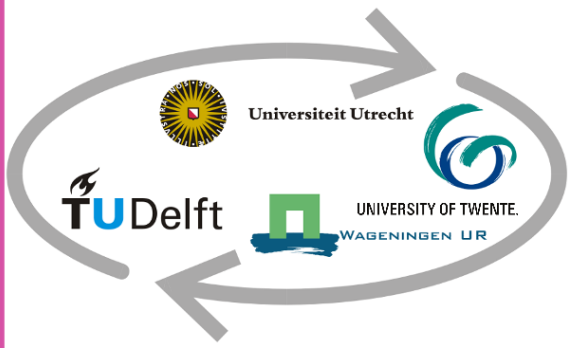
GIMA
Geographical Information Management and Applications

Detecting spatial ship behaviour patterns using clustering of static and dynamic information in AIS data

Master thesis

Student: Suzanne Maljaars
E-mail: j.j.maljaars@students.uu.nl
Student number: 5664683 (Utrecht University)
Supervisors: Martijn Meijers (TU Delft) & Yigit Can Altan
Responsible professor: Peter van Oosterom

Date: 28 February 2020



Abstract

AIS (Automatic Identification System) data is a relatively new type of data and contains a lot of information about ships and their voyage, both dynamic and static data. Dynamic data contains for example information about speed and location. Static data contains information about the ship, like name, sizes and destination. Because ships have to broadcast AIS data very frequently, for data-analysing purposes this means that the researcher has to deal with a large dataset. Different kinds of studies using AIS have been carried out. In many of them researchers try to find patterns within the data to use it for different purposes. To find patterns in large datasets, data mining techniques can be applied. One of the methods that can be used to find patterns in AIS data is clustering. Clustering is an unsupervised learning method. This means that the data will be grouped into clusters, in such a way that the objects within a cluster are more similar to each other than the objects in other clusters. This will be done without any pre-classification.

In this research clustering is used to analyse AIS data for three different patterns. To detect these patterns, non-spatial variables in AIS data are clustered. This means that the clustering method is not applied to get spatial clusters. However, the clusters found are visualised to examine the spatial distribution of the clusters. This is done using the research question: *To what extent can clusters based on ship behaviour conditions in AIS data be used to detect spatial ship behaviour patterns?*

First, based on literature, three patterns are found, which are tried to find using the clustering methods. These patterns were a combination of: 1. Length and beam, 2. Length, beam and speed over ground and 3. Speed over ground and distance to the fairway. The first pattern is based on the values for bow, stern, port and starboard in AIS data. The second pattern is based on the same variables as the first pattern, but speed over ground, retrieved from the AIS data, but also calculated by the researcher, is added as third variable. The third pattern is based on speed, the same values as the second pattern, and distance to the fairway. For data about the fairways, other data than AIS is used. For all patterns, different clusters are found, validated and visualised to get the spatial distribution of the clusters.

For the first pattern, clear different spatial patterns are found for different clusters. Adding a third dimension, speed, to this pattern, did not result in clear other patterns. For the third pattern, in which speed and distance to the fairway are used, patterns were found, but they are mainly related to the speed.

Table of content

Abbreviations

AIS: Automatic Identification System

ARI: Adjusted Rand Index

DBSCAN: Density-Based spatial Clustering of Applications with Noise

DBCV: Density-Based Clustering Validation

COG: Course over ground

CSV: Comma-separated values

GIS: Geographical Information System

GT: Gross Tonnage

IMO: International Maritime Organization

KRW: Kaderrichtlijn Water

MMSI: Maritime Mobile Service Identity

NMEA: National Marine Electronics Association

NWB: Nationaal Wegenbestand

RWS: Rijkswaterstaat

SOG: Speed Over Ground

SQL: Structured Query Language

VHF: Very High Frequency

WFS: Web Feature Service

WGS 84: World Geodetic System 1984

1. Introduction

Maritime transport plays an important role in the international trade system (Du, Monios, & Wang, 2019). The Netherlands has a hub function in maritime freight transport and within this country waterborne transport increases, because more cargo is transported over water and ships are getting larger (Helpdesk Water, 2018).

For safe navigation, sailors can use several tools like radar and a chart plotter (Tu, Zhang, Rachmawati, Rajabally, & Huang, 2018). Since 2004, the International Maritime Organization (IMO) requires ships to carry an Automatic Identification System (AIS) too. It is obliged for ships on an international voyage with a volume of more than 300 gross tonnage (GT) (IMO, n.d.). Also inland ships with a length of more than 20 meters are obliged to use an AIS-transponder (Centrale Commissie voor de Rijnvaart [CCR], 2015). AIS broadcasts static and dynamic information. Static information includes the name of the ship, the MMSI (a number to identify the ship) and the length of the ship. Dynamic information includes for example the position, speed and heading of a ship.

Ships broadcast their own AIS data and collect data from other ships and coastal stations receive this data too. Ships have to broadcast their data in time intervals between a few seconds to a maximum of three minutes. Sailors use AIS data to know the location of other ships, which is not always possible by using radar, like behind hills or around bends. As a result of the implementation of AIS, maritime safety, security and navigation efficiency are improved (Ou & Zhu, 2008; Tu et al., 2018).

Besides the use of AIS data by sailors, this data contains a lot of information for researchers. Yang, Wu, Wang, Jia & Li (2019) identified six categories in research based on AIS data. These categories are: AIS data mining, navigational safety, ship behaviour analyses, environmental analyses, trade analyses, arctic shipping and ship and port performances. This thesis will focus on ship behaviour. Tu et al. (2018) explain four fields of ship behaviour research in which AIS data can be used: anomaly detection, route estimation, collision prediction and path planning.

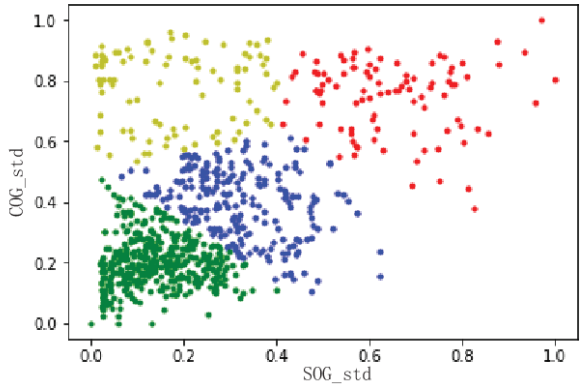
1.1. Existing research and problem

AIS data is a relatively new type of data. It contains a lot of information about ships, and because ships have to broadcast AIS data very frequently, for data-analysing purposes this means that the researcher has to deal with a large dataset. Different kinds of studies using AIS have been carried out. Many of them try to find patterns within the data to use it for different purposes. To find patterns in large datasets, data mining techniques can be applied.

One of the methods that can be used to find patterns in AIS data is clustering. Clustering or cluster analysis is an unsupervised learning method. This means that the data will be grouped into clusters, in such a way that the objects within a cluster are more similar to each other than the objects in other clusters. This will be done without any pre-classification. Cluster analysis can be used to find classes within the dataset (Zhou, Daamen, Vellinga & Hoogendoorn, 2019). Different methods of clustering exist, these will be discussed in section 3.2. This research will focus on using cluster analysis to detect ship behaviour patterns.

Some researchers use clustering for analysing AIS data. Hanyang, Xin and Zhenguo (2019) use patterns found by clustering trajectories to evaluate sailing stability of ships. They use cluster analysis to find clusters based on the standard deviation of course over ground and the standard deviation of speed over ground (Figure 1.1). They cluster based on trajectories. However, they do not use terrestrial AIS, but satellite AIS, because it has a wider range.

Figure 1.1: Example of clustering



Source: Hanyang, Xin & Zhenguo (2019)

In other studies AIS data is used to identify the collision risk. Li et al. (2018) also cluster trajectories by using a new clustering method based on the shortest merged distance between two trajectories, multidimensional scaling and an improved version of the Density-Based spatial Clustering of Applications with Noise (DBSCAN) algorithm. In another study, authors also use trajectory clustering, but they use it for extracting route patterns (Sheng & Yin, 2018). Their clustering method is based on location. Other authors cluster trajectories and combine AIS data with data of ocean currents to extract routes (Yilou & Dejun, 2018). Liu, Wu and Zheng (2019) use the DBSCAN clustering method in a proposed framework to identify regional collision risk. Wang, Claramunt and Wang (2019) cluster AIS data to find global shipping networks. Zhou et al. (2019) use clustering as a method to find ship behaviour patterns. These authors developed a new methodology in which they use clustering for classification. According to them, this can be used for prediction of ship behaviour patterns. They do not use their results for prediction themselves.

In these studies about clustering AIS data, researchers only use spatial clusters of data or only static data (length and beam (Zhou et al., 2019)). In this thesis, the author will try to find non-spatial clusters based on static and dynamic ship conditions using AIS data. The author also will visualize these clusters on a map and evaluate whether these clusters correspond to spatial patterns. Based on these patterns, more knowledge will be available about ship behaviour. If patterns occur, these can also be used as input for finding methods for ship behaviour prediction. When the behaviour of ships can be predicted more accurately, the safety on water might increase.

1.2. Research question and sub-questions

In this research, AIS data will be used to find spatial ship behaviour patterns. These patterns will be based on that characteristics of ships and their voyage that are relevant for prediction. The objective of this research is to investigate to what extent clusters based on these characteristics can be used to find spatial ship behaviour patterns. The clusters found might result in spatial behaviour patterns when visualizing these clusters in a Geographical Information System (GIS). Based on this objective, the central question is:

To what extent can clusters based on ship behaviour conditions in AIS data be used to detect spatial ship behaviour patterns?

To find an answer to this research question, the following sub-questions need to be answered. The methodology to find answers to these questions can be found in section 1.3.

SQ1. What are the behaviour patterns to detect and how to define them?

SQ2. Which conditions are relevant to find spatial ship behaviour patterns?

SQ3. Which clustering algorithm is suitable for clustering AIS data and how to apply this algorithm?

SQ4. How to visualise the clusters found to find spatial behaviour patterns?

1.3. Research methodology

The research will be carried out by following the sequence of the sub-questions. In this chapter, the methodology will be explained for each sub-question.

The first sub-question is: What are the behaviour patterns to detect and how to define them? In this sub-question, different ship behaviour patterns that can be detected using AIS data will be discussed. As a result of this sub-question, it will be clear which kind of patterns will be detected in this research. The patterns that will be found are related to prediction. This will be based on literature review.

The second sub-question is: Which conditions are relevant to find spatial ship behaviour patterns? This sub-question will be answered based on literature. AIS data will be explained first, then the information from AIS data relevant to find the patterns will be discussed. The conditions found will be used as input for the clustering algorithm, which will be done in sub-question 3.

The third sub-question is: Which clustering algorithm is suitable for clustering AIS data based on conditions and how to apply this algorithm? For cluster analysis, different cluster algorithms exist (Lei, 2019). To find out which clustering algorithm will be suitable for clustering AIS data and finding the patterns, a literature review will be carried out. This literature review will also focus on how to measure the performance of the clustering algorithm. The algorithm which seems to be the best suitable one will be used in this research. This will also be done in this step. The best practical method to apply the algorithm will be found in literature and explained in the methodology chapter, chapter 4.

The fourth sub-question is: How to visualise the clusters found to find spatial behaviour patterns? In this sub-question, the clusters found will be visualized in a GIS. With this visualisation, it will be clear what the geographical locations are of the non-spatial clusters found. Based on this visualisation, it will be clear to what extent clusters found also occur as spatial patterns.

1.4. Scope and limitations

The patterns found by this research can be used for ship behaviour prediction purposes. However, this research itself will not result in prediction of ship trajectories. It will only try to find spatial patterns based on clusters of non-spatial conditions, that might be used for prediction purposes. In this research, non-spatial conditions are the characteristics of the ship and/or voyage that are not related to the exact location of the ship. This means that longitude and latitude will not be used in the cluster analysis. Examples of conditions that might be used are ship length, course over ground and speed over ground. The method will be used to find spatial patterns, for example ship routes based on the length and beam of a ship. If this results in clear spatial patterns for ships with different lengths and beams, different tracks for different kinds of ships will be clear. This can be used as input for prediction, for example when ships with specific conditions have another main route or main routes than ships with other conditions. This research will only focus on the Westerschelde. The research area will be explained in more detail in the methodology chapter.

