Using Machine Learning for Unsupervised Maritime Waypoint Discovery from Streaming AIS Data

Andrej Dobrkovic
Centre for Telematics and Information Technology
University of Twente
Enschede, The Netherlands
+31534894144
a.dobrkovic@utwente.nl

Maria-Eugenia Iacob
Centre for Telematics and Information Technology
University of Twente
Enschede, The Netherlands
+31534894134
m.e.iacob@utwente.nl

Jos van Hillegersberg
Centre for Telematics and Information Technology
University of Twente
Enschede, The Netherlands
+31534893513
j.vanhillegersberg@utwente.nl

ABSTRACT

Estimating the future position of a deep sea vessel more than 24 hours in advance is a major challenge for Dutch logistics service providers (LSPs). Their unscheduled arrival in ports directly impacts scheduling and waiting times of barges, propagating throughout the entire supply chain network. To help LSPs' planners improve planning operations, we intend to capture the characteristics of maritime routes for a specific region (the North Sea connecting the Netherlands and United Kingdom) in the form of a directed graph, which can be used as a foundation for predicting destination and arrival time of each associated vessel. To create such graph we need an efficient way to extract waypoints for traffic data and this is the problem we will address in this paper.

Since LSPs only use publicly available data for arrival estimation, our solution is entirely based on Automatic Identification System (AIS) data. Extracting positional information from AIS, we explore various machine learning approaches to identify clusters. We apply DBSCAN algorithm and show its advantages and disadvantages when used on AIS data. The same process is repeated using meta-heuristics, comparing clustering results generated by a genetic algorithm and by modified ant-colony optimization to those produced by DBSCAN. Finally, we present a hybrid approach and its ability to discover waypoints, highlighting the achieved improvements.

To extend the problem, two constraints are added. The first is the requirement to handle large volumes of streaming AIS data on standard PC-based hardware. The second introduces the common situation of "dark areas" in a map due to problems with receiving and transmitting AIS data. The algorithm discovers route waypoints in efficient and effective ways under these constraints.

CCS Concepts

• Information systems→Data analytics • Information systems→Clustering • Information systems→Data stream mining.

i-KNOW '15, October 21-23, 2015, Graz, Austria
 ISBN 978-1-4503-3721-2/15/10…$15.00
DOI: http://dx.doi.org/10.1145/2809563.2809573

Keywords

Automatic identification system; unsupervised machine learning; meta-heuristics; trajectory analysis

1. INTRODUCTION

The SynchromodalIT project was initiated to enable Dutch logistic service providers (LSPs) increase efficiency and customer service by providing a platform that can support the decision making process. The ability to estimate the future state of a system and suggest a favorable action is the essential component of such a platform [1]. Uncertainties related to arrival times of deep sea vessels have been identified as a major challenge by LSPs. In order to add value to the decision making process, the SynchromodalIT platform has to provide means of estimating these values.

Deep sea vessels have priority over barges at port terminals. Any unscheduled arrival propagates throughout entire supply chain network. Consequently, barges and trucks are delayed and an additional safety stock in warehouses is required, which directly increases the LSPs' operational costs. The late arrivals are influenced by various external factors, weather being the most common one. According to Vernimmen et al. [2] only 52% of the vessels arrive on time, due to both internal and external factors.

In this paper we address the problem of "long term prediction" – the ability to estimate future position of a vessel at least 24 hours in advance. This prediction must be made using only data available to LSPs, which constrains the input to publicly available sources. Automatic identification system (AIS) data belongs to that group. With LSPs primarily using AIS data as the source of information for the vessels they do not own, this study will also base long term prediction on the same input type.

AIS contains static information such as vessel name and it's unique number, dynamic information such as position, speed and heading, and voyage-specific information with destination and estimated arrival time. While static and dynamic information are reliable, the voyage-specific part is often either not used, or contains incorrect information [3], making it unsuitable for estimating arrival times. With positional data as the only reliable source of information, any destination and arrival time prediction must begin with discovery of maritime lanes. These lanes will form edges of a directed graph that contains all routes in a specific region, including the probability of using each edge for sailing from one node point to another. By assigning vessel's position to this graph, we can estimate the likelihood that it will arrive at a specific point.

The process of creating this graph relies on adequate isolation of so called waypoints – i.e., positions on map that, when connected, give the shape of maritime routes. Our research objective in this paper is to find an efficient algorithm that identifies waypoints from large volume of streaming AIS data. Since it is common in real business case applications to have areas that are not fully covered by AIS receivers, or where receiving data is not consistent, our goal is to have an algorithm that can tolerate and handle these inconsistencies.

Research done by Lame et al. [4] indicates that vessels' routes are influenced by weather. When weather conditions deteriorate, ships show tendency of choosing routes closer to shore. Consequently, this implies that waypoints are not fixed coordinates, but can move depending on external factors. Using AIS data only, we cannot predict those factors, but we can identify change in the behavior. Because of this, we require solution that can extract waypoints in real time.

The contribution of this paper is the novel approach in handling large volume AIS data streams, process them in real or near real-time and extract maritime waypoints that define routes and lanes. Although we limit our experiments to AIS and waterways, there is no limitation of applying this concept for other modes of transportation.

The methodology followed in this document is the design science research methodology for information systems research by Peffers et al. [5]

The remainder of the paper is organized as follows. In Section 2 we present our solution design. Starting with problem space, we continue with algorithms for waypoint extraction and then present a hybrid approach. Section 3 is used to evaluate the algorithms and explain their strengths and weaknesses. In Section 4, we present related research on this topic. Section 5 is used to summarize the work done and discuss the future work and impact.

