance is larger than ε, then the point is an outlier to this cluster. Then the clusters are updated by the new points coming in and start the second clustering iteration. The algorithm continues until no outliers are closer than ε to all clusters.

4. EXPERIMENTAL RESULTS

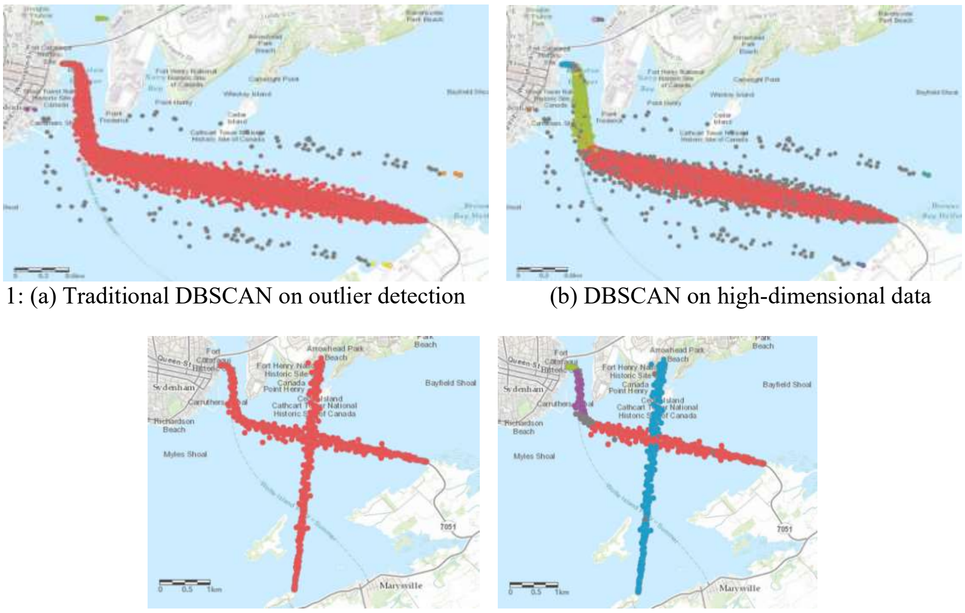
Two data scenarios tested how the proposed algorithm can deal with outlier detection and differentiate crossing data. The two comparison results are presented showing how increasing data dimension and using Mahalanobis distance metric on DBSCAN can improve the clustering performance. The detailed results about clustering parameters and the clustered groups can be found in Figure 2, Figure 3, Figure 4, Table 1 and Table 2.

4.1 Results of DBSCAN on High Dimensional Geospatial Data

Figures 2(a) and 2(c) show the clustering result by the traditional DBSCAN for both data scenarios. Figures 2(b) and 2(d) show the clustering results when the data dimension has been increased. Different clusters are represented by different colours. As presented in Figure 2(a), traditional DBSCAN can detected outliers only by 'density-reachability'. The artificially generated points outside of the main trajectory have all been successfully detected, showing the potential of DBSCAN on trajectory anomaly detection. But as presented in Figures 2(a) and 2(c), traditional DBCSAN grouped all connected points into one single cluster and is not able to distinguish overlapped trajectories with various headings. This problem resolved by applying DBSCAN on High Dimensional Geospatial Data. After finding the proper parameters like ε and *minPts* by using trial-and-error method, the trajectories got 'density disconnected' in the joining point, and thus they can be successfully differentiated. Comparing the clustering result presented in Figures 2(b) and 2(d) to the results in Figures 2(a) and 2(c), the data was group into more clusters and more outliers were detected. In Figure 2(b), some outliers inside of the main trajectory were also been detected as they have inconsistent heading and speed comparing to the surrounding points. Thus, these comparison tests show that increasing the data dimension can optimize clustering performance and efficiently detect the anomalies.

