# Application of AI and In-Memory Computing for Extracting Vessel Movement Patterns from Historical Data

**Dominik Filipiak, Milena Stróżyna, Krzysztof Węcel, Witold Abramowicz**
Poznań University of Economics and Business
POLAND

krzysztof.wecel@ue.poznan.pl

**Matthias Steidel**
OFFIS
GERMANY

Matthias.Steidel@offis.de

## *ABSTRACT*

*We present an approach for extracting vessel movement patterns from historical Automatic Identification System (AIS) messages that are combined with historical weather information. In the maritime domain, safe and efficient vessel operations depend upon a prescient voyage planning and ship monitoring. Realization of these tasks require IT systems that would suggest a safe route based on past experience and best practices in certain sea areas. Considering the volume and velocity of data the proposed solution consists of a combination of a genetic algorithm and a method for generating a graph representing the maritime traffic network. Our solution applies big data technologies to assure efficiency and scalability and to take advantage of fast in-memory computation. The paper presents the current state of the work and an outlook on potential future applications, including application in the military security issues and providing support in decision making.*

## 1 INTRODUCTION

A standard procedure in today's shipping domain is a predictive voyage planning and monitoring. Voyages are planned berth to berth, resulting in a route that consists of waypoints a vessel should follow to assure security and safety. In voyage planning a number of aspects must be considered, such as minimum water depth, existing routes, and meteorological information like wind or tide. This also concerns military operations, whether planned or in emergency. Nowadays such a planning is performed manually by a ship captain with utilization of a dedicated software. An alternative is an automatic route planning by specialized assistance systems that support the navigator both in planning before the voyage starts and in monitoring when the ship is underway. Such systems analyze historical movements of ships and apply various methods to extract from this data maritime traffic patterns. Having the knowledge about typical behaviour patterns under different conditions, they can propose a safe route when planning a voyage. At the same time, other useful parameters, such as weather conditions, can be automatically taken into account.

The routing problem can be considered a classical OR problem. There are, however, several features that make our research problem different. First of all, the task is not about pure routing but about discovery of the network that might be used for route planning. Second, we need to consider huge volume and velocity of data. Here we speak of about 1 GB (gigabyte) of data daily. It is particularly challenging as classical operational research (OR) algorithms do not scale. Also, we plan also to realize more complex scenarios which means that calculations need to be repeated many times with additional constraints, hence the importance of efficiency.

The contribution of the paper is the method for extracting vessel movement patterns from historical maritime data combined with historical weather information that might be then used in an assistance system. We introduce the concept of Recommended Corridors (RC) for modelling the most commonly used traffic patterns between a given origin and a destination. It is intended to represent past experience and best practices. By its definition, a RC represents the area in which vessels most commonly travel. To enable this, RCs must consider the information typically used during voyage planning and monitoring, as defined in regulation of the International Maritime Organization [1]. The following is considered: length, draught, and general manoeuvrability, as well as meteorological conditions, and context information, e.g. minimum draught along the potential route. In order to meet this requirement, there might be different RCs for the same route depending on the considered ship type and meteorological conditions. As a result, the RCs are ship-specific. The added value of the proposed approach lies in speed of computation by applying relevant technologies (here Apache Spark) and optimizing the performance of algorithms. With the speed of calculation it is possible to test many different scenarios and tune hyperparameters of the method to suit the needs. Thus current approach can consider seasonality of movements (daily, weekly, yearly). We have generated many different meshes for different vessel types, different sizes of ships, different draught values, and also for different weather conditions, measured on Beaufort scale and considering the ice coverage where necessary.

A RC is determined based on historical movements of vessels, taken from past AIS messages. The Automatic Identification System (AIS) is a self-reporting system, which is mandatory for vessel with a gross tonnage above 300. Information transmitted via AIS includes position, speed, and course information as well as ship-specific information such as length, type or draught [2]. In the presented approach, the historical AIS data are used to create a graph-based traffic pattern representation, which will be considered as a mesh from here. RCs are then derived from this mesh. In general, a mesh is a graph that represents all RC in a given sea area and is context-sensitive and ship type specific. The proposed solution consists of a combination of a genetic algorithm and a method for generating a graph representing the maritime traffic network.

Due to the obligation to be equipped with AIS for vessels above a certain gross tonnage, a huge amount of vessel tracking information is available, which can be analysed posterior. By applying techniques and methods from the research field of big data, the process of extracting maritime traffic patterns might be streamlined [3]. Therefore, our solution applies big data technologies to assure efficiency and scalability and take advantage of fast in-memory computation. The aim of the paper is to present the current state of the work on our approach and an outlook on potential future applications.