2. SOLUTION DESIGN

2.1 Problem Space

Vessel information, their unique identifier, position, speed, and heading is based on information received from AIS data. To make the simulated environment resemble the scenario of what is available to LSPs as closely as possible, we limit ourselves to publicly available AIS sources. Data used in this experiment is obtained from AIS Hub, through one of the SynchromodalIT project partners. The area of utmost importance for LSPs includes waterways in the vicinity of major Dutch ports. Thus, we constrain the route identification to the region of the North Sea, containing major maritime lanes between the Netherlands and United Kingdom.

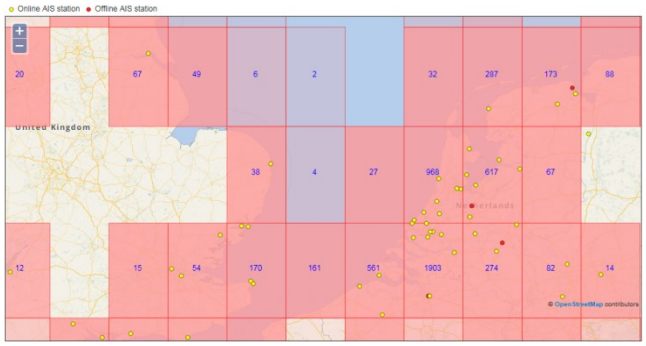

Figure 1: AIS Hub coverage of the North Sea

To analyze different algorithms we stored all AIS messages received from 26-11-2014 13:05:38 to 02-12-2014 05:25:56. During this period we were receiving approximately 693 raw AIS messages per second. These messages originated from all receivers around the world connected to AIS Hub. We stored all raw messages as well as decoded navigation and trip blocks. During simulation we loaded raw AIS messages from our database, setting the limit to the same number of messages per second. Upon loading completion, we performed a decoding and filtering according to the message type. Only messages of type 1, 2, and 3 have been processes, as only these types of AIS messages contain vessel's positional information. Following is another filtration required to isolate AIS messages with latitude and longitude coordinates belonging to the designated problem space within the North Sea.

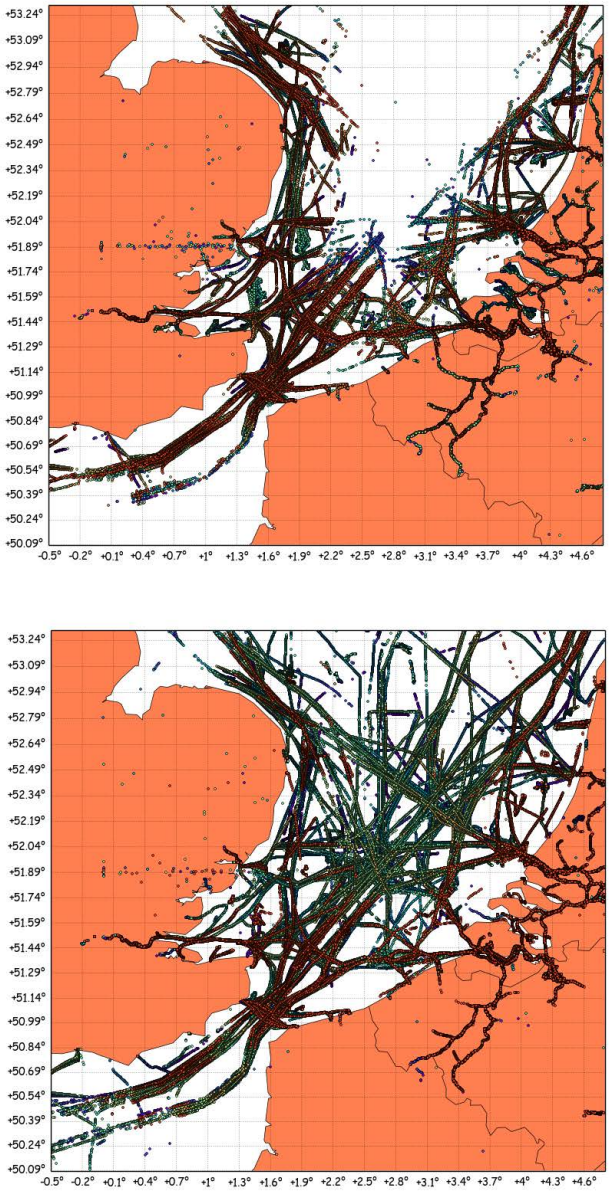

Figure 2: Plotting maritime traffic for interval with limited coverage (above) and complete coverage (below)

We want to emphasize that AIS messages can be picked only from the region covered with online AIS receiver stations. Our

problem region has complete coverage of the coastal and inland areas, but there are zones in the open sea that may not be visible from time to time. This is due to weather conditions limiting how far an AIS message can be received, as well as an AIS station being offline (see Figure 1). Consequently, there are intervals with full AIS coverage for the region, and intervals containing zone from which no information is being received. Figure 2 depicts the difference of data being received with limited and full coverage. The upper image contains a blank area with no information about vessels inside, while the lower image shows the same area with complete information. Since this is the typical scenario LSPs face, we choose to process data in this form without any additional adjustments to the algorithms indicating the level of coverage.

AIS is a self-reporting system, implying that the trustworthiness of information depends on data reported by the vessel, and, as such, is prone to spoofing or intentional incorrect information reporting [6]. For the problem being researched, we do not perform any analysis to assess the quality of the received AIS message, assuming that the frequency of faulty messages is below the threshold that can impact our algorithms in any way.

