2014 IEEE International Conference on Big Data

Knowledge-based Clustering of Ship Trajectories Using Density-based Approach

Bo Liu*, Erico N. de Souza*, Stan Matwin*† and Marcin Sydow‡†
* Faculty of Computer Science, Dalhousie University, Canada
† Institute for Computer Science, Polish Academy of Sciences, Poland
‡ Polish-Japanese Institute of Information Technology, Warsaw, Poland
Email: boliu@dal.ca, erico.souza@dal.ca, stan@cs.dal.ca, marcin.sydow@ipipan.waw.pl

***Abstract*—Maritime traffic monitoring is an important aspect of safety and security, particularly in close to port operations. While there is a large amount of data with variable quality, decision makers need reliable information about possible situations or threats. To address this requirement, we propose extraction of normal ship trajectory patterns that builds clusters using, besides ship tracing data, the publicly available International Maritime Organization (IMO) rules. The main result of clustering is a set of generated lanes that can be mapped to those defined in the IMO directives. Since the model also takes non-spatial attributes (speed and direction) into account, the results allow decision makers to detect abnormal patterns - vessels that do not obey the normal lanes or sail with higher or lower speeds.**

***Keywords*-clustering; trajectory mining; maritime surveillance; rule mapping**

I. Introduction

Automatic Identification System (AIS) is an automatic tracking system for identifying and locating vessels by exchanging data with other nearby ships, AIS base stations and satellites. The use of this system has been required by the International Maritime Organization (IMO) since 2004 and over 400,000 ships worldwide have installed these transponders [1]. IMO also is responsible for establishing the navigational lanes that vessels must use when approaching ports around the world. Unfortunately, current tools do not offer automatic ways to check whether the vessels are obeying these lanes. This work proposes to use the IMO rules in clustering of the trajectories in close to port operations. The first step is to adjust the parameters of the density-based clustering algorithm to find clusters that are close to the main routes defined by IMO. Such clustering algorithm also takes into account non-spatial attributes, like speed and direction, when building clusters. The second step is to extract the center of those clusters, which we call the Gravity Vector (GV). This approach also has the advantage to detect extra tracks not originally defined by IMO rules.

One work which provides mechanisms to manage traffic in the ocean that is similar to ours is an unsupervised framework called TREAD (Traffic Route Extraction and Anomaly Detection), proposed by Pallotta $et\ al.$. [2]. It is designed to detect low likelihood behaviors and predict future vessel positions from maritime traffic data. TREAD is a point-based framework and it has a practical advantage to handle trajectories of unequal length or with gaps, which is common due to the low satellite coverage of AIS around the world. The main difference in relation to TREAD relies on the fact that our work tries to directly map the rules defined by IMO into real data available from AIS readings.

Other works like [3] and [4] are also based on clustering to detect main lanes used by vessels. Both suffer from the same issue: clusters built do not consider speed and direction. This work uses this information, and it also takes IMO rules in consideration, which is not the case in previous works.

Among various types of clustering approaches, density-based clustering algorithms can be most suitable for this work. One representative method is DBSCAN (Density-based spatial clustering of applications with noise) [5]. It can provide good results in many cases since it can discover clusters of arbitrary shape and can filter out noise [6]. In our work we extend DBSCAN to adopt other non-spatial attributes, such as speed and direction, to find normal traffic patterns from moving trajectory dataset.

The rest of the paper is organized as follows. Section 2 proposes our normal traffic patterns extraction model. Section 3 presents the results of experimental evaluation. In the final section, we conclude with a summary and discuss our method's limitations and the potential future work.

II. Normal Traffic Patterns Extraction Model

Vessels always follow different movement patterns in different areas. For instance, cargo ships may travel along straight lines at high speed in the middle of the sea, while they may frequently adjust their directions at low speed in the port or offshore platform areas. As a consequence, Spaccapietra [7] introduced a model for reasoning about trajectories, which is called $Stops\ and\ Moves$. And in the work done by [2][8], maritime trajectories analysis is conducted with this approach as well, moving and stopping patterns distinguished by a threshold of speed. In our framework, we employ a similar method to extract different normal patterns from the historical AIS dataset with a stopping SOG (Speed Over Ground) threshold of 0.5 knot. Instead of merely clustering the waypoints of the trajectory data as [2][8], whole normal moving trajectories and arbitrary shapes of stopping regions will be extracted based on the entire moving and stopping dataset in the specific area respectively.