The rest of the paper is organized as follows. Section 2 studies the existing research in the context of maritime pattern extraction and modelling is presented. The paper is continued by describing the approach, the architecture applied for data analysis and algorithms used for extracting traffic patterns required for providing RCs. Then, the results of initial experiments on the method are presented. We conclude this paper with the description of potential applications that may benefit from this approach.

## 2 MARITIME TRAFFIC PATTERNS

There exists a variety of approaches for extracting and representing maritime traffic patterns from historical AIS data. Following the classification of [4], the existing research on extracting and modelling traffic patterns can be divided into grid-based, vector-based and graph-based approaches.

Grid-based approaches are applying a grid in the considered sea area. To represent traffic patterns, the typical vessel behaviour is modelled in each cell by considering parameters such as speed, course or position. Such a procedure can be found in [5]–[7] . Reference [5] models the typical behaviour of the ships with information about position and speed within a cell. The authors show how their approach can be used for predicting the cell in which the vessel will be in 15 minutes [5]. Reference [6] proposes a knowledge-based prediction approach. The traffic patterns needed for this approach are extracted by modifying the DBSCAN algorithm in such a way that the algorithm can be applied to a grid. Subsequently, these grid-based patterns are used to calculate the typical moving patterns of vessels by using Kernel Density Estimation (KDE) [6]. A similar approach was presented in [7]. A grid was applied to the considered sea area and in an unsupervised learning phase, the maritime traffic patterns are described by extracting speed patterns inside each cell. For this purpose, KDE is applied.

Vector-based approaches assemble tracks or trajectories from associated AIS data points. From these tracks, significant points are extracted, such as points where ships manoeuvre, stop or enter the considered sea area [4]. The approach proposed by Pallotta et.al. [8] is based on [9], which is a method for identifying distinctive events in historical AIS trajectories. The main idea of [8] is to model the variation in the behaviour of vessels travelling on the same route. For this purpose, the authors approximate the movement of the vessels by modelling it as an Ornstein-Uhlenbeck process. Another vector approach can be found in [10]. With their work, the authors propose an extension to the Piecewise Line Segmentation method. This extension enables the compression of historical AIS trajectories without losing relevant information about the trajectory such as stop points or distinctive turning points. Furthermore, the authors show how this representation method can be used for determining the similarity between different trajectories [10].

Graph-based approaches model traffic patterns in a more abstract way than the two previously mentioned methods. All existing approaches have in common that they extract significant points from historical AIS data and represent this in a graph [4]. Such points can be, for example, points at which ships manoeuvre, reduce speed or anchor. Typical traffic patterns can then be described by a sequence of nodes in the graph.

Reference [11] introduces the Route Topology Model (RTM) for the North Sea region. The nodes in this approach are geographic points at which vessels usually perform a manoeuvre. The RTM is a topological representation of traffic, which means that only the nodes have a geographical reference. The edges only connect the nodes and thus model the reachability of the nodes by the ships. Hence, this approach does not allow modelling e.g. the exact course of a waterway or a channel.

In contrast to [11], Varlamis et al. [12] propose a network representation, which is focused on the different navigational states vessels have during their journey. To create the graph, the historical trajectories are analysed taking into account predetermined thresholds for course or speed changes. Points at which the respective threshold values are exceeded are then clustered using the DBSCAN algorithm. A Markov Model is used to model typical behaviour. For this purpose, the transition probabilities from one node to another are stored [12].

Arguedas, Pallotta and Vespe [13] propose a directed-graph representation for typical vessel traffic patterns. This graph is created by identifying waypoints in historical vessel trajectories. Waypoints are defined in this work as geographic points at which significant course changes can be detected. To extract such points from historical vessel trajectories, the authors apply the Douglas-Peucker algorithm, which is a method for smoothing curves [13], [14]. Those points are subsequently clustered with the DBSCAN algorithm. Afterwards, the order and the frequency in which vessels visit the determined waypoints is calculated, which yields the directed-graph.

The idea of waypoints extraction in order to identify traffic patterns is supported also by Dobrkovic et.al. [18], [19]. In the research they show that application of an evolutionary algorithms, such as genetic algorithm (GA) or ant-colony optimization to discover sequential waypoints can be a viable alternative to other machine learning approaches. The proposed GA is able to provide accurate results once good criteria for GA fitness function have been found. Moreover, they addresses the problem of varying density traffic and high computational time of GA by using quad tree structures to pre-process the data and isolate areas of high traffic density. Moreover, this approach handles routes with missing data.