1.5. Reading guide

In this report, first, a literature review about patterns, conditions and AIS data is given. This information can be found in chapter 2. The third chapter contains information about clustering: different clustering algorithms will be compared and clustering validation will be explained. In chapter 4, the methodology for the research will be discussed in more detail. In this chapter, the data and research area will be described. In this chapter, also all practical steps will be explained including the software used. Chapter 5 contains the relevant results for this research. Based on these results, chapter 6 contains a conclusion including a discussion and recommendations for future research.

2. Behaviour patterns and conditions

To detect spatial patterns using clustering, it is important to know which conditions are relevant to find these patterns. In this chapter, these conditions will be found. This chapter will be divided into three sub-sections. Before answering SQ2 about the conditions relevant to find spatial behaviour patterns, first, it is important to know what patterns will be detected. In this section, the type of patterns that might be detected will be explained in 2.1. These patterns are summarised in a conceptual model. In section 2.2 ship and voyage related conditions will be discussed, which can be found in AIS data. This section also contains the relevant conditions for the process of finding patterns.

2.1. Ship behaviour patterns

In this sub-section, the first sub-question will be answered. This sub-question is: What are the patterns to detect and how to define them? First, different kinds of patterns that can be found using AIS data will be discussed. Second, patterns that might be detected by this research will be explained.

Ship behaviour has already been examined in different ways. When relating ship behaviour to AIS data, Tu et al. (2018) identify four aspects of behaviour to which AIS data can be used. They discuss traffic anomaly detection, route estimation, collision prediction and path planning. Anomaly detection can be classified into three groups: position anomaly, speed anomaly and time anomaly. Route estimation is used to predict the future position of vessels. For collision prediction, the collision risk is assessed. This is done based on the expected trajectories of different ships. These trajectories might cross each other. If this is the case, real time AIS data plays an important role, because based on the speed and the near real-time location of the ship, collision risk can be detected. This can be used as input for path planning, for example to choose another route to avoid collision (Tu et al., 2018).

For each aspect, different researchers already tried to use AIS data to detect patterns, because much information is given in this type of data. Hanyang et al. (2019) use AIS data received from a satellite to detect anomalies. They use the trajectories of ships to find locations on the sea around South-Africa where ship sailing behaviour is most unstable, based on the direction and speed of the ship. Some of the areas found are related to fishing ships. These ships have an unstable behaviour, because the sailing direction and speed changes often. Other areas with unstable ship behaviour are areas around ports (Hanyang et al., 2019). In other articles, other anomalous ship behaviours are detected. Lane, Nevell, Hayward, & Beaney (2010) have found five anomalous ship behaviours that can be detected using AIS data: deviation from standard route, unexpected AIS activity, unexpected port arrival, close approach and zone entry.

AIS data can also be used for route prediction. Lo Duca, Bacciu and Marchetti (2017) created a model in which they split the area of interest in cells, to predict the probability a ship will sail into a specific cell. They use location, speed and heading from AIS data for predicting the ship route. For the prediction algorithm, they use the K-Nearest Neighbour classification. Zhou et al. (2019) do not predict routes by themselves, but use AIS data as input for clusters, which they use as input for classification. This classification can be used for ship prediction based on their characteristics. According to these authors, length and beam of ships are explanatory variables for classification. These authors also have found a relationship between the location of ships on the waterway and the speed of these ships in the port of Rotterdam.

Collision risk is discussed by Lei (2019) and Silveira, Teixeira & Soares (2013). Lei (2019) created a model to automatically identify the clusters of conflicting trajectories, based on AIS data. Silveira et al. (2013) assess the number of possible collisions based on prediction of the future distances between ships. Their model uses AIS data as input and compare their result with the number of collisions that really have occurred.

It is also possible to identify main routes or networks based on AIS data. Yilou & Dejun (2018) use the location as transmitted by AIS data to analyse the sailing trajectory of ships in a river estuary. They use the DBSCAN clustering method to cluster the trajectories. They also use an interpolation method to find the centre track line. This track line can assist people who navigate on the waterway. Wang et al. (2019) use AIS data to extract global shipping networks. They use an algorithm based on the DBSCAN algorithm to detect stopping points. The stopping points detected are related to the ports that exist on that locations. Between those points, links are created based on ship journeys.

The author will try to find patterns related to prediction of ship routes and route patterns. In this research, three patterns will be examined. The patterns that will be examined are explained below, the conditions that are relevant for examination will be explained in section 2.2, the clustering method that will be used will be explained in chapter 3 and 4. The research area that will be used to find the patterns is the Westerschelde, the estuary of the Scheldt river, where ships of different sizes and speeds navigate between the North Sea, Vlissingen, Terneuzen/Gent, Hansweert and Antwerp.

1. The first pattern to detect are the routes used by different ship sizes. According to Zhou et al. (2019) length and beam are related to the ship's place on the waterway. However, these authors do not cluster these characteristics themselves and also do not show routes in their research area. So that will be done in this research. The characteristics can be derived from AIS data. Based on this pattern, it will be clear whether ships of different sizes follow the same route on the Westerschelde or not.

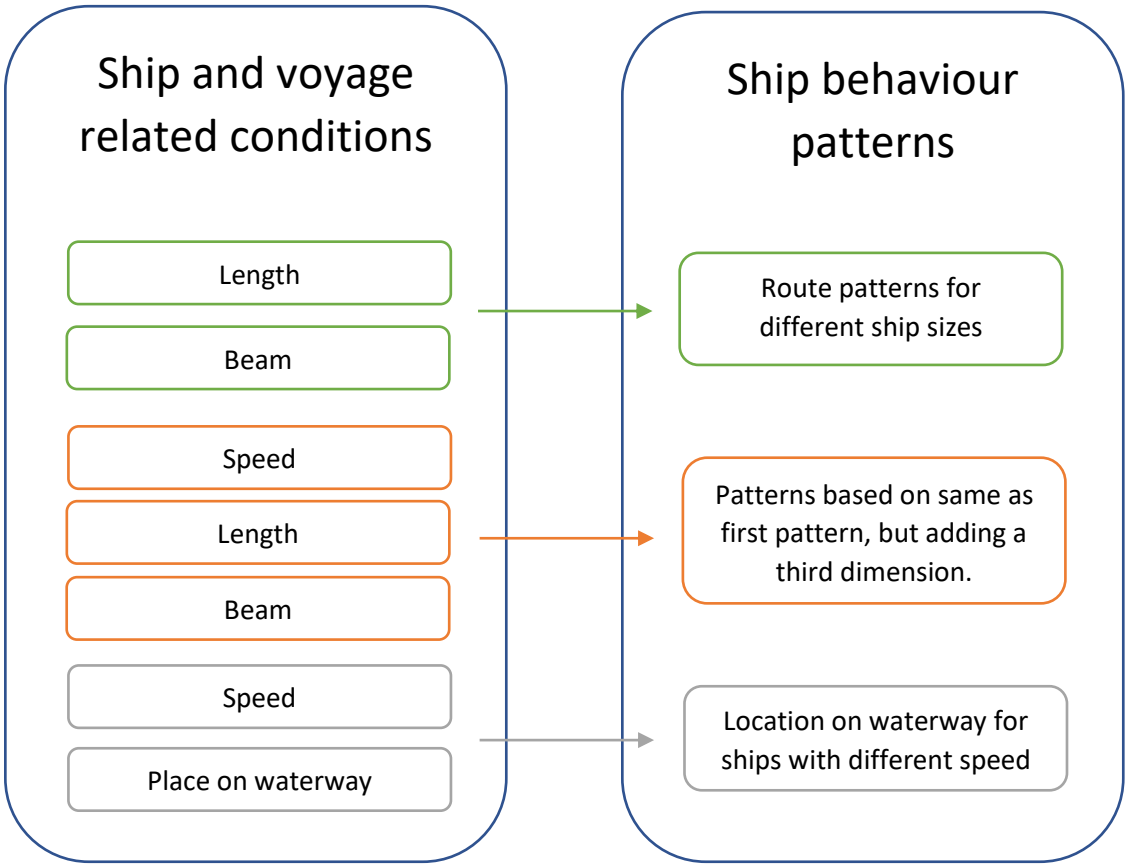
2. Clustering can be done in two dimensions, as will be done for the first pattern, but it is also possible to cluster in higher dimensions. In this research, the second pattern to be found will focus on three-dimensional clustering. The second pattern to examine is based on route estimation as explained by Tu et al. (2018). According to them, ship behaviour differs from vehicles and aircraft. This is related to the speed and sailing direction of a ship. A ship cannot stop abruptly or change their direction fast. Also Lo Duca et al. (2017) use the direction of the ship and the speed for predicting purposes. Based on these articles can be concluded that speed is a relevant variable in detecting ship behaviour patterns. This will be the third dimension of this pattern and will be added to the first pattern. This means that the second pattern will focus on clusters based on speed, length and beam.

3. Besides examining a relationship between speed, length and beam of ships to detect patterns, Zhou et al. (2019) also relate speed to the place on the waterway. These authors detect the place on the waterway based on the distance to the banks. This will be interesting, because this is not used to identify the track for A to B, but to find out whether ships with a higher speed make use of another place of the waterway than ships with a lower speed on the Westerschelde. This will be the third pattern to examine in this research.

This research might be used as input for ship behaviour prediction, so the patterns detected in this research are related to prediction. When a track for a certain pattern is known, this can be used to predict the location on the waterway for ships that have the same conditions as the cluster used as

input for the track. Based on the patterns as described, Figure 2.1 shows a summary based on these patterns.

Figure 2.1: Summary of patterns to find in this research



2.2. AIS data and relevant conditions for clustering

After defining the patterns that will be examined in this research, the conditions in AIS data related to these patterns will be discussed. First, AIS data itself and the information provided will be explained. Second, for each pattern that will be examined in this research, the conditions needed for clustering, which can be retrieved from AIS data, will be discussed. Third, information outside the AIS data that are relevant for the empirical part of the research will be discussed. At the end of this section, an answer will be given to SQ2: Which conditions are relevant to find spatial ship behaviour patterns? The conditions found will be used as input for the clustering algorithm, which will be discussed in chapter 3.

2.2.1. AIS data

As explained in the previous section, AIS data can be used to find different patterns in ship behaviour. In this sub-section, AIS data will be explained in detail.

As written in the introduction, ships on an international voyage with a volume of more than 300 GT and inland ships with a length of more than 20 meters are obliged to use an AIS-transponder. This means that for example small pleasure crafts are not obliged to use an AIS transponder, however it is recommended to install an AIS system in these boats too (CCR, 2015; IMO, n.d.). Ships with an AIS system automatically broadcast their information via a VHF radio channel. Three different categories of information are included in AIS data: static, dynamic and voyage related information. For each category, the information is transmitted frequently. Static and voyage related information is broadcasted every 6 minutes. Dynamic information is broadcasted every 3 minutes when a ship has stopped and between 2 and 10 seconds when a ship is sailing (CCR, 2011). When downloading the data, depending on the time range and geographical area, this might results in a large dataset. This data results in point features when plotting it on a map; one point for each transmitted AIS signal (Lei, 2019).

Table 2.1: Static AIS data

Static	**Data type**
MMSI number	Nine-digit number
IMO number	Seven-digit number
Timestamp	UTC in seconds
Call sign	1-3 letters followed by 1-3 letters or 4 numbers
Ship name	Up to 20 characters
Ship type	Two digit number
Bow	Meters
Stern	Meters
Port	Meters
Starboard	Meters

Source: (Ou & Zhu, 2008; MarineTraffic, 2017)

Table 2.1 shows the static information transmitted by AIS transponders. The MMSI and IMO numbers can be used to identify the ship which is transmitting the data. The difference between these identification numbers is, that the IMO number is related to the hull of the ship. This number does not change when a ship is sold or the name or flag changes. The MMSI number does change when a ship gets a new owner. This number is used for contacting purposes. The first three digits in the MMSI number are related to the nationality, e.g. 245, 246 and 247 represent the Netherlands. In AIS data, MMSI is used as identifier. When using a large historical dataset of AIS data and searching for a vessel using the IMO number, it is possible that more than one vessel will be found, because a ship can have different owners, which means that the MMSI changes (Retsch, 2018). In the AIS data used for this thesis research, the MMSI will be used as identifier, because it is both in the static and dynamic part of the dataset (Pallotta, Vespe, & Bryan, 2013). Other identifiers, like IMO number, but also name and callsign of the ship can be found only in the static part of the dataset. Using the MMSI makes it possible to use both parts together.

Other information provided in the category 'static' is ship type and information related to the ship size. The values used to identify the ship type are codes that are defined as a two-digit number. Each code represents a specific ship type, e.g. tankers or cargo vessels (Ou & Zhu, 2008). Information related to ship size contains the distance from the AIS antenna to bow, stern, port and starboard (Svanberg, Santén, Hörteborn, Holm, & Finnsgård, 2019).