The executions of our algorithms are being done on standard PC hardware, using Python 3 and NumPy, SciKit-learn and Matplotlib libraries. The typical processing uses query to retrieve messages in 5 – 7 minutes interval, storing them in the local memory buffer and using them as the input for the algorithm developed.

2.2 Waypoint Extraction Algorithms

For the process of waypoint discovery from AIS data, we are looking into algorithms that can perform clustering, be tolerant to noise and capable to quickly process large streaming volumes. We demonstrate the performance of three different algorithms: DBSCAN, genetic algorithm, and modified ant colony optimization.

2.2.1 DBSCAN

DBSCAN is the machine learning algorithm that produces clusters from the area of high density [7]. Pallotta et al. [8] perform unsupervised learning of maritime routes assuming waypoints lay in the area of high traffic density, while the low density area is to be treated as the noise. They use incremental DBSCAN to identify waypoints and use them to plot lanes and routes. Unlike other clustering algorithms DBSCAN does not require the number of clusters to be given a priori. It can handle noise and can produce arbitrarily shape clusters. All these characteristics make it a good candidate for solving this type of problem.

Through scikit-learn package, we use DBSCAN on positional data from our dataset. The number and shape of the clusters produced by DBSCAN depend on two variables specified by the user: ε and minPoints. To determine the most suitable values for these parameters, we run several tests using datasets of different size, with different ε parameter (see Figure 3). Each cluster labeled by DBSCAN is represented with a different color, where points belonging to the same cluster have the same color. Noise points are black and are plotted as circles with smaller radius than cluster points.

The experiment shows that results are influenced by the density of maritime traffic. In the top left picture we observe that with higher ε value and bigger dataset, we get acceptable clusters in the coastal region of United Kingdom. However, due to much higher traffic density, especially in inland areas of the Netherlands, DBCSAN detects one huge cluster, enveloping a large portion of those routes. The top right picture presents the result of running the same algorithm with the same dataset, using a lower ε. This leads to better clustering in the Netherlands with the drawback of labeling majority of points within UK as noise only. Going back to the previous ε value, while reducing dataset (bottom left), we get an acceptable number of clusters in the Netherlands, and yet again the majority of UK points get classified as noise. Reduced dataset and ε (as shown in the bottom right picture), results in labeling most of the traffic as the noise.

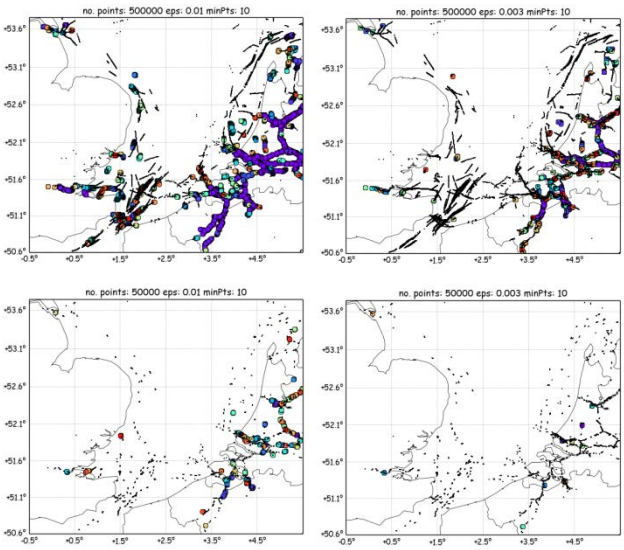

Figure 3: DBSCAN results with varying data size and ε parameter

Hence, we conclude that the DBSCAN algorithm can identify maritime waypoints, but only for the regions without major differences in traffic density. For the observed region of the North Sea, where the number of detected vessels in the vicinity of the Netherlands is considerably higher compared to those in UK, DBSCAN will not produce satisfactory results. The possible improvement is to subdivide the region and use different DBSCAN parameters adjusted to the traffic density.

2.2.2 Genetic algorithm

A genetic algorithm (GA) is a population-based model that uses selection and recombination operators to generate new sample points in a search space [9]. The problem of discovering waypoints can be formulated as optimization problem. If we assume that a waypoint is a circle with a given radius, we can formulate the maximization criteria as to find geometrical positions for a given number of circles, such that they contain the maximum amount of points from the dataset.

Each waypoint can be observed as a gene, containing latitude, longitude and radius values. This version of the genetic algorithm sets same radius value for all chromosomes, making all waypoints of equal area. Genes are grouped in sets forming chromosomes, making each chromosome representation of waypoints set for our problem space. Chromosomes are grouped to form the population of the genetic algorithm.

The fitness criterion used for the evaluation of each chromosome is the number of vessel points contained within all of its genes. The crossover operation is performed by taking two parent chromosomes A and B and choosing random position to subdivide genes into two subsets. A is subdivided into $A_1 + A_2$, while B is subdivided into $B_1 + B_2$. Two new offspring are created, with the first inheriting genes $A_1 + B_2$ and the other $B_1 + A_2$. The initial

population is randomly created using any latitude and longitude value from the given area. During each epoch a fitness evaluation is performed for each chromosome. All fitness values are summed and then two genes are selected using a roulette wheel selection. Following the selection, a crossover operation is performed and it generates two offspring. The process of selecting parent genes and creating offspring is repeated until a new population is created with the number of chromosomes equal to the previous one. This completes one epoch of our genetic algorithm.

Elitism is included to prevent the loss of the best solution. Our implementation limits the elite to always preserve the two fittest genes from each epoch. Mutation is also used as a means to prevent the GA getting stuck in a local maximum. A mutation is assigned with a probability of 5% to occur after each crossover. If the conditions are met and the chromosome has to mutate, a random gene is selected. Then it mutates by picking new, random latitude and longitude values.