4.2 Results of DBSCAN using the Mahalanobis metric and Check Points (Benchmarks)

Since there are no ground truth benchmarks for unsupervised clustering results, we took little amount of human efforts for creating checked points of each cluster, based on the results from Figure 2(b) and 2(d). Then the trajectory data are clustered and anomalies are detected using the Mahalanobis Distance metric. This test simulates two scenarios based on the level of understanding and familiarity on the dataset. Figures 3(a) and 3(c) are the clustering results when only small amount of check points is available,



Data Scenario 1: (a) Traditional DBSCAN on outlier detection (b) DBSCAN on high-dimensional data

Data Scenario 2: (c) Traditional DBSCAN on crossing data (d) DBSCAN on high-dimensional data

Figure 2. Comparison result of DBSCAN on high dimensional geospatial data

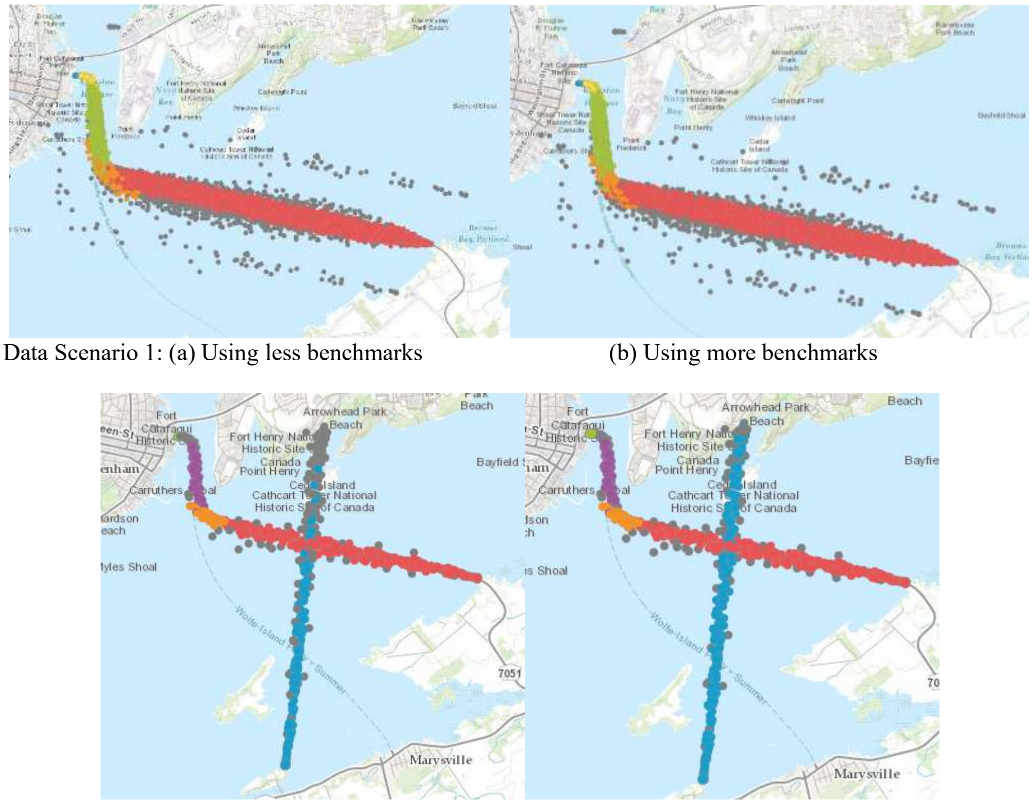
Figure	ε and *minPts*	Clustered Results - Number of Points
2a	ε = 0.2; *minPts* = 30	Cluster 1 (red): 39593; Outliers: 298;
2b	ε = 0.2; *minPts* = 30	Cluster 1 (red): 25518; Cluster 2 (blue): 9337; Cluster 3 (green): 3178; Cluster 4 (orange): 214; Cluster 5 (yellow): 166; Outliers: 1478;
2c	ε = 0.1; *minPts* = 40	Cluster 1(red): 1630; Outliers: 0;
2d	ε = 0.1; *minPts* = 40	Cluster 1 (red): 635; Cluster 2 (blue): 606; Cluster 3 (green): 233; Cluster 4 (purple): 87; Outliers: 37;

Table 1. Clustering parameters and clustered result

while Figures 3(b) and 3(d) are the results with more check points prepared. As presented in Table 2, with additional prior knowledge, more points can be labelled. As shown in both Figure 4(a) and 4(b), in both data scenarios, less iterations are needed for completing the clustering task and the algorithms can be more time efficiency.