Table 2.2: Dynamic AIS data

Dynamic	**Data type**
Latitude	Up to 0.0001 minutes accuracy
Longitude	Up to 0.0001 minutes accuracy
Timestamp	UTC in seconds
Navigational status	1-15
Rate of Turn (ROT)	Right or left (0-720 degrees per minute)
Speed over Ground (SOG)	0 to 102 knots (0.1 resolution)
Course over Ground (COG)	Up to 0.1° relative to true North
Heading	0 to 359 degrees

Source: (Ou & Zhu, 2008; MarineTraffic, 2017)

In the category with dynamic information (Table 2.2), for navigational status codes are used, that indicates if ships are underway, moored, at anchor etc. COG and heading are both related to the sailing direction. COG describes the direction a ship is sailing to. Heading is related to the direction the ship is pointing to (Zhou et al., 2019). COG and heading may differ because of the influence of wind and current.

The last category in AIS data is voyage related information (Table 2.3). This part contains information about the Estimated Time of Arrival, destination and draught and is transmitted together with the static data. To keep this data up-to-date, this data has to be updated for each voyage by the ship's crew.

Table 2.3: Dynamic AIS data

Voyage related	**Data type**
Estimated Time of Arrival (ETA)	Month, day, hour, minute
Destination	Up to 20 characters
Draught	0.1 to 25.5 meters

Source: (Ou & Zhu, 2008; MarineTraffic, 2017)

Although it is possible that all this information is provided by AIS data, in reality some information is missing or wrong (Harati-Mokhtari, Wall, Brooks, & Wang, 2007). MMSI, timestamp, ship name, position, SOG, COG, ship size, type and navigational status are required for inland ships. This means that ROT, heading and voyage related information might be missing or not up to date (CCR, 2015). A value for heading for example is given for many points, but often, it is not available. This is indicated by the value 511 (United States Coast Guard [USCG] Navigation Center, 2019). Another example is the destination that is not always correct. In the dataset used for this research, some ships have destinations like 'Thuis', 'Sesamstraat' or a website. Also ships smaller than 20 meters are not required to carry an AIS system. This means that not all ships are 'visible' by using AIS data (CCR, 215).

2.2.2. Conditions relevant for pattern detection

Based on the patterns as explained in 2.1 and the explanation of AIS data as given in the previous part, the AIS conditions relevant for detecting the patterns will be discussed here. These conditions will be used as input for the clustering algorithm. All patterns will be based on points as input for the clustering, not on trajectories. For each pattern, a description of the conditions needed will be given. Also an explanation will be given how these conditions can be used for pattern detection.

The first pattern to detect is the route for specific ship sizes. According to Zhou et al. (2019), length and beam are relevant for the position of a ship on the waterway. However, they do not cluster these characteristics themselves. That will be done in this research.

Figure 2.2: Beam of a ship

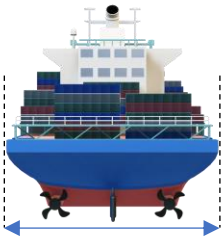
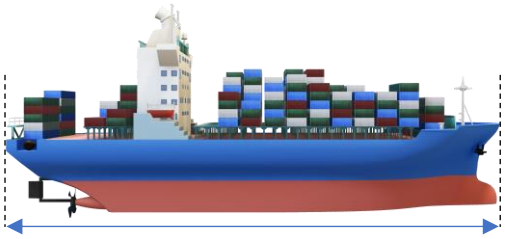
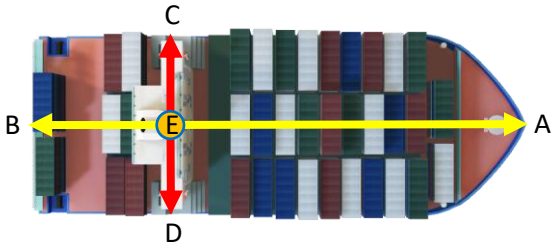


Figure 2.3: Length of a ship



Before explaining the data from the AIS dataset needed, it is important to know what is meant by the ship's length and beam. Figure 2.2 shows the beam of a ship. This is the widest width of a ship. This is often used to know whether it is safe to navigate along an obstacle (Bruno, 2019). AIS data does not contain a value called beam or width. However, ship dimensions are included in AIS data. The characteristics used form AIS data are called dimension to bow, dimension to stern, dimension to port and dimension to starboard.

Figure 2.4: Ship dimensions



These dimensions will be explained by using Figure 2.4. A is the ship's bow, B the stern, C is port and D is starboard. E is the antenna used to transmit the AIS signal. To calculate the length of the ship (as shown in Figure 2.3), the distance from A to B has to be calculated. This can be done by using the calculation:

$$AE + BE = length$$

The beam of the ship, as shown in Figure 2.3, can be calculated by using the calculation:

$$CE + DE = beam$$

Ships with a length greater than 511 meters have the value 511 for their length. Ships with a beam of more than 63 meters have a value of 63 for their beam (Fluit, 2011).

The second pattern to examine is based on length, beam and the speed of a ship. The speed is included in the AIS dataset and is called Speed over Ground (SOG). To know the speed of a ship, it is also possible to use the speed over water (Jassal, 2016). This means the distance travelled with respect to water. This is not included in AIS data. According to Kornacki, Mazurek and Smolarek (2009), the SOG of ships is influenced by currents, which might also occur on the Westerschelde.

The last pattern that will be detected in this research will be based on the relationship of speed and the place on the waterway. This is based on the article of Zhou et al. (2019). For this pattern AIS data is needed, but also information about the waterway. For speed, SOG will be used. For the location on the waterway, Zhou et al. (2019) use the distance from the ship to one of the banks in their research area.

3. Clustering

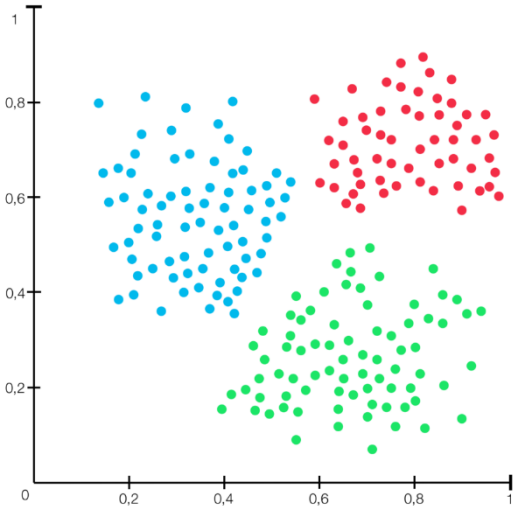
In the previous chapters, the patterns that will be detected and the conditions needed for it are discussed. This chapter will focus on the next step in this research: clustering of the data. This chapter focuses on SQ3: Which clustering algorithm is suitable for clustering AIS data and how to apply this algorithm?

In this chapter, first, the method of clustering will be explained: what is clustering and for what purpose can it be used? In the second section, different clustering algorithms will be compared. This comparison will be used as input for the methodology. The third section will discuss validation of the clusters and patterns. In the fourth section, the practical method to cluster AIS data will be discussed.

3.1. Clustering explanation

Before discussing possible clustering methods, it is important to know what clustering actually is. Clustering is defined as: *"a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups"* (Berkhin, 2002, p.2.). This means that patterns can be found, that are not clear without clustering. Clustering is a unsupervised learning data mining technique. This means that the input data does not have labels for the clusters. An example of clustering can be found in Figure 3.1.

Figure 3.1: Cluster example



Source: Marzell (2019)

In the example in Figure 3.1, the cluster analysis uses two dimensions as input. However, it is also possible to add a lot of dimensions in a cluster analysis. These dimensions are based on the attributes of the dataset (Berkhin, 2002). In this research, the clustering analyses will be two dimensional for the first and third pattern and three dimensional for the second pattern. To find clusters in a dataset, a clustering algorithm is needed. Many different kinds of clustering exist. In this research, two kinds will be discussed in the next section: hierarchical, partitioning and density based methods.

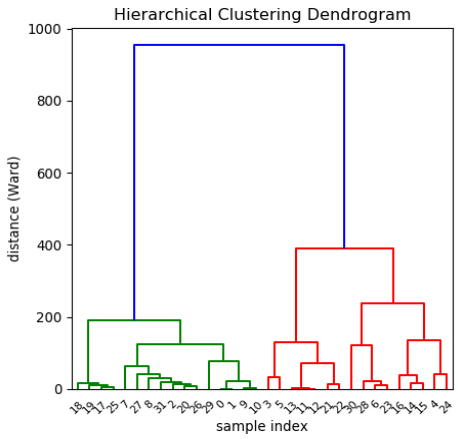
3.2. Clustering algorithms

As section 3.1 explained, a distinction can be made between different clustering categories. In this section, some categories and algorithms that are part of these categories will be discussed. The categories that will be discussed here are hierarchical, partitioning and density-based partitioning methods.

3.2.1. Hierarchical methods

In hierarchical clustering, the dataset is divided into as small clusters as possible and then the algorithm will merge clusters to most similar clusters. This repeats until all clusters are merged to one final cluster. A hierarchical cluster can be represented as a dendogram (Figure 3.2). This is a kind of tree-structure where each branch is a divided into a smaller branch with the smallest cluster as leaves (Cohen-Addad, Kanade, Mallmann-Trenn, & Mathieu, 2017). Two hierarchical clustering methods will be explained here: agglomerative and divisive clustering.

Figure 3.2: Dendogram example



Source: Holtz (2017)

In agglomerative clustering, all points in the dataset are set at the bottom of the dendogram. At each level, the most similar clusters will be merged (Reddy & Vinzamuri, 2019). This is the same as the bottom-up clustering algorithm as explained by Li et al. (2018). These authors also mention a hierarchical clustering algorithm called top-down algorithms. These algorithms are the opposite of agglomerative algorithms. In top-down algorithms, all points in the dataset are clustered as one cluster and will be divided into smaller ones in the next step. This is the divisive clustering method, as explained by Reddy and Vinzamuri (2019). They prefer this method to the bottom-up method, because it is more efficient when it is not needed to get the full hierarchy down to each individual point.

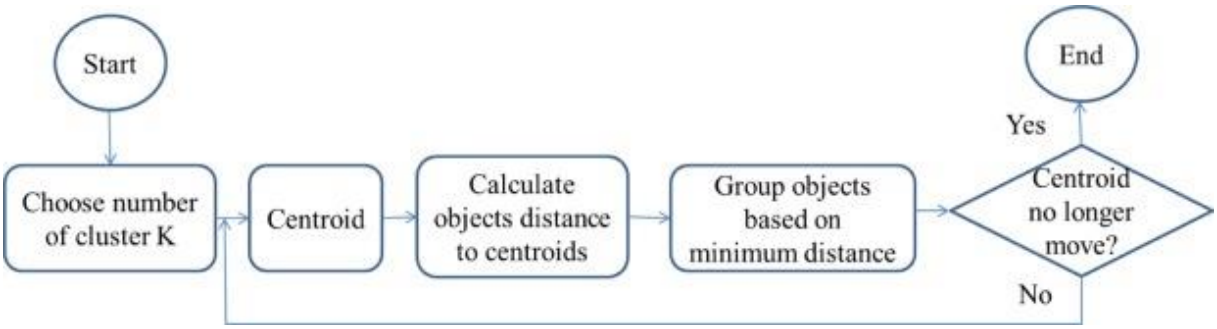
3.2.2. Partitioning methods

The second category of clustering methods that will be discussed are partitioning methods, also known as partitioning relocation methods. In partitioning-based cluster algorithms, the dataset will be divided into partitions of the data. The amount of clusters is defined by the researcher. To obtain the best result, the criterion used is that objects within clusters should have the largest similarity possible and similarity between clusters should be as small as possible (Li et al., 2018). The optimal clustering result can be obtained by using iterative relocation, which means that objects are relocated between the clusters, until the result meets the criterion (Berkhin, 2002). Clusters in reality can have all shapes. However, partitioning methods only find clusters with a spherical shape, because the objects are assigned to the nearest cluster (Li et al., 2018).