In the problem space section, we stated that AIS messages are received in the form of continuous streams. The fact that we are dealing with streaming data, and that waypoints can move due to weather change, makes this problem a good candidate for using GAs to solve.

We run the experiment by loading AIS messages from the database, decoding and filtering those that contain positional data for this region. These positions are stored in the memory buffer, containing vessel data of approximately 7 minutes. In the following text we will refer to this position set as the frame. For each frame we execute one epoch in GA, update fitness scores, and take the best fit chromosome. The results of the experiment are shown in Figure 4. Vessel positions are plotted as black dots, while waypoints are blue circles.

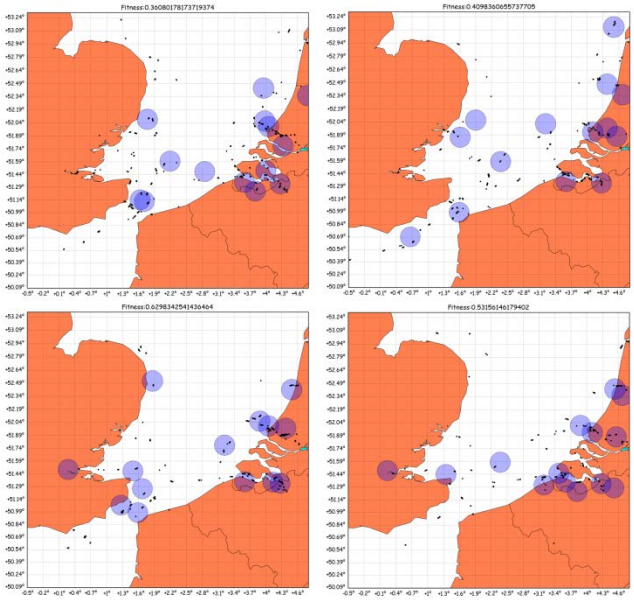

Figure 4: Waypoints identified by genetic algorithm

The top left picture shows the result after the first call to the epoch function. We can see that the majority of the waypoints are located in the highly trafficked areas such as Rotterdam and Antwerp. This is expected behavior since it's the common characteristics of a GA to rapidly converge towards the area of an optimal solution. After running the algorithm for 70 more frames we can see waypoints shifting positions. Also there is an increase of the fitness score from 36% to 40.9% (top right). The GA has achieved the best score after 134 frames (bottom left) with a fitness of 62.9%. Depending how steep the vessel position changes are, fitness can fluctuate, but it shows the tendency to stay above 53.2% (bottom right).

Running the experiment has confirmed that our GA implementation can process streaming data in real time, but the execution time keeps increasing with the increase of fitness values. This is due to the need to evaluate each point within every genome of the population. Adding more waypoints to the chromosome means better precision, but it also negatively impacts the performance. Unlike DBSCAN, the GA approach requires a balanced decision / trade-off between the number of waypoints – precision and the execution time.

Similar to DBSCAN, GA also shows a tendency to converge towards the area of higher traffic density. This can be better illustrated with the result of another experiment, where additional waypoints were added and the waypoint radius decreased. After running GA for 200 frames / epochs, we can see that majority of waypoints now solely lie within the Netherlands (see Figure 5). Without the time constraint, increasing the waypoint number would solve the problem, but in this problem space, a different approach is required. One potential solution is to modify the fitness function in such a way that it penalizes chromosomes with waypoints close to each other.

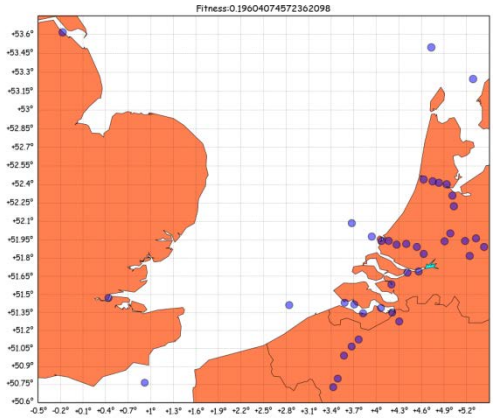

Figure 5: GA converges towards area with higher traffic density

2.2.3 Modified ant colony optimization approach

The third alternative to address waypoint extraction is to view every ship as an ant, leaving their pheromone trail as they go. In such case, major maritime routes can be isolated with higher concentration of pheromone. Waypoints would be intersections of these routes, containing even higher pheromone signature.

In ant colony optimization (ACO) every ant moves randomly and marks a trail with pheromone, while following ants that encounter such trail decide with probability weather to follow it or not [10]. For the problem on hand, we don't have the case of vessels following each other, but we can utilize the concept of pheromone trail to identify sailing density. Adding to the problem that vessels change their sailing patterns due to external factors such as weather, we need a mechanism to isolate the route that was used only in a specific case from the one frequently used.

We will approach the problem in the following manner: every vessel that sails through a point on the map leaves a predefined unit of pheromone. Multiple passing through the same points

sums the pheromone level. After each frame / epoch is complete, a percentage of pheromone will dissipate according to a given constant. If a route is only used once, it will exist for a short period of time and then will disappear, while the frequently used ones will remain. Since vessels following the same path do not have the exact latitude and longitude values, we will subdivide the area into cells, and then for each cell track its pheromone density. We also define parameters that indicate the threshold for high and low density points.