4.3 Discussion

Comparing to the traditional DBSCAN, the use of the Mahalanobis distance metric detects more points as outliers. Traditional DBSCAN enables all core points to grow the cluster, thus the points close to the cluster are high likely to be swallowed. By contrast, the Mahalanobis metric calculates the correlation between a point and a group of points. Thus, even if some points are very close to each other in 2D space, as long as the point is somehow deviated from the main trajectory, they may have small correlation and thus cannot be grouped together. This can be observed from Figures 3(a) and 3(b), where a lot of outliers are very close to Cluster 1 (red). Since the Mahalanobis distance metric calculates the correlation, the result is very sensitive to the check points and parameter setting. The pre-defined cluster can almost handle the correlations. This can be explained by the Figures 4(a) and 4(b), where it is observed that more than 90% points are clustered in the first iteration. The clusters are relatively small and are updated starting from the second iteration. The pre-defined cluster describing the port (Cluster 2 in Figures 3a and 3b) are circle-shaped, then the points outside of the port circle boundary will have low correlation and will not be clustered.



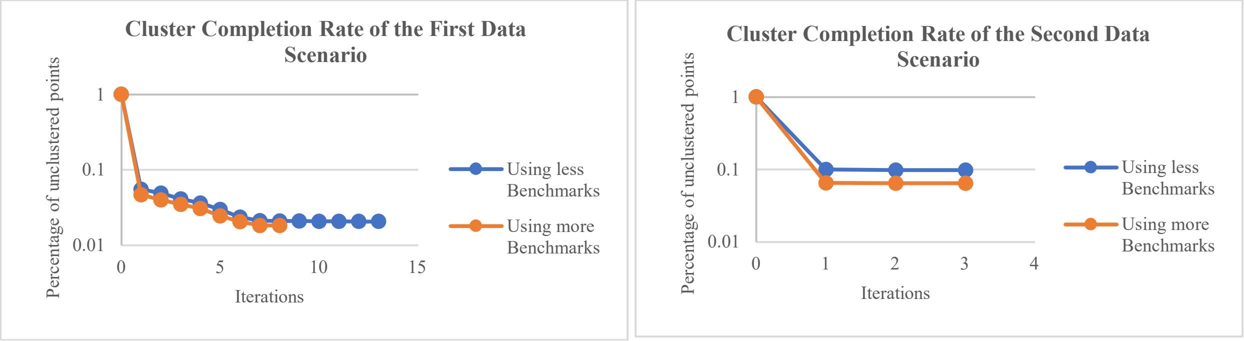
Data Scenario 1: (a) Using less benchmarks (b) Using more benchmarks

Data Scenario 2: (c) Using less benchmarks (d) Using more benchmarks

Figure 3. Results comparison of DBSCAN using the Mahalanobis distance metric

Figure	ε	Pre-defined benchmark clusters (Number of Points)	Clustered Result (Number of Points)	Run Time (s)
3a	ε=2	Cluster 1 (red): 255; Cluster 2 (blue): 93; Cluster 3 (green): 31; Cluster 4 (orange): 30; Cluster 5 (yellow): 30;	Cluster 1 (red): 25190; Cluster 2 (blue): 8175; Cluster 3 (green):3256; Cluster 4 (orange): 761; Cluster 5 (yellow): 1144; Outliers: 825;	247.12
3b	ε=2	Cluster 1 (red): 1275; Cluster 2 (blue): 466; Cluster 3 (green): 158; Cluster 4 (orange): 50; Cluster 5 (yellow): 50;	Cluster 1 (red): 25389; Cluster 2 (blue): 8815; Cluster 3 (green): 3225; Cluster 4 (orange): 704; Cluster 5 (yellow): 1030; Outliers: 728;	207.58
3c	ε=4	Cluster 1 (red): 63; Cluster 2 (blue): 60; Cluster 3 (purple): 20; Cluster 4 (orange): 15; Cluster 5 (green): 23;	Cluster 1 (red): 597; Cluster 2 (blue): 509; Cluster 3 (purple): 83; Cluster 4 (orange): 42; Cluster 5 (green): 210; Outliers: 157;	6.25
3d	ε=4	Cluster 1 (red): 317; Cluster 2 (blue): 303; Cluster 3 (purple): 43; Cluster 4 (orange): 20; Cluster 5 (green): 116;	Cluster 1 (red): 593; Cluster 2 (blue): 567; Cluster 3 (purple): 81; Cluster 4 (orange): 40; Cluster 5 (green): 214; Outliers: 103;	5.71