Swarndeep and Pandya (2016) discuss four types of partitioning methods. These are the K-Means algorithm, the K-Medoid Algorithm, CLARA and CLARANS. The K-Means algorithm is a popular clustering algorithm. The name of this algorithm refers to the centroid, the mean of all points in a cluster, which is the centre of the cluster. This represents the cluster. The letter K comes from the amount of clusters, which is represented by K (Berkhin, 2002). When using the K-Means algorithm, first, K objects from the dataset are chosen as initial centroids. In the second step, all objects in the dataset are assigned to the centroid with the lowest dissimilarity to the object. This is based on the

Euclidean distance. Because distance is used in this clustering algorithm, it can only be used for numerical data. For the third step, when all objects are assigned to a cluster, the centroids will be updated by calculating the new means. The second step will be repeated after this update. This process is finished when no changes occur in the centroids (Saxena et al., 2017; Zhou et al., 2019). The process is shown in Figure 3.3. The result of this method strongly depends on the choice of the initial centroids and the value chosen for K (Saxena et al., 2017). It is important to choose the right K, because too many clusters do not give a good representation of patterns in the dataset. Too few clusters might combine some patterns, so this also means that this is not representative (Zhou et al., 2019).

Figure 3.3: K-Means clustering process



Source: Saxena et al. (2017)

A method to identify the number of clusters (K), is using the Elbow method. This means that for a range of values for K the sum of squared errors will be calculated. The results will be plotted as a line chart, this might look like an arm. The location of the 'elbow' shows the best value for K. This can be found by a visual inspection. The line after this value should have a linear decrease (Bholowalia & Kumar, 2014; Gove, 2017; Hanyang et al., 2019). Another method to find the best number of clusters is using the average silhouette score. The silhouette index indicates the compactness and separation of clusters per possible K (Rendón, Abundez, Arizmendi, & Quiroz, 2011). The resulting average scores are between -1 and 1, with 1 indicating a well clustered result (Scikit-Learn, 2020b). Zhou et al. (2019) use the statistical t-test to find the optimal number of clusters. The data in two clusters need to be significantly different, which can be tested by using this method. The initial centroid can be found by selecting the minimum and maximum values and percentile values for a behaviour attribute (Zhou et al., 2019).

The K-Medoids algorithm is an adapted version of the k-means algorithm. Instead of calculating the means for each cluster, a representative object will be found for each cluster. This is called the medoid (Reynolds, Richards, & Rayward-Smith, 2006). An example of an algorithm that uses medoids is the Partition Around Medoid (PAM) algorithm. The advantage of such algorithms is that the sensitivity to outliers in the data is lower. The disadvantage compared to the K-Means algorithm is that it has a higher time complexity (Swarndeep & Pandya, 2016).

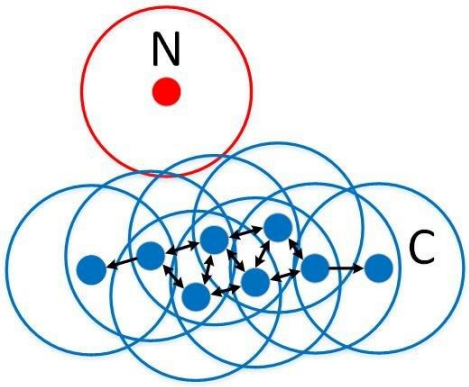
CLARA (Clustering for LARge Applications) is related to the K-Medoids algorithm. The K-Medoids algorithms finds medoids for the whole dataset, CLARA draws a sample of the dataset and finds medoids for the sample using the PAM algorithm. For the best clustering result, CLARA draws multiple samples (Ng & Han, 2002). Ng and Han (2002) developed an improved version of CLARA called CLARANS (Clustering LARge Applications based on RANdomized Search). The aim of this algorithm is to find spatial structures in data. This algorithm can be used to cluster polygons efficiently.

3.2.3. Density-based partitioning methods

Density-based methods are, as the name suggests, methods that cluster data based on density. High-density areas are separated from areas with a low density. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a well-known density-based clustering algorithm. It is introduced by Ester, Kriegel, Sander and Xu (1996). DBSCAN does not need a predefined number of clusters, the algorithm finds them automatically (Li et al., 2018). It can find clusters of any shapes, while for example K-Means clustering only finds spherical shaped clusters. It can also eliminate the noise (Liu et al., 2019). The DBSCAN algorithm uses two input parameters: Eps and Minpts. Eps specifies the distance between two points to belong to the same cluster. Points are neighbours when the distance between them is lower or the same as the Eps. Minpts is the value for the minimum of objects to form a cluster.

The first step involved in this algorithm is to choose a point *p*. In the next step all points that can be reached from point *p* will be retrieved using Eps and Minpts. A cluster will be formed in the next step, only when *p* is a core point, which means that *p* is inside the cluster. In the fourth step, when *p* is detected as a border point, which means that this point is on the border of the cluster, the algorithms will check the next point in the database. These steps will be repeated until all objects in a database are checked (Parimala, Lopez, & Senthilkumar, 2011). Points that are not part of a cluster are called noise points (Liu et al., 2019). An example of the DBSCAN method is shown in Figure 3.4. The blue points (C) are a cluster, the red point (N) is a noise point (Wu, Taheri, & Kolmanovsky, 2018).

Figure 3.4: Example of DBSCAN clustering



Source: Wu, Taheri, & Kolmanovsky (2018)

The DBSCAN algorithm is an efficient algorithm to find spatial clusters in large spatial databases (Saxena et al., 2017). Also some researchers who examine ship behaviour use the DBSCAN method. Liu et al. (2019) use to find clusters of ships as input for detection of collision risk. Yilou and Dejun (2018) use this algorithm to cluster trajectories and find main tracks of ships. Li et al. (2018) use the DBSCAN as input for trajectories too. Sheng & Yin (2018) use a revised version of this algorithm to find shipping routes. Wang et al. (2019) use the DBSCAN algorithm to find stopping points of ships which they use as input for extracting global shipping networks.

3.3. Clustering validation

To check the performance of a clustering algorithm, different approaches exist. Rendón et al. (2011) distinguish three kinds of clustering validation: internal, external and relative. Internal validation is based on the information intrinsic in the data. For external validation, knowledge form outside the data is used. Relative validation compares the results of the clustering with clustering with for example other values for K in K-Means. All validation approaches focus on two main aspects: the compactness of the data, this means the closeness of the points in the cluster and separability. This is the distinction between two clusters. For each validation approach, different indices can be used. The indices used for validation of clustering in this research are described in the next chapter, chapter 4.

4. Methodology

Based on the literature as discussed in the previous chapters, this chapter contains the methodology for the practical part of this research. In 4.1 the research area will be explained. In 4.2 the data will be explained shortly, because most data (AIS data) is already explained in 2.3.1. Section 4.3 contains the software used for the practical part of this research. Section 4.4 contains the practical method for each pattern to be found, including visualisation. Section 4.5 contains a research design scheme based on the methodology.

4.1. Research area

The research area of this research is the Westerschelde. This is an estuary of the Schelde river and is located in Zeeland in the south-west of the Netherlands. As explained in 2.3.2, the Westerschelde contains many shoals. These are visible in Figure 4.1. These makes it a difficult area for captain to navigate on this waterway. Therefore, ships are required to have a river pilot aboard (Scheldemonden, n.d.). The Westerschelde is directly connected to three port areas: Vlissingen-Oost, Terneuzen and the port of Antwerp in Belgium. At Terneuzen, ships can also sail into a canal to visit the port of Ghent. At Hansweert, ships can sail into a canal to visit the port of Rotterdam.

Figure 4.1: Research area: Westerschelde

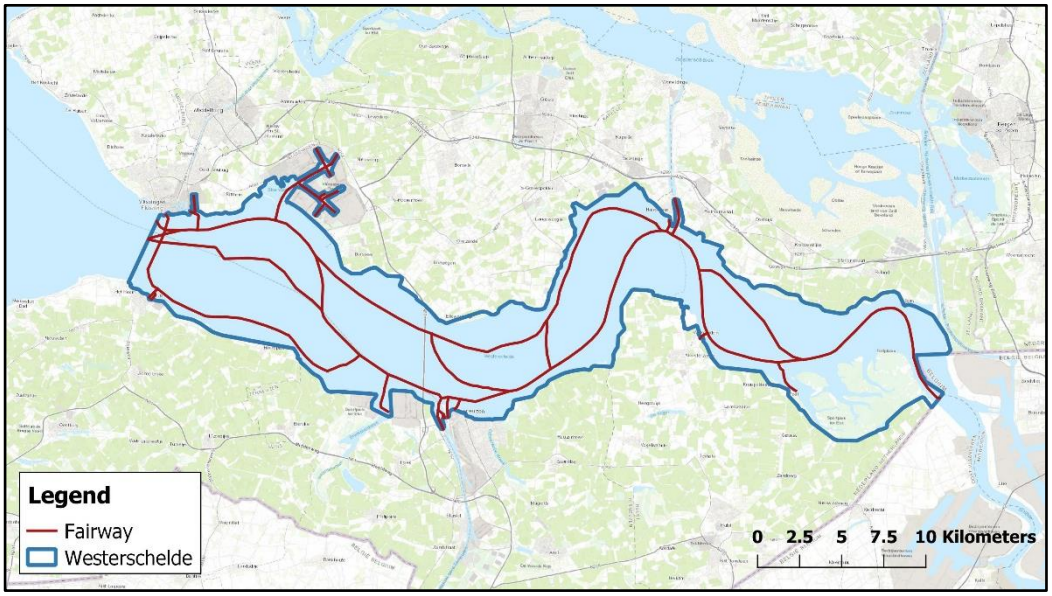
On the Westerschelde, a strong tidal stream might effect the speed of ships. This can be taken into account in clustering, for example by adding a condition containing the tide. This can be combined with heading, to know whether a ship has advantage or disadvantage of the tide. However, this will not be done in this research. This research will mailny focus on AIS data, not related to external data.

In the Westerschelde, ships cannot sail everywhere because of shoals (Figure 4.2). Shoals are naturally created accumulations of sand in a river or sea. These shoals rises from the bed to near the water level. They can also rise to above the water level (Encyclopeadia Brittannia, 2010).

Figure 4.2: Shoals in the Westerschelde (J. van Houdt)

This means that other input than banks, as used by Zhou et al. (2019), is needed to calculate the place on the waterway. For this research, the distance from a ship to the center of the fairway will be calculated. This information is not included in AIS data, so external data will be used. The location of the fairways can be found in Figure 4.3.

Figure 4.3: Fairways Westerschelde



4.2. Data

The AIS data is received from one of the supervisors and can be retrieved from sources like MarineTraffic. Other sources that can be used to download this data are for example AIShub and VesselFinder (Cotteleer, 2019). The dataset received contains AIS data from 10 consecutive days, from 10-19 December, 2016.

This data is transmitted by ships with mostly a length of at least 20 meters. This AIS data includes: MMSI, name of the ship, callsign, type of the ship, IMO number, length, beam, position, speed over ground, course over ground, heading, the moment the data was transmitted, draught and destination. A more detailed explanation of this data can be found in section 2.2.1. The AIS data as downloaded is NMEA data, data from the National Marine Electronics Association. This is decoded to comma-separated (CSV) files. The downloaded dataset contains AIS data from a larger area than the Westerschelde. The data is clipped by using the Westerschelde from Kaderrichtlijn Water (KRW) Oppervlaktewaterlichamen Nederland 2014 from Rijkswaterstaat (RWS) (2014).

Other data used is data about the fairways. The dataset used is 'Nationaal Wegen Bestand (NWB) - Vaarwegen - vaarwegvakken (RWS)' (Rijkswaterstaat, 2018a). This dataset contains fairways within the Netherlands and is retrieved via WFS from data.overheid.nl. This dataset is clipped by using the Westerschelde area from the KRW Oppervlaktewaterlichamen.

For visualisation in maps, as basemap the topographic basemap in ArcGIS Pro is used. Only for Figure 4.1 the basemap used is 'imagery' because of visualisation of the shoals in the Westerschelde.

4.3. Software

To carry out this research, different software is used. To decode the data from the NMEA file, a Python script, received from one of the supervisors, is used. The data was converted into CSV files. QGIS is used to select the Westerschelde from the Oppervlaktewaterlichamen file. The programme is also used to clip the AIS data based on the Westerschelde. Also the fairways are clipped in QGIS by using the Westerschelde area. In ArcGIS Pro, the results of the clustering and the dynamic data are joined into one file based on the MMSI. However, not for all MMSI values in the dynamic AIS data, an MMSI value for the static data is available. The data without static data is not used for the clustering in first pattern, only for visualisation. To manage the data in an efficient way, a PostgreSQL database is created and the Structured Query Language (SQL) of this system used to store, manipulate and receiving the data. The queries used can be found in Appendix I. For clustering and validation, Python is used. The Python modules used can be found in Table 4.1. For the visualisation part, ArcGIS Pro is used. In the next section the steps per pattern will be discussed and will be explained how the software is used.