An example of lane discovery process using modified ACO is given in Figure 6. Black points are vessel position in the selected frame. The pheromone concentration is indicated by colors: green, yellow, red and blue to indicate low, medium, high and maximum values respectively. In the figure we show the pheromone levels after 1, 50 and 100 frames. The last picture on the bottom right displays pheromone only and was generated after the completion of the algorithm.

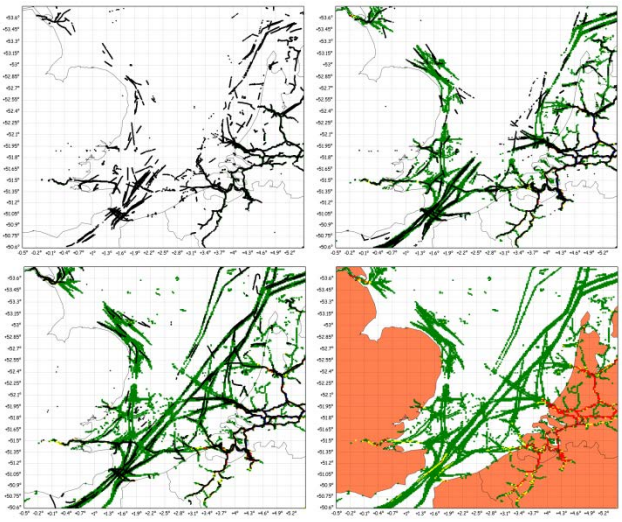

Figure 6: Lane discovery using modified ACO

The blank area in the center of the problem space requires experimentation with different parameters, in order to isolate maritime lanes passing through it, and yet avoid noise pollution. We run two tests (see Figure 7). In the first, we set the following threshold parameters: LOW = 100, HIGH = 1200, MAX = 1800, while in the second we use: LOW = 50, HIGH = 1000, MAX = 1500. Each passing of a vessel through a cell increases pheromone level per frame by 1 for both tests. The dissipation factor for both is based on the percentage of its previous level. In the first test, modified ACO is set to lose 1.5% of its pheromone level. In the second, this factor is 1%.

The results of the first test identify major shipping lanes, but due to the blank area, they are interrupted. Also, the lanes used less frequently are not visible. Reducing the pheromone threshold as well as the dissipation factor, yields additional lanes, but the presence of noise becomes especially visible in the coastal area of UK.

The benefits of this approach are that this is the fastest of all attempted algorithms and with a good dataset, it can clearly isolate major lanes. The negative side is either the misinterpretation of noise that becomes evident when pheromone thresholds are reduced, or not being able to identify some lanes if these values are high.

2.3 Hybrid Approach

Every algorithm covered in Section 2.2 can partially contribute to waypoint discovery, but each one has disadvantages, preventing them to give satisfactory results in isolation. To overcome this, we propose a hybrid approach, combining strengths of all three algorithms.

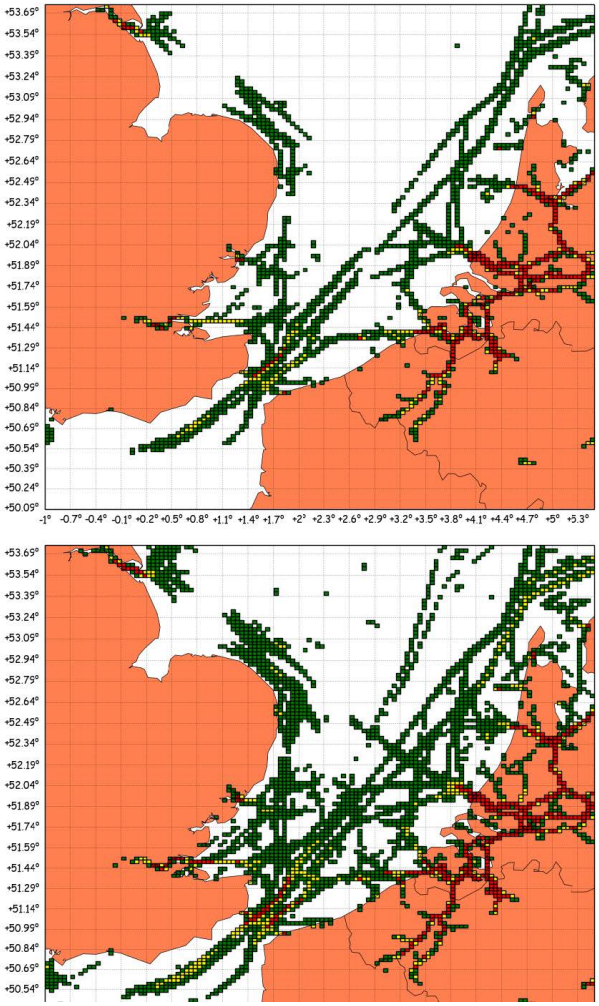

Figure 7: Lane discovery with different pheromone thresholds

GA has shown the capacity to quickly discover waypoints and given more time and processing power, it can produce good discovery results. We keep the ability to quickly converge towards solution, yet remove randomness by replacing GA with a basic Quad Tree (QT) structure. QT is able to decompose space into adaptable cells, with each cell containing points up to the maximum capacity [11]. Using this ability we can instantly subdivide the area into cells, solving speed issues caused by the GA. This also solves the problem for DBSCAN originating from the different traffic densities, as each cell is guaranteed to have a number of points lower than a given threshold. For each cell we run separate DBSCANs and save the discovered clusters.