A. Representing Trajectory Data

A trajectory can be represented by a multidimensional time series [9] or a sequence of multi-dimensional points [3]. And in this paper, we modify the definition of its raw form [10] to adapt it for our work. And in the definition, COG and SOG are Course Over Ground and Speed Over Ground respectively.

Definition A trajectory is a finite sequence T = $((x_1, t_1), (x_2, t_2), ..., (x_m, t_m))$ where x_i is a set of $< Latitude, Longitude, COG, SOG >$ and t_i is the timestamp.

B. Normal Moving Trajectories Extraction

Compared to stopping patterns, the moving part plays a more important role in the anomaly detection task since there is a higher collision risk for a vessel sailing at high speed than at low speed in the stopping areas. To extract normal patterns from the moving AIS data, we propose a new clustering algorithm DBSCANSD (Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction).

DBSCANSD (shown in Algorithm 1) is a density-based clustering algorithm based on the algorithm DBSCAN [5]. The main observation behind our approach is that it is common for different types of ships to sail with different velocities in one similar area of the sea. For instance, the cruise speed of a cargo ship can be faster than a fishing ship. Even the same type of vessels can behave differently in relation to direction. Imagine that an oil tanker sails between two different countries. When the vessel is full, its speed is slower than when it is empty.

As we know, the key idea of DBSCAN [5] is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points. Here in our paper, we adopt this idea but also consider other two factors, maximum speed variance (MaxSpd) and maximum direction variance (MaxDir). The intuition behind this is that the neighbors of a point should be not only near enough, but also with similar COG (Course Over Ground) and SOG (Speed Over Ground). Thus, we can modify the definition of Eps-neighborhood in [5] to the following:

Definition Given a database D of moving trajectory points in a specific area, the $Eps - neighborhood$ of a trajectory point p, denoted by $N_\epsilon(p)$, is defined by $N_\epsilon(p) = \{q \in D \mid dist(p, q) < \epsilon$ and $|p.SOG - q.SOG| < MaxSpd$ and $|p.COG - q.COG| < MaxDir\}$

Note that $dist(p, q)$ is the Geographical Distance [11] between p and q, instead of Euclidian distance, because it is necessary to take in consideration the Earth's curvature to calculate distances.

Algorithm 1 DBSCANSD

procedure DBSCANSD($DatasetM$, eps, $MinPts$, $MaxDir$, $MaxSpd$)
  Mark all points in moving dataset $DatasetM$ as unclassified
  $clusterList \leftarrow$ empty list
  for each unclassified point P in $DatasetM$ **do**
    Mark P as classified
    $neighborPts \leftarrow$ queryNeighborPoints ($DatasetM$, P, eps, $MinPts$, $MaxDir$, $MaxSpd$)
    if $neighborPts$ is not $NULL$ **then**
      $clusterList$.add($neighborPts$)
  for each cluster C in $clusterList$ **do**
    for each cluster C' in $clusterList$ **do**
      if C and C' are different clusters **then**
        if mergeClusters(C, C') is TRUE **then**
          $clusterList$.remove(C')
  return $clusterList$
procedure QUERYNEIGHBORPOINTS($data$, P, eps, $MinPts$, $MaxDir$, $MaxSpd$)
  $cluster \leftarrow$ empty list
  for each point Q in data **do**
    if distance(P,Q) $< eps$ **then**
      if $|P.SOG - Q.SOG| < MaxSpd$ **then**
        if $|P.COG - Q.COG| < MaxDir$ **then**
          $cluster$.add(Q)
  if $cluster$.size $> MinPts$ **then**
    Mark P as core point
    return $cluster$
  return $NULL$
procedure MERGECLUSTERS($clusterA$, $clusterB$)
  $merge \leftarrow FALSE$
  for each point Q in $clusterB$ **do**
    if point Q is core point and $clusterA$ contains Q **then**
      $merge \leftarrow TRUE$
      for each point Q' in $clusterB$ **do**
        $clusterA$.add(Q')
      break
  return $merge$