Table 4.1: Modules used in Python

Python module	**Purpose**
Matplotlib	Data visualisation
NumPy	Array-processing
Pandas	Create data frame
Psycopg2	Connection to Postgres database
SciPy	Statistical measures
Seaborn	Data visualisation
Scikit-Learn	Clustering and validation indices

4.4. Method to detect ship behaviour patterns

The practical part of this research is subdivided into three phases: pre-processing, clustering, validation and visualisation. The first pattern in this research is based on the clusters of beam and length. The second pattern is based on clusters in length, beam and speed over ground. The third pattern is based on speed over ground and distance to the fairway. In this section, the method will be explained for each phase. All steps for the first pattern are done for 1 day first, when all steps were clear, seven days of data were used. For the other patterns, only seven days of data are used.

4.4.1. Data pre-processing

The first phase is the pre-processing phase and includes all data pre-processing steps that are required before the actual clustering can be done.

As explained in section 4.3, the AIS data is stored in NMEA files. To decode this data, a Python script is used. This resulted in two csv files, one with dynamic and one with static data. The dataset received

contains AIS data for ten consecutive days, but for the research, 7 days are used. The data decode in csv files per day, to use one day as test for the different steps. After all steps are done for one day, the data was merged for seven days and all steps have been carried out for the seven days dataset.

After the csv's were created, the csv containing the dynamic data was loaded in QGIS. As explained in section 4.1, the research area is the Westerschelde. The dataset received contains data from a rectangular area between the coordinates: 0.0, 50.0 (south-west) and 6.0, 53.0 (north-east). This data is clipped to get the data for the Westerschelde only. This was exported to a csv and imported in a Postgres table.

Pattern 1

The csv containing the static data was uploaded to the Postgres database too. Because the first pattern only focuses on the length and beam, only the static data is used as input for clustering. The tables including the static and dynamic data were joined based on the MMSI, to remove the static data from ships that were sailing in other areas than the Westerschelde. The resulting table only contains distinct MMSI numbers and the values for bow, stern, port and starboard. The length and beam are calculated by setting length as bow + stern and beam as port + starboard. This is also explained in section 2.2.2.

To get better results, some data is removed from the dataset. Points including a length or beam with value 0 are removed, because a ship cannot have no length or beam. To make it possible to join the table based on different MMSI numbers, the table is checked for MMSI numbers that were present more than one time. Some ships had different values for bow, stern, port and starboard, but the values for length and beam were the same. Other ships did have different values for length and beam for one MMSI number. To deal with this problem, for each MMSI, the minimum value is chosen. This is done, because small ships can sail everywhere, but large ships cannot. When visualizing the clusters found, choosing the minimum value will lead to more reliable results than choosing the maximum value.

Pattern 2

After clustering of the first pattern, the data is joined to the dynamic data, to know the exact location of the clusters. The resulting file is used as input for the second and third pattern. The second pattern is based on the same values as the first pattern, but the variable 'speed over ground (SOG)' is added. For this pattern, only data from the Westerschelde is used, the ports are excluded, because of some strange values. For this pattern, the data for SOG is checked by calculating the speed using time and distance. This is done using PostgreSQL and ArcGIS Pro. Postgres is used to find the next point for each point (Baars, 2004). This results in a table like Table 4.2.

Table 4.2: Example next point

ID Point	Timestep Point	ID Next Point	Timestep Next Point
1	00.00.01	2	00.00.21
2	00.00.21	3	00.00.41
3	00.00.41	4	00.01.01

Based on this table, the previous point can be found too. This is done in ArcGIS Pro by adding the table twice and joins. For each point, the previous and next point were found and the distance between the point and the previous point is calculated using the XY to line tool. This is also done for the distance

between the point and the next point. Based on the distance between the previous and next point and the time difference between these points, the speed is calculated in km/hour. This is converted to knots, because this unit is used in AIS data for SOG. In Postgres this speed is compared with the SOG as available in the AIS data. Points were speed was NULL are removed. Ships with a SOG of 102.3 were also removed, because this means that the speed was not available. A SOG higher than 31 knots is not reliable, so for all points with a speed higher than 31 knots and a difference lower than 5 knots between the calculated speed and the SOG from the AIS data, the SOG from the AIS data is used. For the other points with a SOG less than 31 knots, the calculated speed is used, if that value was lower than 31 knots. For some ships, the registered SOG was 0, but the actual speed was between 20 and 30. For these ships, the speed difference value is used. This is done for ships like pilot vessels.

Pattern 3

For the third pattern, the distance between each point and the fairway is calculated. This is done for the same area as the second pattern, because this pattern is related to the distance to the fairways and the speed in the Westerschelde area, not the ports. To calculate the distance between the points and the fairways, the near tool in ArcGIS Pro is used.

Normalization

The data used has different ranges for the attributes. For example, the range of beam values is 3 meters – 59 meters, the length values are between 11 meters and 399 meters. Using these values might result in wrong clustering results. To overcome this problem, the data is normalized. Data normalization results in comparable value. Different normalization methods exist, like Z-score normalization and min-max normalization. Z-scores are used for normal distributed values. To calculate these scores, the values are standardized based on the mean and standard deviation of the dataset (Bin Mohamad & Usman, 2013). For non-normal distributed datasets, min-maxscaling can be used. In this normalization method, the minimum value for the attribute is subtracted from the value and divided by the range of values of the attribute. This is shown in the formula:

$$minmax\ value = \frac{X_v - X_{min}}{X_{max} - X_{min}}$$

In this formula, X is the attribute (column) in the dataset. X_{min} is the minimum value in this column. X_{max} is the maximum value of this column. This results in values between 0 and 1 (Bin Mohamad & Usman, 2013; Gopal, Patro, & Kumar Sahu, 2015). To check the distribution of the dataset for the variables, the skewness and kurtosis can be calculated. Skewness is a measure of asymmetry and results in a positive or negative value. Negative values show a left skewed distribution of data, a positive value means a right skewed distribution. A perfect normal distribution has a value of 0 (Wang, Smith, & Hyndman, 2006). Values between -0.5 and 0.5 are used to indicate a normal distribution dataset, values between -1 and 1 indicate a relatively normal distribution, higher or lower values indicate that the distribution is not normal (Van der Zee, 2015). Another measure is kurtosis. This measure shows whether the dataset has a peak or flat distribution. A normal distribution has a value of 0, the minimum value is -3, which means that the data has a flat distribution. No maximum value exists (Wang et al., 2006). These values are calculated for the data used in this research. This is done using the scipy.stats module in Python.

The data used for the first pattern does not have a normal distribution. To normalize the dataset, the min-maxscaling method, as explained before, can be used for clustering. This method is also used in this research, for all variables in all patterns (Rhodes, Cole, Upshaw, Edgar, & Webber, 2014). For this step, PostgreSQL is used.

4.4.2. Clustering

In the previous chapter, different clustering methods are explained. For all patterns in this research, two different clustering methods are used: one partitioning method and one density based method. The algorithms chosen are K-means and DBSCAN. These algorithms are used and the results are compared. To determine the best value for K in K-means, the Elbow method and the silhouette index are used (as discussed in 3.2). The Elbow method plots the distortion per different value for K. The distortion is defined as the average of squared distances from the cluster centers of each cluster. The best amount of clusters is the value after which the distortion has a linear decrease (Bholowalia & Kumar, 2014). The silhouette index indicates the compactness and separation of clusters per possible K (Rendón et al., 2011). The resulting average scores are between -1 and 1, with 1 indicating a well clustered result (Scikit-Learn, 2020b). For K-means, DBSCAN and the methods to determine the best K, the python modules sklearn.metrics.silhoutte_score, sklearn.cluster.KMeans and sklearn.cluster.DBSCAN are used.

4.4.3. Validation

To validate the clustering result, two methods of validation can be used: internal, external and relative clustering validation. Internal validation is related to validation using information intrinsic in the dataset. External validation uses previous knowledge. Relative clustering validation means comparing the result of one clustering method with other clustering methods (Rendón et al., 2011).

In this research internal validation is done using the Calinsky-Harabasz index and the Davies-Bouldin index (Rendón et al., 2011). The Calinsky-Harabasz index is a value of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters. Dispersion is the sum of squared distances. The distances used are the distances from a point to the cluster center. The best score for this index is the highest score (Scikit-Learn, 2020a). The formula to calculate this score is:

$$\frac{SS_B}{SS_W} \times \frac{N-k}{k-1}$$

In this formula, SS_B is the overall between-cluster variance, SS_W is overall within-cluster variance, N is the total number of points and k is the number of clusters.

The Davies-Bouldin score is related to the separation between clusters. It results in a value between 0 and 1, with a value closer to 0 indicates a better cluster separation (Scikit-Learn, 2020a). The value given is based on the similarity between a cluster and the cluster that looks like the most to that cluster (Rendón et al., 2011). The formula to calculate this score is:

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i$$

In this formula, n_c is the number of clusters, R_i a measure of how good the clustering is (Choudhari, n.d.).

For the patterns in this research, no previous knowledge is available to compare the clustering results with existing labels for external validation. However, in this research, a kind of external validation is applied by dividing the dataset in two randomly generated groups, one containing 30% of the data and one containing 70% of the data. First, the entire dataset (7 days of AIS data) is used for clustering. After that, the 30% part is clustered again and the results of that cluster analysis are compared with the values from the first clustering. This is done using the Adjusted Rand Index (ARI). This index compares the labels of the clustering with the predicted labels, in this case the labels retrieved from the first clustering. Perfect labelling has a score of 1, bad labelling will result in negative values or values close to 0 (Scikit-Learn, 2020a).

The DBSCAN clustering result is validated based on the Calinsky-Harabasz, Davies-Bouldin and Silhouette index. However, these indices are not very useful for DBSCAN, because they are developed for the validation of spherical clusters. Another option was to validate using the Density-Based Clustering Validation (DBCV) method, which is developed for clusters based on density (Moulavi, Jaskowiak, Campello, Zimek, & Sander, 2014). It is possible to use this method in Python. However, it resulted in some technical problems, so it is not used by the researcher.

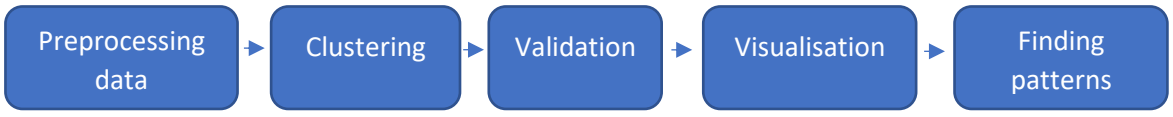
4.4.4. Visualisation

The clustering results found are visualized for each pattern using ArcGIS Pro. For the first pattern, first, the results of the clustering are joined with the dynamic data to get the coordinates. In the other patterns, the coordinates were included in the datasets used. For the first pattern, 3 visualisation method are compared: showing points in different colours, spatial DBSCAN and the aggregate points tool. The best option is selected for the other visualisations. This was the aggregate points tool. First, for each clustering result, K-Means with different K's and DBSCAN, the dataset is divided per cluster, by using the split by attribute tool. This resulted in a layer per cluster. For each layer, the aggregate points tool is used to create polygons based on the points. For this tool, the aggregation distance is set to 600 meters, which is chosen by visual inspection of the points (ESRI, 2020).

4.5. Research design

For each of the patterns, the same steps will be carried out. This is visualised in Figure 4.4.

Figure 4.4: Research design scheme



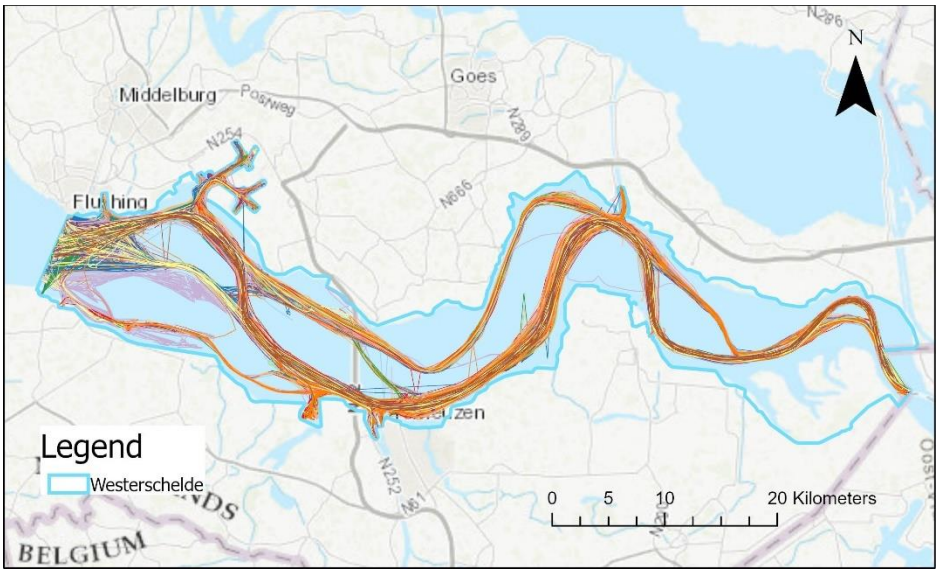
5. Results

This chapter contains the results from the method as explained in the previous chapter. The results will be described per pattern. First, the dataset used will be described in more detail.