We improve the output of the algorithm by making additional modifications. The first one is to check the number of points contained within a new cell as soon as it is being created. If it contains only a few points (lower then user defined threshold), this cell is guaranteed not to have any waypoints, so we can

immediately exclude it from further processing and discard the points inside. This additionally increases the algorithm's speed and reduces memory requirements. The second modification is the introduction of maximum depth level, after which we prevent any further cell subdivision. Before implementing this modification, areas with the highest densities, typically lying in the port vicinity, had very high cell depth, which meant many tightly packed points in a small area. Consequently, every point in such an area would be identified by DBSAN as belonging to a single cluster, leading to many waypoints just next to each other. Using a maximum depth this problem is eliminated. Figure 8 illustrates this process for two arbitrary frames. In the upper part of the figure, the subdivision of the region into cells is being shown. Cells with enough points to make them candidates for possible waypoints are colored red, while discarded ones are yellow. The result of DBSCAN is given per cell, where each cluster is marked with a different color. In the lower part, we plot discovered waypoints on top of ship positions for that frame.

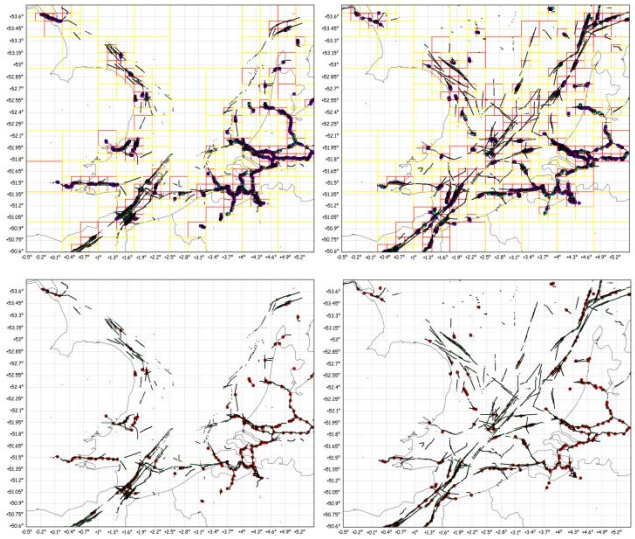

Figure 8: Waypoint discovery using hybrid approach

To meet the requirements of the research question, this algorithm needs to be able to process streaming AIS data in real time. To that end we apply the pheromone trail concept from the modified ACO. Instead of ships, we assign discovered waypoints as trail carriers and for each frame we measure the pheromone density per cell. After processing the entire dataset, or whenever the user requires, the hybrid algorithm returns frequently used waypoints.

3. EVALUATION

Each of the algorithms presented in the previous section has different strengths and weaknesses. To evaluate how adequate each of them is for the extraction of waypoints, we formulate the following criteria:

1) Extraction quality
2) Algorithm efficiency
3) Traffic density handling
4) Noise tolerance
5) Blank region tolerance

The first criterion relates to the ability of the algorithm to produce such waypoints, that they can be used to reconstruct maritime routes in the region. To evaluate the quality of extraction we create the simulation with obvious lanes (edges of a rectangle) and compare the percentage of the rectangle we can recover from discovered waypoints. Algorithm efficiency is about speed and memory requirements. A good performance requires the algorithm to process all input data at the same speed or faster than the speed with which the AIS data streams have been received. The third criterion is to evaluate the algorithm on its ability to successfully extract waypoints in regions with varying traffic densities. Noise tolerance is used to show how tolerant the algorithm is with route deviations, interpreted as noise. Finally, we want to see how good an algorithm is when blank regions are occurring in the problem area.

To test these algorithms we simulated vessel data such that we have four distinctive lanes with four intersections (see Figure 9). Maritime lanes are represented with letters: "a", "b", "c" and "d". Lane "b" contains traffic density five times greater than lane "a". Lane "c" is the opposite and contains only one third of density in comparison to "a". Lane "d" uses same parameters as lane "a", while only sailing speed is set to be twice as fast in comparison to all other lanes. In Figure 9 we can also identify four intersections: A, B, C and D, marked by capital red letters.

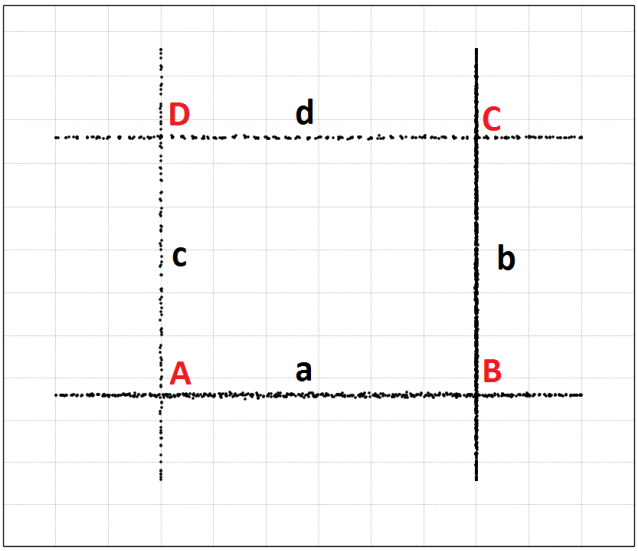

Figure 9: Simulation data

We run all algorithms on this simulated data. Since DBSCAN gives varying results depending on data density and minPoints and ε parameters, we include three different results with parameters best fitted for one of the lanes "a", "b" and "c". The results are given in Figure 10.