Table 2. Clustering parameters and clustered result



(a) (b)

Figure 4. Cluster completion rate of the two data scenarios

The pre-defined cluster describing the trajectories are normally line-shaped, and only the points on the same direction can be grouped into this cluster. Since this algorithm is not able to generate clusters by its own and still depends on pre-defined clusters, the check points are very crucial for the clustering performance. Besides, well pre-defined check points can improve the clustering cost efficiency. Figures 4(a) and 4(b) show that with more benchmarks, the algorithm can cluster more points in less iterations. The future work of the algorithm developing includes find optimal ε automatically and clustering without prior knowledge. As summary, the result demonstrates the proposed algorithm has better performance to detect the outliers and distinguish the crossing trajectories with smaller computational costs.

5. CONCLUSIONS AND FUTURE WORK

This paper presented the details of enhancing the traditional DBSCAN clustering method by incorporating the Mahalanobis distance metric and how the proposed algorithm can be applied on marine trajectory clustering. The high dimensional data have different 'density reachability' than 2D space, thus new findings and knowledge can be discovered by clustering them. The Mahalanobis distance metric calculating the correlations between a point to a group of points enhanced the clustering process for the points with more similarities. Furthermore, after finding benchmarks of the clusters, time efficiency of the algorithms can be much improved for clustering the new data. Overall, this paper demonstrated the effectiveness of proposed enhanced DBSACN method. The results provide key important insights to marine transportation route planning, marine transportation monitoring, finding abnormal routes, reduce accident risks and providing foundations for autonomous vessels. However, there are still some limitations remains on the proposed method. In this paper, the proposed algorithm requires some prior knowledge about the data and pre-defined ε and check points. Having an automatics way of defining the latter parameters is for future investigation.

ACKNOWLEDGEMENTS

This work is financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC Discovery grants) and York University.

REFERENCES

Bian, J., Tian, D., Tang, Y., Tao. D., 2018. A survey on trajectory clustering analysis. DOI: https://arxiv.org/abs/1802.06971

Cho, K., Chen, X., 2014. Classifying and visualizing motion capture sequences using deep neural networks, *Computer Vision Theory and Applications (VISAPP), 2014 International Conference* on, Vol. 2, IEEE, 2014, pp. 122–130

Esmaelnejad J., Habibi J., Yeganeh S.H., 2010. A Novel Method to Find Appropriate ε for DBSCAN. *Intelligent Information and Database Systems. ACIIDS 2010. Lecture Notes in Computer Science*, vol 5990. Springer, Berlin, Heidelberg

Ester, M., Kriegel, H.P., Sander, J., Xu. X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of *the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD).* 226–231

Ferreira, N., Klosowski, J.T., Scheidegger, C.E., Silva, C.T., 2013. Vector field k-means: Clustering trajectories by fitting multiple vector fields, *Computer Graphics Forum,* Vol. 32, Wiley Online Library, 2013, pp. 201–210

Galluccio, L., Michel, O., Comon, P., Hero, A.O., 2012. Graph based k-means clustering, *Signal Processing* 92 (9) (2012) 1970–1984.

Gao, Y.J., Li, C. Chen, G.C., Chen, L., Jiang, X.T., Chen, C., 2007. Efficient k-nearest-neighbor search algorithms for historical moving object trajectories, *Journal of Computer Science and Technology* 22 (2) (2007) 232–244.

Gurung, S., Lin, D., Jiang, W., Hurson, A., Zhang, R., 2014. Traffic information publication with privacy preservation, *ACM Transactions on Intelligent Systems and Technology (TIST)* 5 (3) (2014) 44.

Hall, A.M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten. H.I., 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations11*, 1 (2009), 10–18.DOI: http://dx.doi.org/10.1145/1656274.1656278

Hou, J., Gao, H., Li, X., 2016. DSets-DBSCAN: A Parameter-Free Clustering Algorithm, *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3182-3193, July 2016.

Karami, A., Johansson, R., 2014. Choosing DBSCAN Parameters Automatically using Differential Evolution, *International Journal of Computer Applications*, Volume 91 - No. *, April 2014

Laxhammar, R., Falkman, G., 2014. Online learning and sequential anomaly detection in trajectories, *IEEE transactions on pattern analysis and machine intelligence* 36 (6) (2014) 1158–1173.