5.1. Description of data

The dataset received contains data within a square area between the coordinates : 0.0, 50.0 and 6.0, 53.0. The static part of the data contains 699,424 points for seven days, with 6913 different ships. The dynamic data is clipped for the Westerschelde and contains 672,060 points. The tracks per MMSI are shown in Figure 5.1. In this map, no legend is added for the tracks, because of the high number of tracks.

Figure 5.1: Tracks per ship in Westerschelde



5.2. Pattern 1: Length and beam

5.2.1. Pre-processing

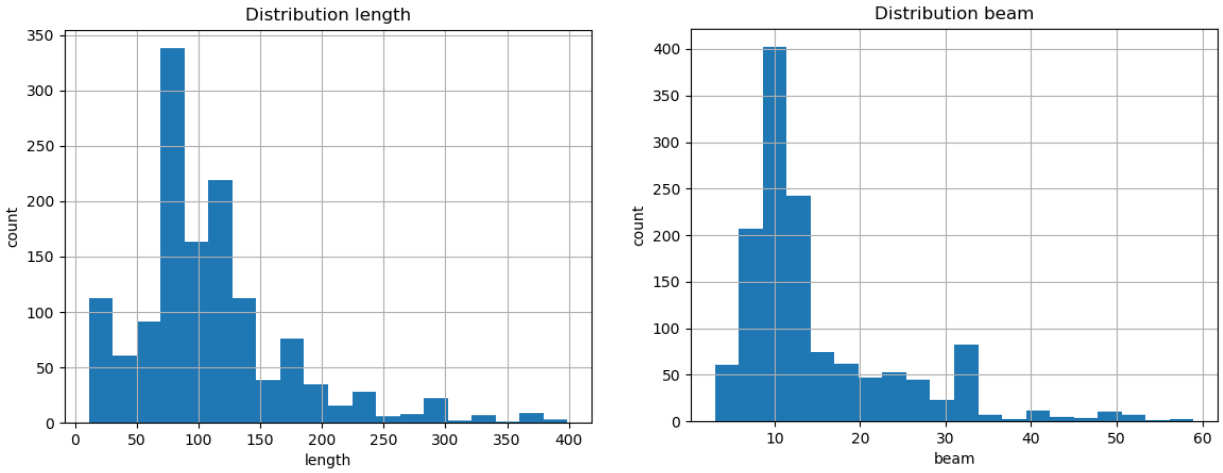
The first pattern to detect in this research is the pattern based on length and beam. During the pre-processing part, first, the data is checked for length and beam. Some MMSI number did have different values for length and beam for different points. Some MMSI number were related to different ships, like MMSI 244645890, which is related to the ships 'Delivery', 'KVB-Nomadisch1/2' and 'Maasstroom 11'. Within the data, it is not possible to check which point belongs to which ship, because the only identifier in the dynamic data is the MMSI. Some other MMSI's did not belong to different ships, but did have different values for length and beam, for example MMSI 244010770, which belongs to a ship called 'Teunis', a pusher boat. This shiptype has a length that changes according to the transported cargo (shown in Figure 5.2 and Figure 5.3). After the pre-processing was done for this pattern, 1350 ships were included.

Figure 5.2: Ship 'Teunis' without push barges (C.R.J.A. Stravers)

Figure 5.3: Ship 'Teunis' with push barges (P. Westeel)

For both variables, the skewness and kurtosis are calculated. The skewness for length is: 1.48, for beam: 1.68. The kurtosis for length is: 3.17, for beam: 2.74. This means that the data has a right skewed distribution for both values. This is also visible in Figure 5.4. The data is not normally distributed, which means that the min-maxscaling method is applied.

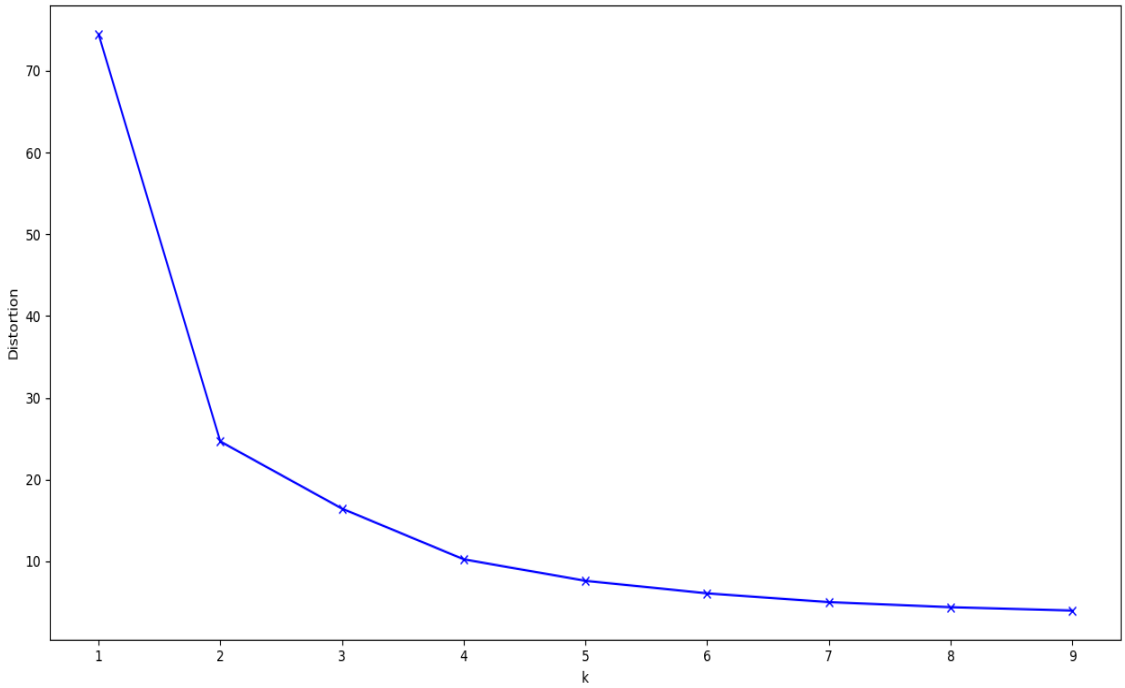
Figure 5.4: Histograms length and beam



5.2.2. Clustering

As clustering algorithms, K-Means and DBSCAN are applied. First, the results of the K-Means clustering will be discussed. Before applying this algorithm, the best number of clusters (K) needs to be detected. This is done using the Elbow method and the silhouette score. The result of the Elbow method can be found in Figure 5.5. In this figure, it is clear that after K = 5, the line has a linear decrease, which means that the best number of K is 5 based on this method. The corresponding scores can be found in Table 5.2.

Figure 5.5: Elbow method showing the optimal K for pattern 1



The second method to detect the best number of clusters is the silhouette score. The results can be found in Table 5.2. Based on these results, the best number of clusters is 2, because a value closer to 1 will give better clustering results.

Table 5.1: Elbow method pattern 1

Number of clusters	Distortion
1	74.48
2	24.65
3	16.43
4	10.23
5	7.60
6	6.06
7	4.99
8	4.36
9	3.94

Table 5.2: Silhouette score pattern 1

Number of clusters	Average silhouette score
2	0.678
3	0.631
4	0.491
5	0.479
6	0.440
7	0.440
8	0.454
9	0.447

Both values (2 and 5) are used as input for the K-Means clustering in Python. The scatterplot of both results can be found in Figure 5.6 and Figure 5.7. In these figures, the black dots are the centroids. The labels given to the clusters are based on the location of the clusters within the figures.

Figure 5.6: Result K-Means clustering with K=2, pattern 1

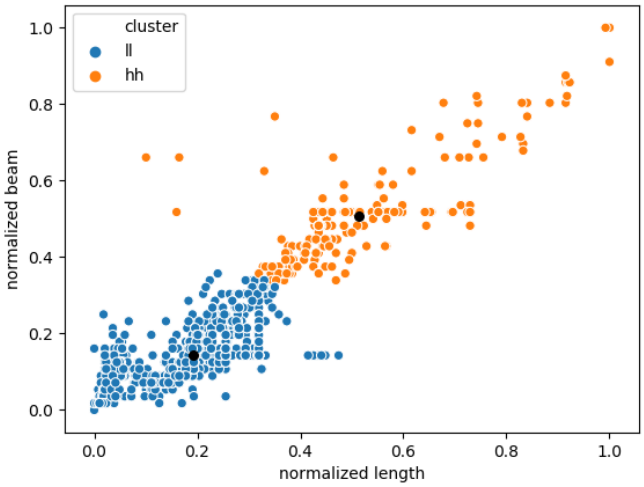
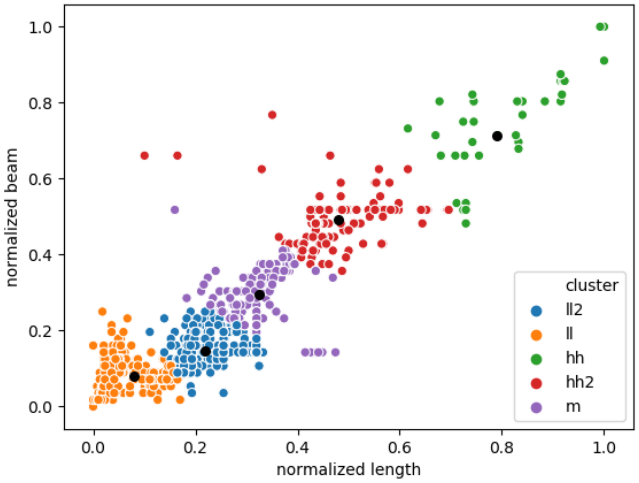
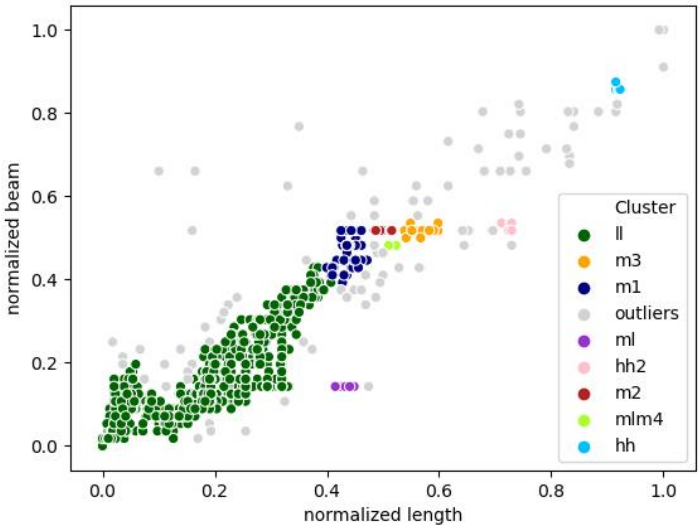


Figure 5.7: Result K-Means clustering with K=5, pattern 1



The DBSCAN algorithm is also applied for this pattern. The results of this algorithm can be found in Figure 5.8. This algorithm resulted in 8 clusters and 100 outliers. Compared to the K-Means clustering, the results of the DBSCAN have more clusters, but these are not equal in size. The largest group of the DBSCAN algorithm is the lower-left group which is a bit the same as the lower left group in the K-Means clustering with K=2. Most groups in the DBSCAN clustering are very small, containing only a few points.

Figure 5.8: Results DBSCAN pattern 1



5.2.3. Validation

To validate the clustering result, internal and external validation is applied. For internal validation, the Calinsky-Harabasz and the Davies-Bouldin index used. The Calinsky-Harabasz index is a value of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters. Dispersion is the sum of squared distances. The distances used are the distances from a point to a cluster center. The best score for this index is the highest score. The Davies-Bouldin score is used to detect the separation between clusters. For this index, the values are within a range between 0 and 1, with 0 as the best separated clusters. The scores for these indices are calculated for 2 up to 6 clusters and can be found in Table 5.3.