In contrast to the works presented above, Lamm and Hahn [15] propose to use a ship type specific graph for representing typical vessel movement patterns. The idea is to identify the points in historical vessel tracks at which vessels performed a manoeuvre. Hence, the resulting graph is called a manoeuvre net. A manoeuvre is defined in [15] according to the definition of Fossen [16], who describes a manoeuvre as a change in course or speed. To identify a manoeuvre, Lamm and Hahn [15] propose to use the cumulated sum (CUSUM) procedure. CUSUM is a method for change point detection in a stochastic processes, which usage for manoeuvre detection was demonstrated in [17]. In order to distinguish between soft manoeuvres that are performed to maintain the course and strong manoeuvres that are necessary in order to follow the vessel's route, a threshold for course and speed change is defined. The simple moving average (SMA) is then used to determine if the defined thresholds are exceeded. This procedure yields a set of manoeuvre points for each ship type, which are clustered by applying DBSCAN. To create the manoeuvre net the centre of each cluster is extracted by applying the medoid calculation. At the end, the consecutive manoeuvre points can be connected with edges, which yields the manoeuvre net [15].

As described above, there exist a variety of approaches for extracting and representing maritime traffic patterns. All the approaches considered share a common basic procedure: at the beginning, historical AIS data is processed. Either the data is processed and analysed in a grid or the historical AIS data is merged to tracks and further processed accordingly. Another option is to extract distinctive points from historical tracks and model them in a graph. It becomes evident that all approaches only use historical AIS data as an information source. Context related information such as weather conditions are not considered in the analysis and representation. Furthermore, some of the chosen approaches are inefficient for large amounts of data, such as grid-based approaches or those using the DBSCAN algorithm. In addition, most approaches are evaluated with a relatively small amount of data covering only a small area.

In this work, we present an approach for tackling these shortcomings. We introduce the concept of RCs to represent context-sensitive maritime traffic patterns. For this purpose, existing concepts are extended in such a way that context-sensitive patterns can be extracted and modelled. Furthermore, we describe a more efficient approach for extracting those patterns by using the Lambda-Architecture and in-memory computing technologies.

# 3 GRAPH-BASED EXTRACTION OF RECOMMENDED CORRIDORS

In order to extract context-sensitive traffic patterns, we extend and combine existing approaches in such a way that historical AIS data is augmented with historical weather information. This data is first used to generate a ship type specific manoeuvre points by applying the CUSUM algorithm. Subsequently, these manoeuvre points are used as an input for a genetic algorithm in order to identify waypoints and then generate the mesh and recommended corridors.

In this section, we describe the algorithms used in a greater detail. Moreover, propose an architecture that enables fast and efficient data analysis and allows to update the extracted traffic patterns as soon as new data becomes available. Moreover, we describe the data used for initial experiments.

## 3.1 Scope of Data

The extraction of context-sensitive traffic patterns requires both historical vessel movement data and historical weather data. In addition, the sea weather data must cover the same period and area as the historical vessel movement data. Here, historical AIS data from 2017 to 2018 is available. Coverage of the data is concentrated on the German Bight and the Baltic Sea. In addition, AIS data from other areas are also available, so it covers almost the entire world. This yields a total amount of available AIS data of approximately 4 TB.

The appropriate historical sea weather data can be obtained from services offered by COPERNICUS[1]. COPERNICUS is a monitoring service for the marine environment sponsored by the European Commission. The required data can be obtained by using a Rest-API and is then fused into a single dataset in order to be combined with the historical AIS data.

## 3.2 System Architecture

As described above, the RC are intended to support mariners both in voyage planning and monitoring. Hence, the underlying system for the calculation and provision of RC must be able to regularly process large amounts of data and extract patterns from them. In the event that the route must be replanned, an alternative route has to be provided within a reasonable period of time. In addition to this, one has to consider that the provided traffic patterns may change in time and that RCs are not static and should be updated in that case. In order to ensure the sustainability of the RC concept, the database has to be extended by new data regularly. Furthermore, the data must be analysed regularly in order to extract the most recent patterns. To meet these requirements, this paper suggests the use of an architecture that is in general related as the Lambda architecture.