Algorithm 1 presents that DBSCANSD requires 5 parameters as input. $DatasetM$ is a list of all the moving points in the trajectories. eps and $MinPts$ are the reachable distance and reachable minimum number of points (see Ester *et al.* [5]). $MaxDir$ and $MaxSpd$ are the two new parameters (maximum direction variance and maximum speed variance). The time complexity of our algorithm is $O(n^2)$ where n is the size of $DatasetM$, which is same as that of DBSCAN [5].

We used external knowledge based on the rules defined by IMO to adjust the parameters of the algorithm; therefore, we are using the lanes defined in IMO publications to fit the clustering results. As we know, once the area is determined, the two parameters $MaxDir$ and $MaxSpd$ should be fixed. Multiple pairs of values were tested to find the optimal $MaxDir$ and $MaxSpd$. Our experiments indicate that 5° and 5 knots are the best parameters for both tested ports.

On the other hand, the parameters eps and $MinPts$ must be defined for each port. Thus, another set of experiments were done to select the best pair for each area. For instance, when a lane generated by the clustering algorithm has any gaps compared to the one in the IMO map, the analyst must increase the value of eps or decrease that of $MinPts$. And the two parameters eps and $MinPts$ selected in this procedure will be directly used in the next stopping area extraction phase.

After applying DBSCANSD to the ship trajectory dataset, geographically close trajectory points with similar direction and speed will be grouped together to form a cluster. Then, we can not only get arbitrary shapes of clusters, but we can also separate those close points with different normal SOGs and COGs into multiple clusters. Moreover, the algorithm can even treat a ship's acceleration or deceleration as a cluster, as long as the $MaxSpd$ is well defined.

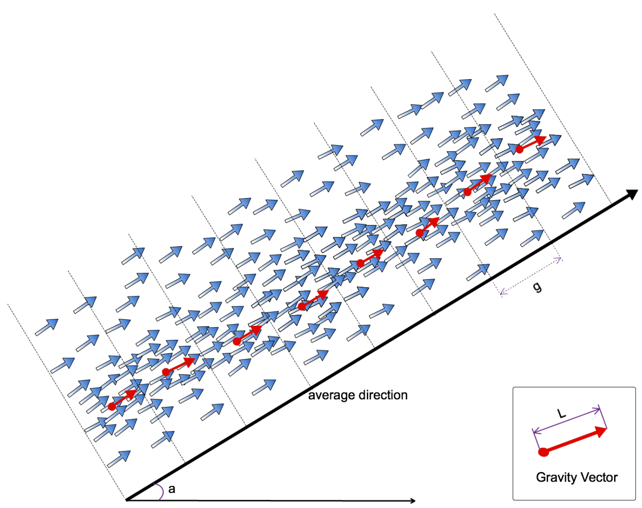

Figure 1. The process of calculating Gravity Vectors (GV) for one moving cluster. Blue arrows stand for the points of the cluster, red arrows are the Gravity Vectors. The length(L) of the GV is the SOG_{avg} of the GV. The width of a grid g is the pre-defined length for partitioning the points.

Although the clustering results can reflect normal patterns of the vessels, it is not feasible to employ all the points in every cluster to decide the abnormality of the new coming trajectory data in terms of the time complexity. The idea of centroid and envelope has been used for path modeling in vision-based trajectory learning [12]. The centroid can minimally specify the corresponding path and the envelope is used for denoting the path extent.

In our case, we adopt a similar idea as centroid and envelope. Here we call it **Gravity Vector (GV)**. A GV is extracted by partitioning a cluster in multiple parts, therefore, each cluster will have multiple GVs. The grid width (g) can be decided based on the domain knowledge or multiple experiments' results. Here in this paper, we choose the length of eps used in Algorithm 1 as that of g. A Gravity Vector is a vector formed by 5 features: average COG, average SOG, average Latitude, average Longitude and Median Distance. Then one Gravity Vector GV_i can be denoted by: $GV_i =< COG_{avg}, SOG_{avg}, LAT_{avg}, LON_{avg}, D_{median} >$.