Table 5.3: Validation indices pattern 1

Number of clusters	Calinsky-Harabasz index	Davies-Bouldin index
2	2725.25	0.53
3	2379.46	0.58
4	2819.29	0.68
5	2949.05	0.66
6	3033.89	0.69

Based on the Calinsky-Harabasz index, the best amount of clusters was 6, because that score is the highest one. However, 2 and 5 are chosen. Based on the values in table 5.3, it is clear that K=3 would have been the worst choice. Based on the score for the Davies-Bouldin index, the best amount of clusters was 2, because this value is the closest to 0.

The Adjusted Rand Index score for the K-Means clustering result was 0.99 for clustering with K=2, which is a good score, because the best score for this index is 1. For K=5 the score was: 0.51. This means that the clustering performance for K=2 was better than for K=5.

For the DBSCAN algorithm, validation is done using the Calinsky-Harabasz, Davies-Bouldin and silhouette index. As explained in the methodology, these indices are not very suitable for DBSCAN clustering. The results for validation of this method are: Calinsky-Harabasz: 279.98, Davies-Bouldin: 6.77 and silhouette index: -0.59. Because there is only one amount of clusters for the DBSCAN method, it is hard to say something about these values. Compared to the Calinsky-Harabasz values in Table 5.3, the result is much lower. The result of the Davies-Bouldin index is strange, because a normal result is between 0 and 1, so this index cannot be used for validation. The silhouette index is low, because a value close to 1 shows a good result.

5.2.4. Visualisation

The clusters found are visualised in a map. First, different visualisation methods are compared. In Figure 5.10 and 5.11, three methods are shown. In Figure 5.10, all points are visible with a colour corresponding to the cluster they belong to with K = 5. This resulted in a large dataset. In Figure 5.11, for one cluster (ll), the DBSCAN method is applied in ArcGIS Pro. This means a spatial clustering. This also resulted in a large dataset, but some outliers are visible. The third method is also shown in Figure 5.11. This is the aggregate by points tool in ArcGIS Pro. This tool is used for the other patterns too, because it resulted in one or more polygons per cluster.

Figure 5.10: Results K = 5, points with different colors

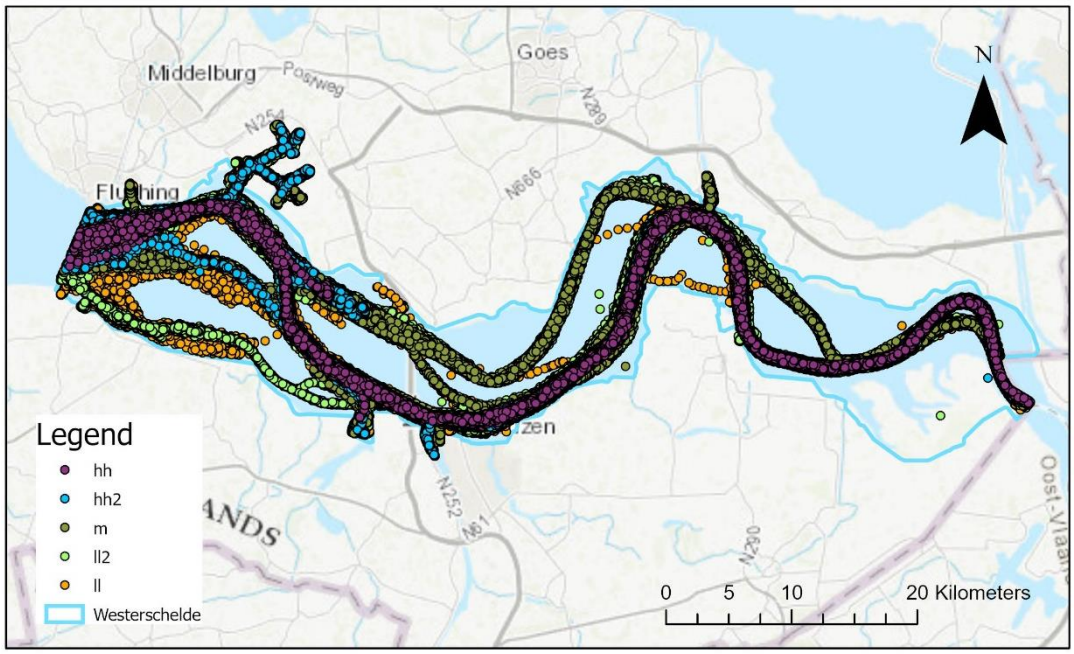
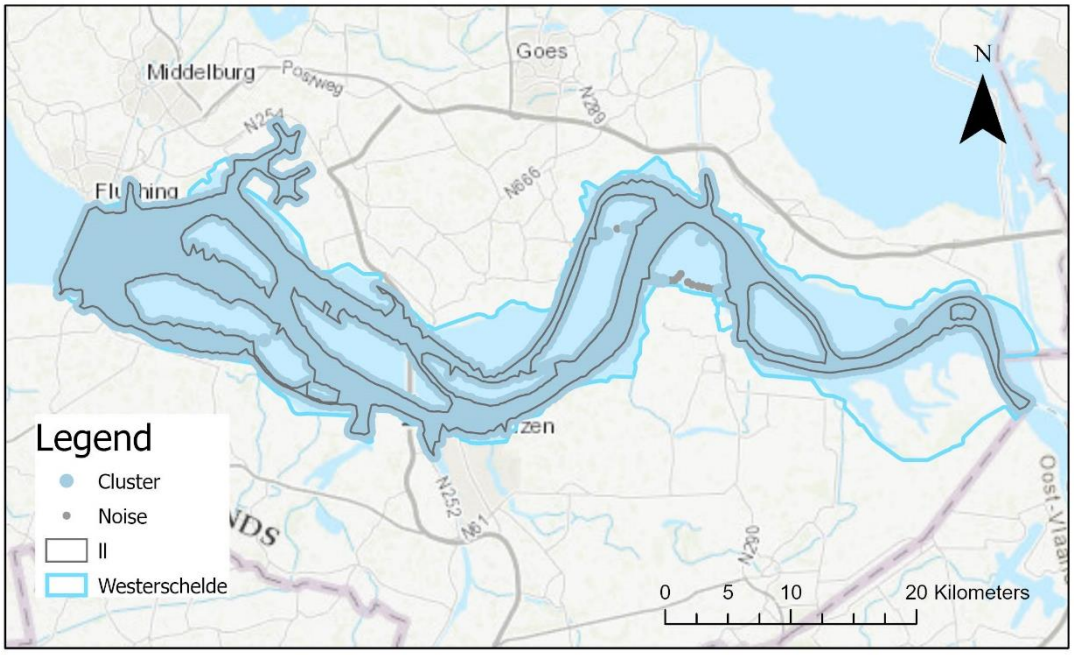


Figure 5.11: Results cluster ll, DBSCAN in visualisation compared to aggregate by points



The results for all values of K can be found in Figure 5.12 – Figure 5.13. For the clusters based on K-Means, clear tracks are visible. The labels given and shown in the legend are based on the location of the cluster in Figure 5.7. Especially the track for ships with the highest length and beam is clear. It is also clear that not much difference exist in the tracks of hh (high length, high beam) and hh2 (second highest length and beam) in Figure 5.13. These groups are one cluster in 5.12.

Figure 5.12: Results K = 2 pattern 1

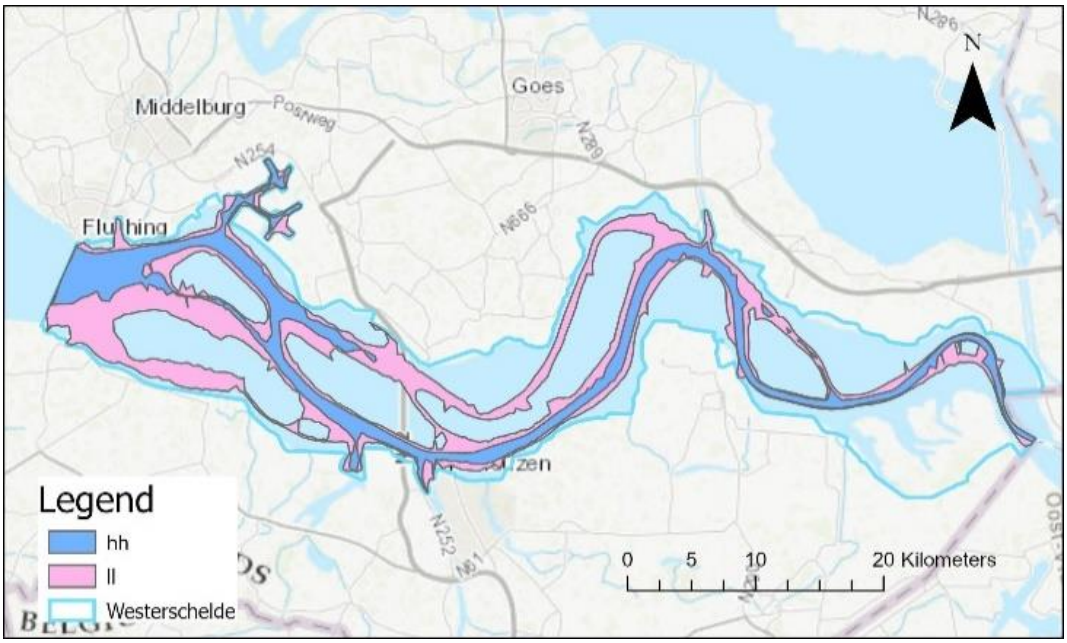
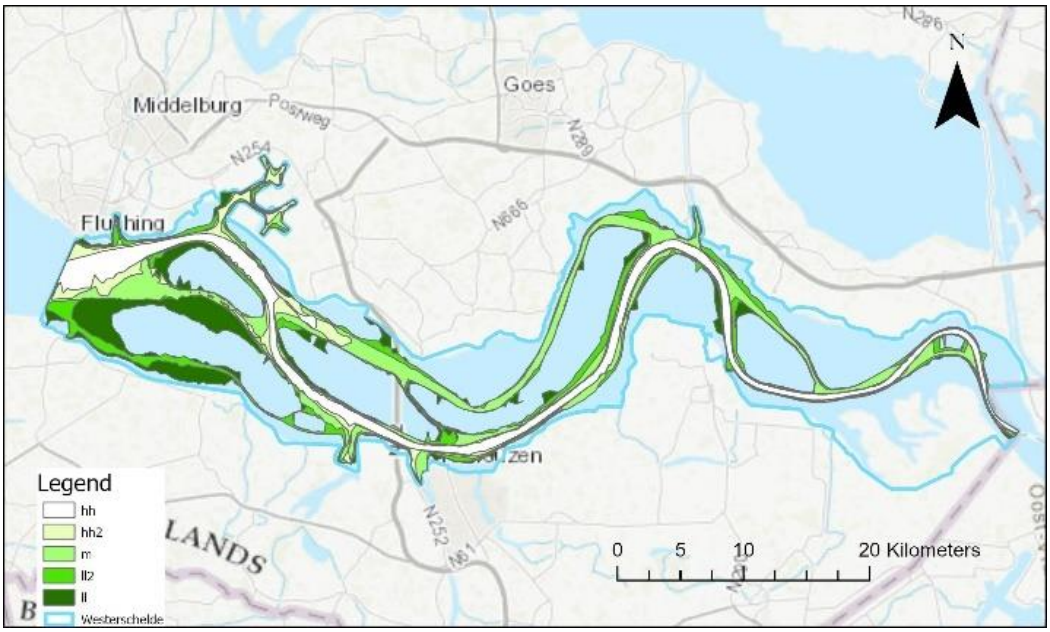
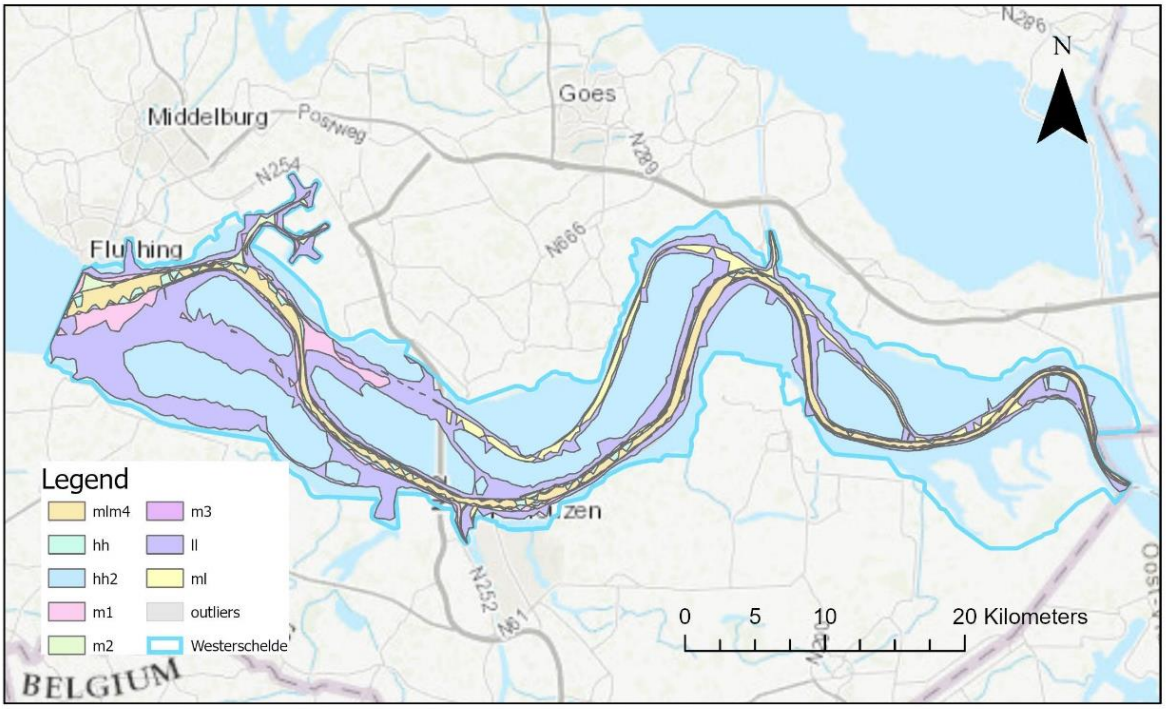