Fig. 1 illustrates the proposed architecture that consists of three layers. The basic AIS and weather data is stored inside the *Batch Layer*, which will be called master data set. As soon as new data, i.e. historical AIS and weather data, is available, it will be appended to the master data set. This layer also contains all expensive operations such as pattern extraction, mesh calculation and RC generation. It considers also some other static data sources, such as ports coordinates, or geographic information (continental contours, for example).

The Lambda Architecture of the system is intended to separate batch data processing and calculations operations (such as pattern extraction) from requests, which tend to be resource consuming.

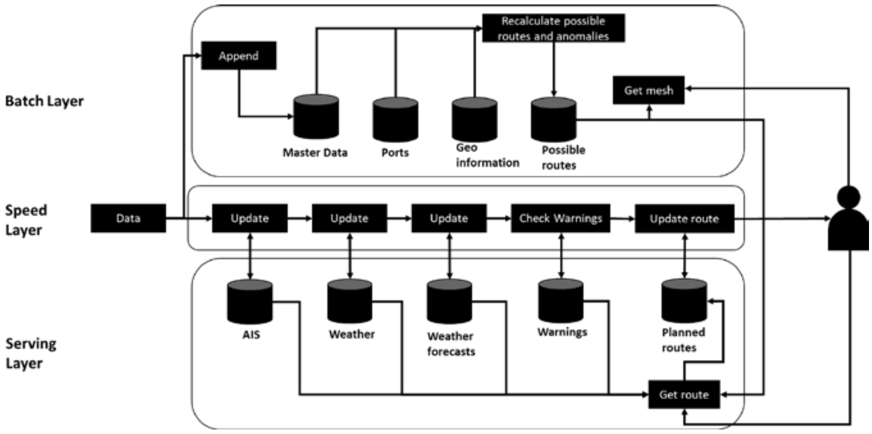


**Fig. 1. The Lambda Architecture of the proposed system.**

As in a standard Lambda architecture implementation, the end user, who e.g. wants to request an RC, accesses the Speed layer. The results and data provided by the Batch and Speed Layer are merged and are made available to the user.

[1] https://marine.copernicus.eu/

## 3.3 Proposed method for traffic patterns and RC extraction

In order to provide valid RCs, knowledge about all possible routes in a sea area must be available and represented adequately. For this purpose, we propose to use a mesh. Such a mesh consists of nodes and edges, whereas the nodes represent geographical points which are significant for vessel movement. The edges represent the connection between nodes – a vessel can travel from node *a* to node *b*, if they are connected.

For generating this graph-based vessel traffic representation, at first we combine the CUSUM algorithm and a genetic algorithm (GA). As presented in Section 2, the CUSUM algorithm detects the manoeuvre points based on speed and course changes of vessels. Applying this algorithm to a set of historical vessel tracks yields a set of manoeuvre points (which was approximately 10 times smaller than the original dataset in our experiment presented in Section 4). These manoeuvre points are then used as an input the genetic algorithm. Since this algorithm is rather expensive to run, the data are partitioned. Dobrkovic et al. [18] argued that the use of spatial partitioning will improve the search for proper waypoints. Instead of the suggested quadtrees, we used *k*-d trees, a well-known spatial partitioning algorithm in a *k*-dimensional space [20]. Each partition ideally should contain the same number of manoeuvre points – in our experiments *k*-d trees outperformed quadtrees in this task.

For calculating the graph-based traffic representation, we roughly follow the approach presented by Dobrkovic et al. [18] – the genetic algorithm is applied in each partition to iteratively find good waypoint candidates in them. The manoeuvre points generated by the CUSUM algorithm are taken as the *population* for the genetic algorithm. At the beginning, a number of random points are drawn from the existing manoeuvre points in each partition. These coordinates, along with some constant radius (0.07 nautical miles, for example), which in the GA terminology will be called *genes*, are gathered in *chromosomes*. A gene contains a triple $(x,\ y\ ,r)$, which represents a waypoint using a circle $x^2 + y^2 \leq r^2$. Intuitively, a good waypoint candidate will gather many other points inside. To evaluate fitness of a chromosome, we check how many points are within circles from each gene. Bad points (containing no other points) are later replaced by better genes. We also penalize overlapping points. Additionally, we introduced a small mutation factor, which randomly replaces some genes in order to endure population diversity. For each epoch, we use roulette wheel selection and one-point crossover. After reaching a pre-defined number of operations, our waypoint finding GA stops. The result of GA is a set of waypoints.

Waypoints are equivalent to nodes of the generated mesh. What is necessary, is to discover the edges, i.e. which waypoints should in fact be connected. This operation is conducted based on historical AIS data. By looking at every single trajectory of all vessels (that pass an area of interest), we can track which waypoints they 'visited'. It is therefore necessary for each AIS point to assign it a nearest waypoint.