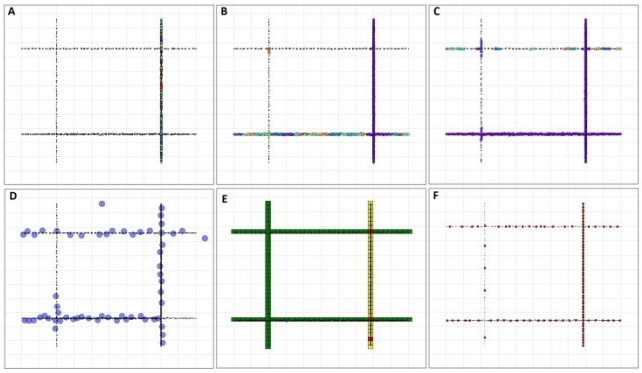

Figure 10: Visualizing results on simulation data

In the top left corner we have DBSCAN result with parameters fitted for lane "b". In the middle, labeled as B, the same DBSCAN algorithm was fitted for lane "a", and in the upper right (C), we increase sensitivity to identify intersection D. In the bottom part, we give results from genetic algorithm (D), modified ACO (E) and hybrid approach (F). To quantify the quality of extracted lanes we compare the percentage of reconstructed lanes against simulated data. We also check the ability of algorithms to discover intersections A, B, C and D. The results are given in Figure 11. From the results we conclude that only genetic algorithm and hybrid approach detect 3 of 4 intersections and that waypoints extracted can be used to reconstruct 69% and 90% of actual lanes, respectively. DBSCAN shows vulnerability to different traffic densities in lanes "b" and "c", therefore no good parameters could be found that would result in effective extraction for this data set. Modified ACO, although visually almost identical to simulated data, cannot identify waypoints in the lane, and for intersections, it only detected points B and C.

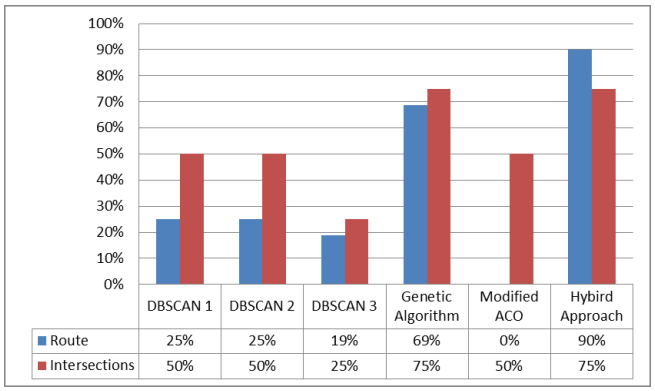

Figure 11: Lane extraction quality per algorithm

To determine algorithm efficiency we measure execution time per one data frame and present them in Table 1. Both DBSCAN and modified ACO perform under 10 seconds. The exact figure cannot be accurately measured since it varies upon initialization time of the simulated environment. The hybrid approach takes 20s – 43s to extract waypoints, depending on number of subdivided regions. Genetic algorithm was the only one that failed to complete after one hour and the execution was terminated. For result evaluation we used waypoints detected by GA at the moment of termination. As can be observed from Figure 10, result D, there are still 2 waypoints that could be assigned to one of the lanes provided the algorithm had longer execution time. Since frame updates do not occur faster than once per five minutes, we conclude that all algorithms with execution time below that value are sufficient for the task. That means that all except genetic algorithm passed this requirement.

Table 1: Execution time

Algorithm	DBSCAN	Genetic Algorithm	Mod. ACO	Hybrid Approach
Execution time (sec.)	10 *	3600 **	10 *	20 - 43

Traffic density is directly related to extraction quality. In our simulation (see Figure 9), lanes "b" and "c" are given in a way to reflect the high density of maritime traffic, as observed in inland and coastal areas of the Netherlands, and that of lower volume, as observed in coastal areas of UK. From the Figure 10, images A, B and C, we can see that varying density prevents DBSCAN to extract waypoints in effective way, either creating huge clusters enveloping entire lane (C), or by failing to identify intersections (A). The remaining algorithms (D, E, F) are able to distinguish between lanes "b" and "c". It is important to note that GA is a special case. Even though it can detect waypoint effectively from lanes with varying density, this variation has impact on execution time, that on previous test (see Table 1) was determined to be unacceptable. Therefore we conclude that only modified ACO and hybrid approach can cope with varying traffic density.

For the noise tolerance, no specific test was made since no problems were observed in real case scenario, with only modified ACO showing higher noise sensitivity with lower pheromone threshold.

Blank region tolerance is the biggest challenge for all algorithms. The existence of such regions leads to degradation of the data quality and thus of the overall results. All algorithms are vulnerable to it, but the presence of "memory" of former waypoint reduces degradation time. DBSCAN is the only one without "memory", causing appearance of blank regions to immediately produce different waypoints, while in the case of the remaining algorithms this change is not instant and occurs over time.

Summing up all criteria we present Evaluation results in Table 2. Satisfactory performance is labeled with '+' and unsatisfactory with '-'. Those cases when algorithm partially complies we label with '+ / -'.

Table 2. Algorithm evaluation

Criteria	**DBSCAN**	**GA**	**Modified ACO**	**Hybrid Approach**
Extraction quality	+ / -	+	-	+
Algorithm efficiency	+	-	+	+
Traffic density	-	+ / -	+	+ / -
Noise tolerance	+	+	+ / -	+
Blank region tol.	-	+ / -	+ / -	+ / -

From the table we see that the biggest challenge for all algorithms is the blank region handling. Noise is quite the opposite and with exception to modified ACO, not deemed to be a problem. The same applies to the extraction of waypoints, however vulnerability to traffic density reduces the overall extraction quality for DBSCAN. Traffic density is also somewhat problematic for GA, forcing it to converge first on high density area. Judging algorithms by their execution speed, only GA underperforms. Although GA is still viable if a restriction on the number of waypoints is imposed, with the increase of data volume, it falls behind. Overall, the hybrid approach has shown better or equal performance for all criteria and we find it to be the best choice for unsupervised discovery of maritime waypoints under these conditions.