Lee, J.G., Han, J., 2007. Trajectory Clustering: A Partition-and-Group Framework, *2007 ACM SIGMOD international conference on Management of data*, June 2007 Pages 593–604, https://doi.org/10.1145/1247480.1247546

Li, X., Hu, W., Hu, W., 2006. A coarse-to-fine strategy for vehicle motion trajectory clustering, *18th International Conference on Pattern Recognition (ICPR'06),* Vol. 1, IEEE, 2006, pp. 591–594.

Liu, B., 2015. Maritime Traffic Anomaly Detection from AIS satellite Data in Near Port Regions. http://hdl.handle.net/10222/60105

MarineCadastre.gov, Vessel Traffic Data, marinecadastre.gov/ais/.

Marković, N., Sekuła, P., Laan, Z.V., Andrienko, G., 2019. Applications of Trajectory Data in Transportation: Literature Review and Maryland Case Study, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1858-1869, May 2019.

Marr, B., 2019. The Incredible Autonomous Ships of The Future: Run by Artificial Intelligence Rather Than A Crew, *Forbes, Forbes Magazine.*

Merkel, D., 2019. Autonomous Ships, Opportunities; Challenges, *MarineLink, Maritime Activity Reports, Inc*.

Mikhail, K., Loris, F., Christian, K., Alexei, P., Vadim, T., Devis, T., 2009. Machine learning models for geospatial data

Piciarelli, C., Micheloni, C., Foresti, G.L., 2008. Trajectory-based anomalous event detection, *IEEE Transactions on Circuits and Systems for video Technology* 18 (11) (2008) 1544–1554

R Core Team. 2015. R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing*. Retrieved fromhttp://www.r-project.org/.

Ren, Y., Liu, X., Liu, W., 2012. DBCAMM: A novel density-based clustering algorithm via using the Mahalanobis metric, *Applied Soft Computing*, Volume 12, Issue 5, 2012, Pages 1542-1554, ISSN 1568-4946

Rolls, 2018. Royce and Finferries Demonstrate World's First Fully Autonomous Ferry

Safety of Life at Sea (SOLAS) convention Chapter V. Regulation 19.

Sangeetha, M., Padikkaramu, V., Chellan, R.T., 2017. A Novel Density Based Clustering Algorithm by Incorporating Mahalanobis Distance, *International Journal of Intelligent Engineering & Systems*

Sawant, K., 2014. Adaptive Methods for Determining DBSCAN Parameters, *IJISET-International Journal of Innovative Science, Engineering & Technology*, Vol. 1 Issue 4.

Schubert, E., Koos, A., Emrich, T., Züfle, A., Schmid, K.A., Zimek, A., 2015. A framework for clustering uncertain data. *Proceedings of the VLDB Endowment 8,* 12 (2015), 1976–1979. DOI: http://dx.doi.org/10.14778/2824032.2824115

Sheng, P., Yin, J., Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. *Sustainability*. 10. 2327. 10.3390/su10072327.

Silveira, P., Teixeira, A., Soares, C., 2013. Use of AIS Data to Characterise Marine Traffic Patterns and Ship Collision Risk off the Coast of Portugal. *Journal of Navigation*, 66(6), 879-898. doi:10.1017/S0373463313000519

Smiti, A., Eloudi, Z., 2013. Soft DBSCAN: Improving DBSCAN clustering method using fuzzy set theory, *2013 6th International Conference on Human System Interactions (HSI),* Sopot, 2013, pp. 380-385.

Vespe, M., Visentini, I., Bryan, K., Braca, P., 2012. Unsupervised learning of maritime traffic patterns for anomaly detection. *IET Conference Publications*. 2012. 1 -5. 10.1049/cp.2012.0414.

Xia, L., Jing, J., 2009. SA-DBSCAN: A self-adaptive density-based clustering algorithm, *Journal of the Graduate School of the Chinese Academy of Sciences*

Xiang, T., Gong, S., 2008. Spectral clustering with eigenvector selection, *Pattern Recognition* 41 (3) (2008) 1012–1029.

Zheng, Y., Zhou, X., 2011: *Computing with Spatial Trajectories*, ISBN: 978-1-4614-1628-9