We refer to the process of adding information about the nearest waypoint to each AIS message as "AIS enrichment". We add both the identifier of a waypoint and the distance to it. For finding the nearest waypoints we have used kNN algorithm available in Sklearn[2]. It was necessary to adjust the metrics as well as the partitioning function. We started with Minkowski distance metrics and k-d tree partitioning but ended up with haversine distance and ball tree partitioning. Taking into account the number of AIS data to be enriched, the process of assigning waypoints to AIS data proved to be very time consuming. Therefore, we introduced many optimization techniques to make the task feasible. It was also necessary to optimize code in PySpark i.e. use Pandas UDF[3] instead of plain iteration over rows in a CSV dataset.

[2] https://scikit-learn.org/stable/modules/neighbors.html

[3] https://databricks.com/blog/2017/10/30/introducing-vectorized-udfs-for-pyspark.html

Having all AIS points annotated with the nearest waypoints, it may seem straightforward to reconstruct the connections between waypoints. Generation of edges proved to be a less challenging task at least from the performance point of view. General approach to reconstruction of edges is presented in Fig. 2.

```
function DISCOVEREDGES(AIS)
    AIS_f ← FILTERAIS(AIS)
    AIS_w ← AIS_f.PARTITIONBY('mmsi').ORDERBY('timestamp_ais')
    AIS_w ← AIS_w.WITHCOLUMN('to_waypoint')                ▷ mark current waypoint
    AIS_w ← AIS_w.WITHCOLUMN('from_waypoint')              ▷ mark previous waypoint
    AIS_w ← AIS_w.WITHCOLUMN('changed')                    ▷ identify rows with changed waypoints
    AIS_c ← FILTERTRAJECTORY(AIS_w)
    edges ← AIS_c.GROUPBY('from_waypoint', 'to_waypoint')
    edges ← FILTEREDGES(edges)
    edges_d ← edges.WITHCOLUMN('distance_km')              ▷ calculate distance between waypoints
    return edges_d
```

**Fig. 2. Algorithm for edges discovery.**

Unfortunately, due to incomplete distribution and often low quality of AIS data, several measures need to be undertaken to achieve good results. A visual introspection of maps that show a generated mesh, proved that the method generated ‘impossible’ or ‘inappropriate’ connections between some waypoints which further on had to be eliminated.

The algorithm lists several filtering functions that are applied either to obtain meshes for different conditions or just to improve the quality:

- FILTERAIS – the function selects a subset of data for a given vessel type (e.g. tanker) or weather conditions (e.g. heavy wind). It is also used to build a mesh on a subset of points like only on important manoeuvre points as identified by CUSUM.
- FILTERTRAJECTORY – the function is applied to trajectories of a ship. It is responsible for selection of points out of which edges will be constructed. For example, it can only leave AIS points that reflect the changes in the waypoint. In another variant it is used to consider only points that are sent within specific time period.
- FILTEREDGES – the function is applied to edges. For example, we can filter out edges that are too long (e.g. distance > 250 km) or are very rare (e.g. followed by only a single ship). The method is mostly applied to visualizations.

# 4 IMPLEMENTATION AND RESULTS OF INITIAL EXPERIMENTS

The idea of RC extraction and mesh generation presented in Section 3 was implemented and initially tested. In this section we describe the initial results.

Having the system’s architecture defined, the algorithms presented in Section 3 (CUSUM, k-d tree partitioning, genetic algorithm and mesh calculation) were implemented and initial experiments were conducted.

From the perspective of evolutionary computing approach, performing GA calculations on separate partitions of roughly equal size help to prevent densely populated regions from dominating the set of resulting waypoints. Such a partitioning has one more advantage – data can be processed separately, which can offer a significant speed-up stemming from parallel computations. To leverage this property, we implemented our solution in

Apache Spark, a popular data processing framework written in Scala. Our experiments proved that this step greatly significantly benefits from the distributed nature of disjoint spatial partitioning and Apache Spark.

Fig. 3 shows the result of the three used algorithms (CUSUM, k-d tree partitioning, and genetic algorithm) with week-long sample of AIS data in the area of the German Bight. The sample dataset was narrowed only to tankers, passenger, and cargo vessels. The genetic algorithm was run with chromosomes of length 2 and In our experiment, 300 epochs (which seems to be were sufficient.). The leftmost picture presents the area of the German Bight, while the other one shows a zoom on the area of the northern Netherlands. The spatial partitioning represents density clusters as expected – the areas near busy ports are represented by small rectangles, whereas more sparsely populated areas are contained in larger figures. The results of the already conducted experiments are promising and show the convergence of waypoints to the busiest area of manoeuvres.