Figure 1 presents an example of calculating the gravity points of one cluster. The steps of this are as following:

Step (1) Calculate the average COG of the whole cluster and let it be COG_w. Note that COG_w is not COG_{avg}.

Step (2) Partition all the trajectory points in the cluster along the direction of COG_w by a pre-defined grid width (g in Figure 1).

Step (3) For each grid of the partitioning results, generate the gravity vector of the grid.

In step (3), assume there are k trajectory points in one grid and TP_i is the ith trajectory point, we can calculate the first 4 features using the following formulas: $COG_{avg} = \frac{\sum_{i=1}^{k} TP_i.COG}{k}$, $SOG_{avg} = \frac{\sum_{i=1}^{k} TP_i.SOG}{k}$, $LAT_{avg} = \frac{\sum_{i=1}^{k} TP_i.Latitude}{k}$, $LON_{avg} = \frac{\sum_{i=1}^{k} TP_i.Longitude}{k}$.

Then to calculate the last feature D_{median}, we should first calculate the distances between all k points in the grid and the geographical point (LAT_{avg}, LON_{avg}) generated by the above equations. After this, we can apply a linear time complexity algorithm for computing median D_{median} based on Hoare's PARTITION algorithm [13].

In Lemma 2.1, we present the time complexity of calculating GVs for a cluster.

Lemma 2.1: The worst time complexity of calculating a specific cluster's Gravity Vectors is O(n), where n is the total number of points in the cluster.

Proof: Assume there are n points in one cluster and the cluster is partitioned into m grids. Step (1) takes O(n) time to calculate the average COG of the whole cluster. Step (2) takes O(n) time to map all the points to the axis of average COG and O(n) time to partition them into m grids. In step(3), to calculate one particular GV for the ith grid with k_i points, it takes $O(k_i)$ time to calculate the first 4 features and $O(k_i)$ time to calculate the median distance in linear time using one of the algorithms based on the Hoare's partition algorithm [13], for example the Blum-Floyd-Pratt-Rivest-Tarjan algorithm [14] that has linear worst time.

And this procedure needs to be repeated for m times, the total time for step (3) can be calculated as following:

$$W(n) = \sum_{i=1}^{m} O(k_i) = O(n)$$

Thus the pessimistic total time complexity for the whole procedure from step (1) to step (3) is O(n).

■

The feature median distance can be used to measure the width variance of a moving cluster. In the work done by Etienne $et\ al.$ [15], it also shows the choice of the statistical decile used to compute the spatio-temporal channel can give a tolerable estimate of this channel's width. However, we use median rather than ninth decile in [15] to provide a more robust width estimation metric. And the distance is not the exact width of the channel, instead, it is a relative distance for outlier detection.

C. Normal Stopping Areas Extraction

In stopping areas, such as ports and wharfs, there may be different types of ships at anchor with zero-velocity or new coming vessels entering the area with extremely low speed. In this case, vessels can stop with their prows pointing to any directions and change their headings frequently to arrive at the anchorage berths. Therefore, direction is no longer an essential part for analyzing the risk of collision. The possibility of a low-speed vessel colliding with another ship is very low; therefore this situation was excluded from the clustering process. In other words, only when a ship sails with a higher speed than the stopping threshold in a stopping area can we label the data as outlier. Based on the analysis of this region, the original DBSCAN [5] algorithm can be employed for stopping points clustering.

It should be noted that it is common to have both stopping clusters and moving clusters in a specific region at the same time, e.g., a shipping lane can pass through a stopping area. Our approach can handle these kinds of issues because both types of clusters can be generated during the normal traffic patterns extraction phase and the abnormality of a point will be decided based on both types clusters.

Although DBSCAN [5] can generate stopping points clusters, these original clustering results are still not adaptable for the next anomaly detection phase in terms of the large amount of points in each cluster. Consequently, just like GV, a similar representative point called **Sampled Stopping Point (SSP)** is proposed. Algorithm 2 shows the whole procedure for generating the final SSPs in a specific region.