4. RELATED WORK

In [12] we conduct a literature review on current state of the art using AIS data for predictions. We classify articles according to prediction objective and horizon. According to that classification, most of the papers are mainly concerned with short term prediction, where the prediction horizon does not go further than

one hour in the future. The reason for that is that these papers focus on detecting anomalous behavior or collision avoidance, hence there is no need for estimations far in the future.

Pallotta et al., Liu et al. and Lei at al. are authors whose research presents solutions that can be used for long term prediction. Pallotta et al. in [8] classifies waypoints as entry, exit, anchor, port and turning points. For turning points, which connect shipping lanes into routes, authors use incremental DBSCAN to extract them from AIS data. That concept is presented as an improvement of [13] by Vespe et al. In [11] turning points are extracted through change detection of ship's rate of turn. DBSCAN approach in [8] attempts to improve the work of [13], where turning point are extracted in close proximity to another. Our initial approach builds on the solution proposed by Pallotta. However, as mentioned in section 2.2.1, we find that DBSCAN alone can discover waypoints in effective way, only if there is no major difference in traffic density. Since our problem space contains that situation, we had to search for other discovery methods.

Liu et al. focus on recovering missing data using tensor CP decomposition [14]. Since this method works on recovering missing data in past as well as into the future, it can be used to predict vessel positions. We choose not to go with this approach due to difference in input data. In [14] all AIS data is stored and accessible, while we focus on streaming data, without the option to store all positions received by AIS Hub stations.

Article [15] by Lei et al. also use DBSCAN to identify "hot regions" and use TMP algorithm based on PST probabilistic suffix tree. They focus on vessel's moving behavior instead of maritime patterns.

5. CONCLUSIONS

In this paper we addressed the problem of discovering waypoints of maritime routes, based on data obtained from streaming AIS messages. We presented the problem space, constraints and desired outputs. Then we described three different algorithms used for waypoint extraction as well as their strengths and weaknesses when applied to the particular problem. Following, we presented a hybrid approach, incorporating best features of all previous algorithms. In the separate section we showed the criteria used to evaluate the adequacy of each approach for waypoint extraction. We concluded that the hybrid approach is the best choice to use, especially, when the problem space contains streaming data that cannot be stored completely, has to be processed in real-time, and has potential to include blank areas. Finally, we mentioned the related work in this domain, pointing out similarities and differences of proposed solutions in those papers to the one presented here.

Although the hybrid approach can successfully extract waypoints from AIS data, the complete prediction requires the generation of a directed graph, representing maritime routes for the given area. In future work we plan to extend this work with a novel approach in extracting edges (i.e. sea lanes of a graph) from the same input data. Also, all algorithms are to be tested on a much larger dataset, preferably spanning several months.

Completing the process of extracting maritime lanes in the form of a graph will allow long term predictions. All major characteristics, such as lane length, average speed of a lane and the frequency of usage can be incorporated as attributes of the graph's edges. Predicting the future state of a vessel will require only mapping the ship's current position on that graph. It is our expectation that this approach will contribute towards uncertainty reduction related to unscheduled arrival times of deep sea vessels, and aid LSPs increase the efficiency of their operations.

6. REFERENCES

[1] Dutch Institute for Advanced Logistics. (2013, August 11). *SynchromodalIT*. Available: http://www.dinalog.nl/en/projects/r_d_projects/synchromodalit/

[2] B. Vernimmen, W. Dullaert, and S. Engelen, "Schedule unreliability in liner shipping: origins and consequences for the hinterland supply chain," *Maritime Economics & Logistics,* vol. 9, pp. 193-213, 2007.

[3] A. Harati-Mokhtari, A. Wall, P. Brooks, and J. Wang, "Automatic Identification System (AIS): data reliability and human error implications," *Journal of navigation,* vol. 60, pp. 373-389, 2007.

[4] O. D. Lampe, J. Kehrer, and H. Hauser, "Visual Analysis of Multivariate Movement Data using Interactive Difference Views," in *VMV*, 2010, pp. 315-322.

[5] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems,* vol. 24, pp. 45-77, 2007.

[6] F. Katsilieris, P. Braca, and S. Coraluppi, "Detection of malicious AIS position spoofing by exploiting radar information," in *Information Fusion (FUSION), 2013 16th International Conference on*, 2013, pp. 1196-1203.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, 1996, pp. 226-231.

[8] 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, pp. 2218-2245, 2013.

[9] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing,* vol. 4, pp. 65-85, 1994.

[10] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 26, pp. 29-41, 1996.

[11] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica,* vol. 4, pp. 1-9, 1974.

[12] A. Dobrkovic, M.-E. Iacob, J. van Hillegersberg, M. Mes, and M. Glandrup, "Towards an approach for long term AIS-based prediction of vessel arrival times," in *Lecture Notes in Logistics*, ed: Springer, (forthcoming 2015).

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

[14] C. Liu and X. Chen, "Vessel Track Recovery With Incomplete AIS Data Using Tensor CANDECOM/PARAFAC Decomposition," *Journal of Navigation,* vol. 67, pp. 83-99, 2014.

[15] P.-R. Lei, J. Su, W.-C. Peng, W.-Y. Han, and C.-P. Chang, "A framework of moving behavior modeling in the maritime surveillance," *Journal of Chung Cheng Institute of Technology,* vol. 40, pp. 33-42, 2011.