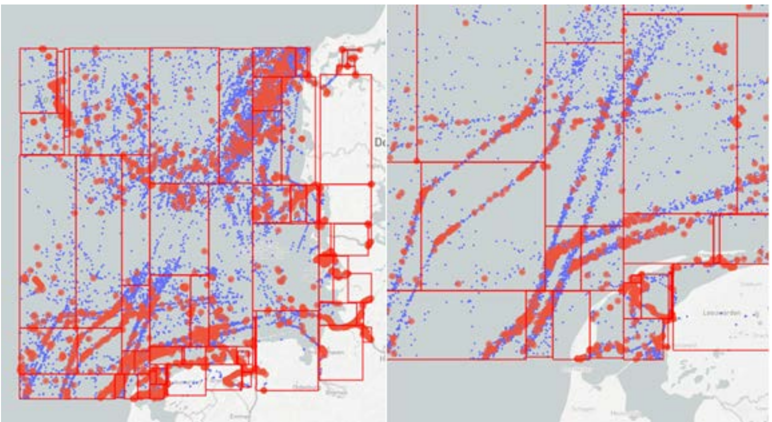

**Fig. 3. Results of application of CUSUM (manoeuvre points, marked in blue), k-d trees (partitions, red rectangles) and genetic algorithm (waypoints, marked as red points).**

In the mesh generation, the most time-consuming operation was AIS enrichment. As can be seen in Table 1 good implementation decisions can result in performance 100,000x faster than naïve solutions.

**Table 1. Comparison of methods for AIS enrichment.**

| Method | Throughput (rows/s) |
|---|---|
| kNN, iteration over RDD with flatMap | 7.3 |
| kNN, with UDF | 399 |
| Brute-force search, with UDF | 16,508 |
| kNN Minkowski, with Pandas UDF | 1,517,969 |
| kNN haversine, with Pandas UDF | 824 |

The quality of meshes was evaluated using visual introspection by comparing the routes available from navigation software visualization and routes generated by our solution. Meshes generated without any restrictions were not overlapping much with reality. Therefore, we have introduced several refinement steps.

The most reliable results were obtained by implementation of FILTERTRAJECTORY function (see Section 3.3). We have observed that many problematic edges stemmed from the long delays between AIS messages. At this stage, the aim was to choose only those passes of vessels between waypoints, where the time between messages is restricted, i.e. edges will be time-bound. As the focus was on more realistic time deltas between waypoints, we assumed that a time period between two consecutive messages connecting two waypoint areas should not be longer than 15 minutes.

By applying the above approach, we further reduced the number of rows used for generation of edges. We obtained 1,408,451 rows compared to 1,494,227 rows of a baseline method (borderpoints, no filtering). The reduction does not seem to be significant – just 85,776 rows. Nevertheless, the number of resulting edges was reduced significantly from 170,644 (borderpoints) to 156,103 (time-bound). Reconstruction indeed avoided too long hops. The visualization of the mesh on the map, generated according to the time-bound approach, is presented in Fig. 4.