As shown in Algorithm 2, stopping points are first clustered by the DBSCAN algorithm (line 1). The same parameters (eps and $MinPts$) used by DBSCANSD are still employed in this stopping areas extraction phase. Then the algorithm starts to generate the SSPs for every cluster. Lines 4 and 5 calculate the four border values of the cluster. Border values are extreme values of one cluster. The Up Border Value is the maximum of all points' Latitude values while Down Border Value is the minimum. While the Left and Right Border Values are the minimum and maximum of all points' Longitude values. It is obvious that any two or three of them (or even all the four points) can be the same one point. It is noteworthy that if one cluster crosses the Greenwich meridian, the way to calculate the two border values, $lon1$ and $lon2$, will be changed.

Algorithm 2 Extract SSP From Stopping Dataset

Input: The list of the stopping points of the region, $DatasetS$; Reachable distance, eps; Reachable minimum number of points, $MinPts$

Output: The list storing the region's SSP, $resultSSP$

1: $StoppingPointsClusters \leftarrow$ DBSCAN($DatasetS$, eps, $MinPts$) ▷ see DBSCAN algorithm in [5]
2: $resultSSP \leftarrow$ empty list
3: **for** each cluster C in $StoppingPointsClusters$ **do**
4: $lat1, lat2 \leftarrow$ minimum and maximum of all the points' Latitude values in cluster C
5: $lon1, lon2 \leftarrow$ minimum and maximum of all the points' Longitude values in cluster C
▷ estimate the sample size for cluster C
6: $area \leftarrow |(lat1 - lat2) * (lon1 - lon2)|$
7: **if** $area$ = 0 **then**
8: $sample\_size \leftarrow 1$
9: **else**
10: $sample\_size \leftarrow$ Ceiling($area/(\pi * eps^2)$)
▷ sample the stopping area points
11: $count \leftarrow 0$
12: **while** $count < sample\_size$ **do**
13: randomly select one point P from cluster C
14: **if** P is far from all points in $resultSSP$ **then**
15: $resultSSP$.add(P)
16: $count$++
17: **return** $resultSSP$

After extracting the four border values, the area of the rectangle defined by the border values is calculated. With this area result and the eps used for clustering, the sampling number is estimated (lines 6−10). Lastly, we use a random selection strategy to sample the stopping points. However, there still exist some sub-regions with higher density of points compared with other sub-regions in the same cluster and as a result, random selection tends to select points from these higher-density micro-regions. So we need to modulate the sampling process to give each sub-region equal probability to be selected (lines 11−16). The distance condition in line 14 can be defined by eps as well. The algorithm has linear time complexity.

III. Experiments

In this section, we evaluate the effectiveness of our maritime normal patterns detection model in two regions. The data set was provided proprietarily, therefore this data is not public. It contains non-anonymized messages from ships in the region of Juan de Fuca Strait and the region of Los Angeles Long Beach. And the movement rules for port areas defined by IMO from the IMO publication [16] are shown in Figure 2 and 3 and presented with OpenSeaMap[1].

[1]http://www.openseamap.org

This strategy is to adjust the parameters of the density-based clustering algorithm to find clusters that are close to the rules and then generate the clusters' GVs and SSPs.

A. Juan de Fuca Strait

The Strait of Juan de Fuca separates the south coast of Vancouver Island from the north coast of State of Washington [17]. Figure 2 shows a map of this area and the predefined rules for it. The IMO rules are presented with arrows and dotted lines in purple color.

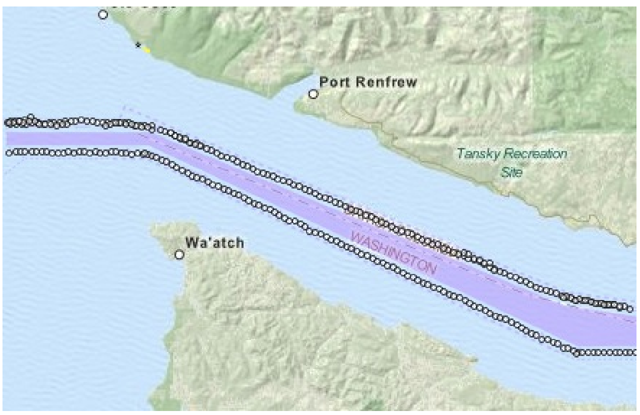

Figure 2. GVs and SSP extracted in the area of Juan de Fuca Strait [16]

The data set prepared for this experiment is two months of trajectory data from November 1 to December 31 in 2012. It consists of 67,850 trajectory points. The whole dataset is not used; instead 46,000 records (40,000 moving points and 6,000 stopping points distinguished by the SOG threshold 0.5 knot) are selected.