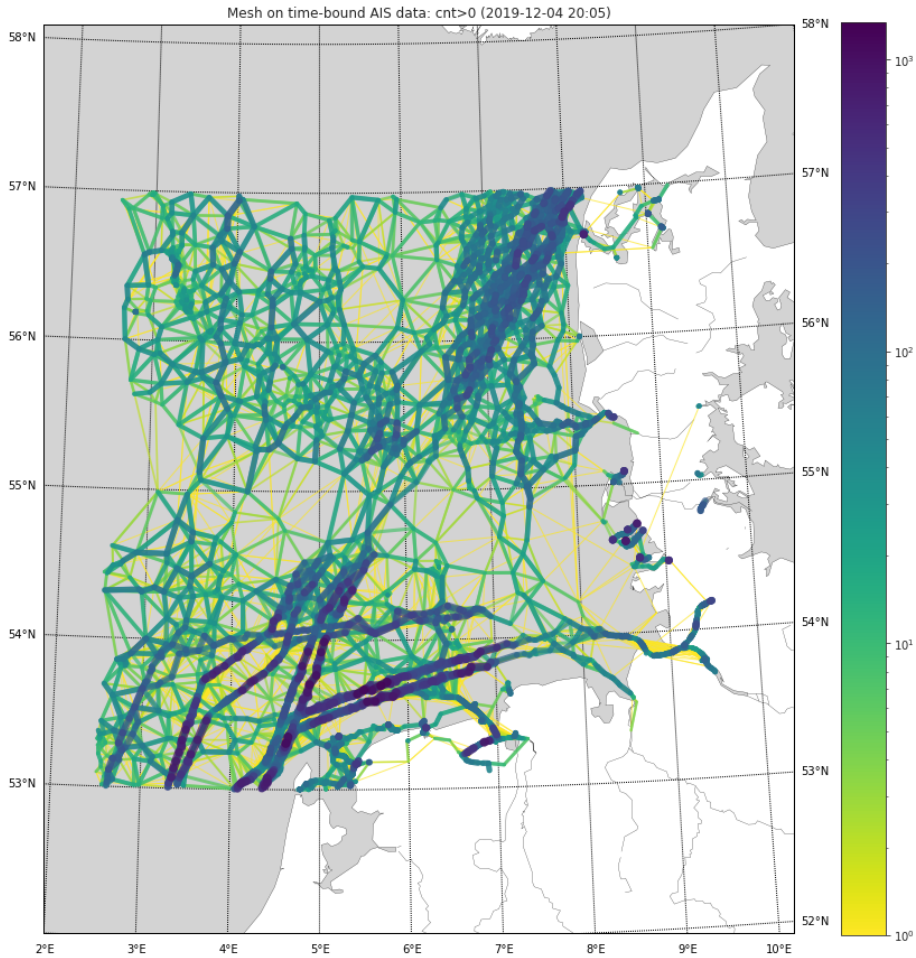


**Fig. 4. Mesh generated for German Bight using time-bound approach**

Overlapping with real data, we can identify traffic separation corridors at the south part of the German Bight. It is not observed in the north part where such regulations are not applied. The less dense mesh in the middle of the figure results from the less dense AIS data.

Having the mesh generated, the final step is getting the RC. By providing a starting location and a destination of a vessel, the RC might be generated based on the mesh.

# 5 SUMMARY AND FUTURE APPLICATIONS

Predictive voyage planning is a crucial task that enables safe and efficient vessel operation. To support maritime actors in this task, we introduced the concept of RC, which is the method for representing the context-sensitive vessel movement patterns. The traffic patterns are extracted based on analysis of historical AIS data. We propose to combine the CUSUM algorithm for manoeuvre point detection with a genetic algorithm to extract waypoints. These waypoints are then connected into a mesh that represents all recommended routes between a starting location and a destination. Based on the mesh, the recommended corridor for a given vessel and a given route is created. We address the problem that pattern extraction is generally a time-consuming operation by using the Lambda architecture for the system and applying the k-d partitioning for performing the calculations. Thanks to that, the data can be processed separately on different partitions, resulting in a significant speedup of computations.

The presented approach has the potential to be used on board of ships as well as onshore to support maritime operators in sea surveillance (including military operations), resulting in increase of the safety at sea in general. On board it can used in two ways: RCs can be proposed to mariners during a voyage planning, especially in case of unknown or dense sea areas. In that case RCs provide mariners with insights to best practices and past experience of other mariners that already travelled in a given area. Furthermore, RCs can assist during voyage monitoring and replanning of the route in case of changing weather conditions. RCs can also be used to detect alert about different anomalies in vessel behaviour (e.g. by checking if it follows the RC) in order to prevent maritime accidents or the intrusion in Marine Protected Areas that could have negative impact on the environment. The last scenario would be most relevant for military operations. The recommended corridor can also be visualised in a form of a heatmap, showing sea areas mostly visited by vessels of interest. Such heatmap could then support planning of relocation of forces necessary to assure safety on the sea.

Future work includes further optimization and validation of the proposed algorithms. It includes further experiment on generation of RCs and the mesh for a longer period of time and application of context-specific information (e.g. weather conditions and vessels characteristics). Furthermore, methods for anomaly detection will be developed in order to show an application of the RC-concept. Finally, the RC-concept itself will be evaluated to examine its accuracy and usefulness in a real (production) environment.

# ACKNOWLEDGEMENT

This work has been conducted in the context of the HANSA project, which is funded by the MarTERA partners German Federal Ministry of Economic Affairs and Energy (BMWi), Polish National Centre for Research and Development (NCBR) and Research Council of Norway (RCN) and is co-funded by European Union's Horizon 2020 research and innovation program under the framework of ERA-NET co-fund.