After applying the two clustering algorithms, 14 different clusters including 13 moving clusters and 1 stopping cluster are extracted. And the total number of the points composing the 13 moving clusters is 20,817 while the total number of the points composing the stopping cluster is 4,800.

Figure 2 shows the GVs of the moving clusters and the SSP of the stopping cluster in black empty circles. After this extraction phase, only 302 points are generated which include 301 GVs and only one SSP. From the figure we can see that the clustering results can reflect the traffic trends in this strait and the GVs can match IMO rules perfectly too.

B. Los Angeles Long Beach

The approach has also been tested in the area of Los Angeles Long Beach, the traffic of which is more complicated and heavier. Figure 3 shows a map of the area with rules defined. In this experiment, the same period (November 1 to December 31 in year 2012) dataset has been prepared. There are 327,694 records (99,937 moving points and 227,757 stopping points) in this dataset. We adopt all the moving points in this dataset for moving clusters generation and the first 20,000 stopping points for stopping area generation.

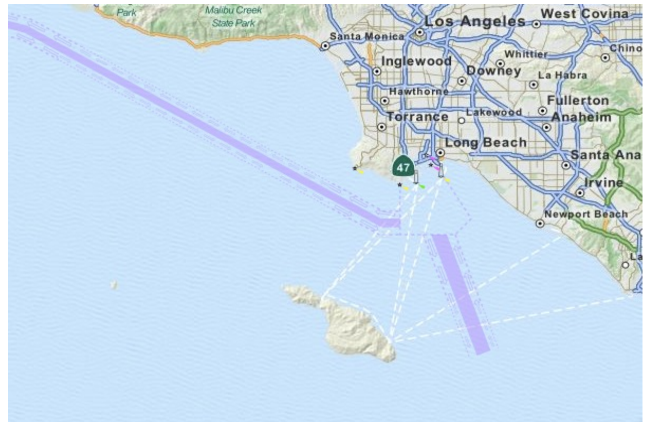

Figure 3. Map of Los Angeles Long Beach and the rules defined [16]

During stopping clusters sampling procedure, 7 different stopping clusters with 19,089 stopping points were generated by DBSCAN algorithm [5]. Then after applying Algorithm 2, only 26 SSPs (shown with filled circle in dark green color in Figure 4) are selected with excellent quality in terms of the representativeness.

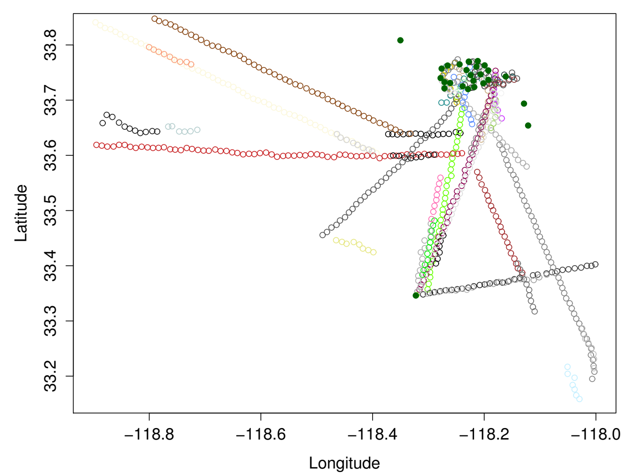

Figure 4. The GVs and SSPs extracted from the clusters in Los Angeles Long Beach area.