# REFERENCES

[1] International Maritime Organization, „Guidelines for Voyage Planning“. 25-Nov-1999.

[2] International Maritime Organization, „International Convention for the Safety of Life At Sea, 1974“. 01-Nov-1974.

[3] L. Cazzanti und G. Pallotta, „Mining maritime vessel traffic: Promises, challenges, techniques“, in OCEANS 2015 - Genova, Genova, Italy, 2015, S. 1–6.

[4] M. Riveiro, G. Pallotta, und M. Vespe, „Maritime anomaly detection: A review“, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Bd. 8, Nr. 5, S. e1266, Sep. 2018.

[5] N. Bomberger, B. Rhodes, M. Seibert, und A. Waxman, „Associative Learning of Vessel Motion Patterns for Maritime Situation Awareness“, in 2006 9th International Conference on Information Fusion, Florence, 2006, S. 1–8.

[6] Z. Xiao, L. Ponnambalam, X. Fu, und W. Zhang, „Maritime Traffic Probabilistic Forecasting Based on Vessels’ Waterway Patterns and Motion Behaviors“, IEEE Transactions on Intelligent Transportation Systems, Bd. 18, Nr. 11, S. 3122–3134, Nov. 2017.

[7] B. Ristic, „Detecting Anomalies from a Multitarget Tracking Output“, IEEE Trans. Aerosp. Electron. Syst., Bd. 50, Nr. 1, S. 798–803, Jan. 2014.

[8] G. Pallotta, S. Horn, P. Braca, und K. Bryan, „Context-enhanced vessel prediction based on Ornstein-Uhlenbeck processes using historical AIS traffic patterns: Real-world experimental results“, in Information Fusion (FUSION), 2014 17th International Conference on, 2014, S. 1–7.

[9] G. Pallotta, M. Vespe, und K. Bryan, „Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction“, Entropy, Bd. 15, Nr. 6, S. 2218–2245, Juni 2013.

[10] G. de Vries und M. van Someren, „Recognizing Vessel Movements from Historical Data“, in Situation Awareness with Systems of Systems, P. van de Laar, J. Tretmans, und M. Borth, Hrsg. New York, NY: Springer New York, 2013, S. 105–118.

[11] Jan-Hendrik Oltman, „ACCSEAS North Sea Region Route Topology Model (NSR-RTM)“. 05-Mai-2015.

[12] I. Varlamis, K. Tserpes, M. Etemad, A. S. Júnior, und S. Matwin, „A Network Abstraction of Multi-Vessel Trajectory Data for Detecting Anomalies“, in Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference, Lissabon, 2019, Bd. 2322.

[13] V. Fernandez Arguedas, G. Pallotta, und M. Vespe, „Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring“, IEEE Trans. Intell. Transport. Syst., Bd. 19, Nr. 3, S. 722–732, März 2018.

[14] D. H. Douglas und T. K. Peucker, „Algorithms for the Reduction of the Number of Points Required to Represent a Digitizied Line or its Caricature“, Cartographica: The International Journal for Geographic Information and Geovisualization, Bd. 10, Nr. 2, S. 112–122, Dez. 1973.

[15] A. Lamm und A. Hahn, „Statistical Maneuver Net Generation for Anomaly Detection in Navigational Waterways“, in 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 2019, S. 1438–1443.

[16] T. I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control: Fossen/Handbook of Marine Craft Hydrodynamics and Motion Control. Chichester, UK: John Wiley & Sons, Ltd, 2011.

[17] A. Lamm und A. Hahn, „Detecting maneuvers in maritime observation data with CUSUM“, in 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao, 2017, S. 122–127.

[18] A. Dobrkovic, M.-E. Iacob, und J. van Hillegersberg, „Maritime pattern extraction and route reconstruction from incomplete AIS data“, Int J Data Sci Anal, Bd. 5, Nr. 2–3, S. 111–136, März 2018.

[19] A. Dobrkovic, M. Iacob and J.V. Hillegersberg "Using machine learning for unsupervised maritime waypoint discovery from streaming AIS data" in: Proceedings of the 15th Int. Conf. on Knowledge Technologies and Data-driven Business, ACM, 2015.

[20] J. L. Bentley, „K-d Trees for Semidynamic Point Sets“, in Proceedings of the sixth annual symposium on Computational geometry - SCG ’90, Berkley, California, United States, 1990, S. 187–197.