Then the algorithm DBSCANSD is applied to moving points. Finally, 48,404 points are selected to form 51 moving clusters from the original 99,937 points. The last step for this normal traffic pattern extraction is to calculate the GVs for the clusters. The GVs of the area can be seen in Figure 4 and from the figure we can see that in the stopping area, there are multiple moving clusters too. This may be because that vessels in the area have to change their headings frequently to arrive at the specified anchor location and our algorithm will treat this curve-shape movement as multiple moving clusters with slight COG differences.

IV. Discussion And Conclusions

This paper presented a method based on clustering to associate IMO rules with real vessel trajectories. The proposed method also allows detection of moving and stopping areas. The main advantage of our work is that two non-spatial attributes, speed and direction, are taken into account during the clustering phase. In this way, those geographically close trajectory points with similar direction and speed can be grouped together to form a cluster. The approach also extracts the representative center of the cluster for each region: Gravity Vector for moving clusters and Sampled Stopping Point for stopping clusters.

To show the effectiveness of our approach, two real data sets from different regions are used. We have compared the generated results with the rules defined by IMO and show that the results can be successfully mapped to the rules.

One limitation is that the method is sensitive to parameters. The model requires four parameters during the moving traffic patterns extraction phase and two parameters for stopping area generation. So, if the model is applied in another area, a new set of parameters need to be given to achieve a good clustering result. Besides, the framework also relies on the users' domain knowledge. For instance, when one needs to distinguish moving points from stopping points in the first step, a reasonable SOG threshold is required.

One of possible directions of future work is to improve the efficiency of the presented algorithms by applying more sophisticated data structures (like spatial indexes, for example). This issue is important in the context of big sizes of the processed data. In this paper the focus was not on optimising the efficiency of the algorithms but to present the ideas. We envisage optimisation of the algorithms in the continuation work.

The Gravity Vectors and Sampled Stopping Points detected by the proposed approach can help authorities update their rules (e.g. re-position the buoy markers) and support anomaly detection work, which is also one of our future directions. And some other research studies like route planning and vessel position prediction can also be conducted based on our model's results.

Acknowledgment

The authors acknowledge the generous support of GSTS, Inc., ExactEarth, Inc. and the Natural Science and Engineering Research Council of Canada for this research.

References

[1] H. Ball, *Satellite AIS for Dummies*. Mississauga, ON: Wiley, 2013.

[2] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 6, pp. 2218–2245, 2013.

[3] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD Inter. Conf. on Management of Data*, ser. SIGMOD '07, 2007, pp. 593–604.

[4] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, and C. T. Silva, "Vector field k-means: Clustering trajectories by fitting multiple vector fields," *CoRR*, vol. abs/1208.5801, 2012.

[5] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[6] J. Han, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[7] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data Knowl. Eng.*, vol. 65, no. 1, pp. 126–146, Apr. 2008.

[8] M. Vespe, I. Visentini, K. Bryan, and P. Braca, "Unsupervised learning of maritime traffic patterns for anomaly detection," in *Data Fusion & Target Tracking Conf. (DF&TT 2012): Algorithms & Applications, 9th IET*. IET, 2012, pp. 1–5.

[9] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. CRC Press, 2013.

[10] L. Rikard, "Anomaly detection in trajectory data for surveillance applications," *Studies from the school of science and technology at rebro university 19*, 2011.

[11] C. Veness. Calculate distance, bearing and more between latitude/longitude points.

[12] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance." *IEEE Trans. Circuits Syst. Video Techn.*, no. 8, pp. 1114–1127.

[13] C. A. R. Hoare, "Algorithm 63: Partition," *Commun. ACM*, vol. 4, no. 7, pp. 321–, Jul. 1961.

[14] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, "Time bounds for selection," *J. Comput. Syst. Sci.*, vol. 7, no. 4, pp. 448–461, Aug. 1973.

[15] L. Etienne and T. Devogele, "Spatio-temporal trajectory analysis of mobile objects following the same itinerary," *Advances in Geo-Spatial Information Science*, vol. 17, no. 1, pp. 11–34, 2012.

[16] IMO, *Ships' Routeing*. International Maritime Organization, 2013.

[17] U. S. D. M. A. H. Center, *Sailing Directions (enroute) British Columbia*, ser. Pub. (United States. Defense Mapping Agency. Hydrographic/Topographic Center), 2012.