Figure 5.13: Results K = 5



Based on the DBSCAN algorithm it is hard to see differences between the different tracks (Figure 14). Only the ll cluster, the largest cluster, has a specific own track: these ships sail everywhere. The other clusters can be found especially on the same track as the hh cluster in Figure 5.12.

Figure 5.14: Results DBSCAN pattern 1



5.3. Pattern 2: Length, beam and SOG

5.3.1. Clustering

For the second pattern, K-Means and DBSCAN are both applied too. To get comparable units for the different variables, also for this pattern min-maxscaling is applied. The ports are excluded from the dataset, because it was the purpose to find the patterns for the Westerschelde. This also reduces the number of values with speed = 0. In this pattern, 268,038 points are included. The highest value for speed is 30.9 knots, the lowest value is 0. For this pattern, the results of the Elbow method to detect the best number of K can be found in Figure 5.15.

Figure 5.15: Elbow method pattern 2

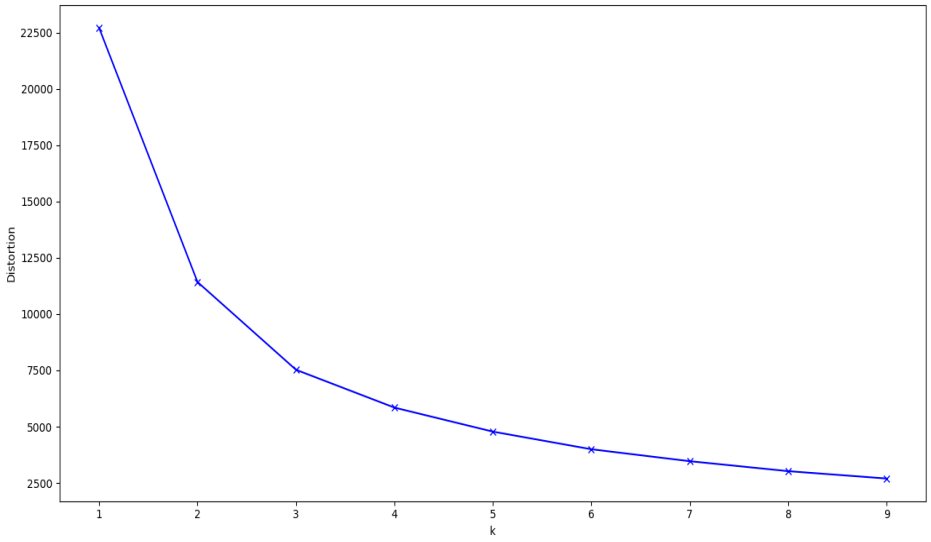


Table 5.4: Elbow method values pattern 2

Number of clusters	Distortion
1	22714.54
2	11424.46
3	7526.96
4	5847.18
5	4779.85
6	3997.24
7	3466.62
8	3024.37
9	2696.97

Based on this figure, for this pattern K=4 is chosen. The silhouette method is also applied to this pattern. The results can be found in Table 5.5. Based on these scores, the best number of clusters is 2. For this pattern, both K = 2 and K = 4 are used.

Table 5.5: Silhouette method values pattern 2

Number of clusters	Average silhouette score
2	0.541
3	0.450
4	0.447
5	0.446
6	0.357

The results of the K-Means clustering can be found in Figure 5.16 and 5.17. Comparing these figures, it is seems that the left cluster in Figure 5.16 is divided into two clusters in Figure 5.17, based on the SOG. The points in the left part of the figures with a high SOG might be related to tender ships or pilot vessels. An example of the speed and pattern for one pilot vessel is given in Figure 5.18. This ship has a length of 23 meters and a beam of 6 meters. The maximum speed for this ship in the data used is 29.4 knots, as is shown in Figure 5.18.

Figure 5.16: Result K-Means clustering K = 2 for pattern 2

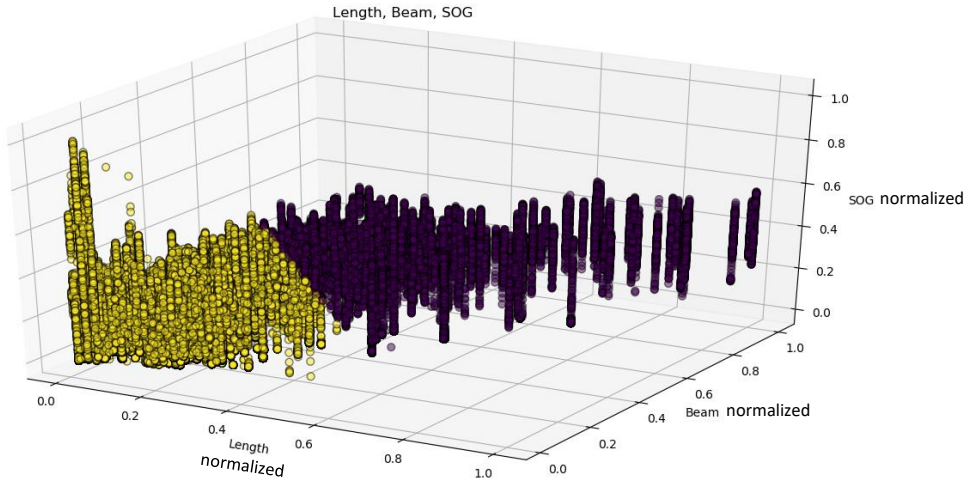


Figure 5.17: Result K-Means clustering K = 4 for pattern 2

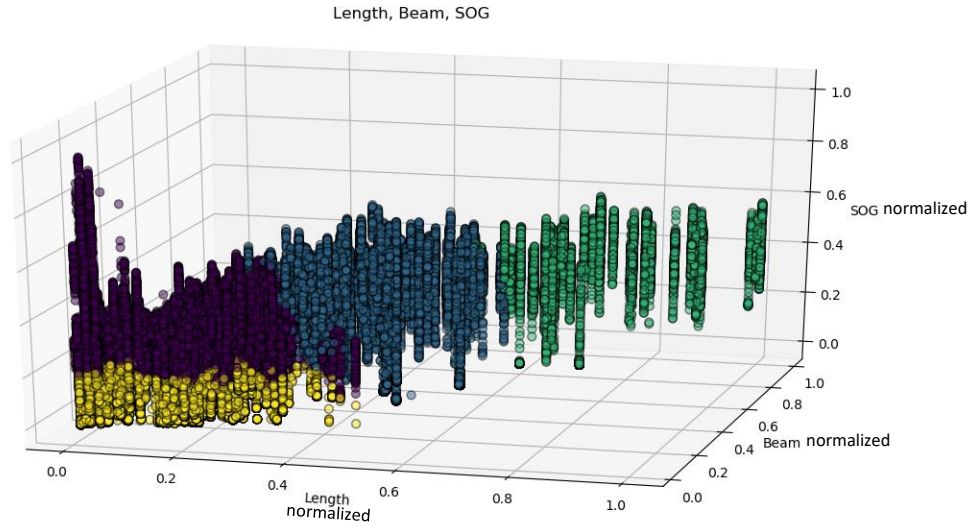
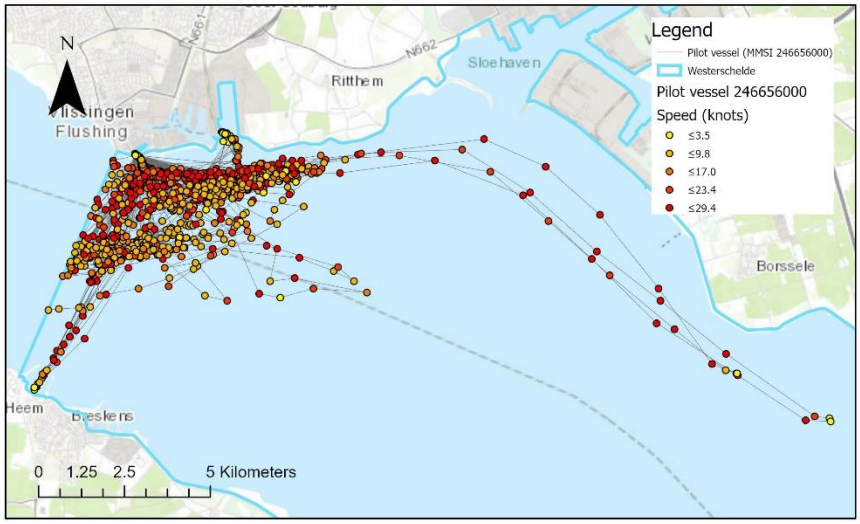


Figure 5.18: Speed and pattern of a pilot vessel (MMSI 246656000)



The DBSCAN algorithm is also applied for this pattern. This resulted in 44 clusters and 58 outliers. The patterns for this algorithm will be discussed in the next sub-section.

5.3.2. Validation

The clustering results are internally validated using the Calinsky-Harabasz index and the Davies-Bouldin index. An explanation of these indices can be found in 4.4.3. The values for these indices are shown in Table 5.6. For the Calinsky-Harabasz index, the best score is the highest score. The highest score is 270414.87 for K = 3. This is interesting, because based on the Elbow method and the silhouette index, the optimal number of clusters was 2 or 4. For the Davies-Bouldin index, all scores are relatively high, because the scores are within a range between 0 and 1. The best scores are 0.78 (K = 2) and 0.79 (K = 5). Based on these values, the only match with the used K-values is K = 2.

Table 5.6: Validation indices pattern 2

Number of clusters	Calinsky-Harabasz index	Davies-Bouldin index
2	264883.47	0.78
3	270414.87	0.83
4	257732.94	0.85
5	251454.98	0.79
6	251021.49	0.94

The score for the Adjusted Rand Index (ARI) is 0.996 for K = 2 and 0.985 for K = 4. These are both high scores, because the highest score for ARI is 1. This means that based on this index K = 2 performs better. For DBSCAN the score for the Calinsky-Harabasz index is: : 4133.45, which is much lower than the Calinsky-Harabasz index values for K-Means. The value for the Davies-Bouldin index for DBSCAN is 1.80, which is an unexpected value because the values should be between 0 and 1. The silhouette-index resulted in a value of -0.26, which incidates a not well clustered result.

5.3.3. Visualisation

The results of the visualisation of the clusters found are shown in Figure 5.19 and 5.20. Based on Figure 5.19, it is clear that for this pattern, where SOG is added as variable compared to pattern 1, the results do not differ much when comparing Figure 5.19 and Figure 5.12. This is the same for Figure 5.20 and Figure 5.13. In Figure 5.18 and 5.19, the visualisation for the DBSCAN results are shown. Because 5.22 has many classes (44), in Figure 5.21 the largest class is shown. Compared to the visualisation of pattern 1, it is expected that this class is based on a low length and a low beam, because this cluster can be found in the whole area, not just one track.

Figure 5.19: Visualisation pattern 2 with K = 2

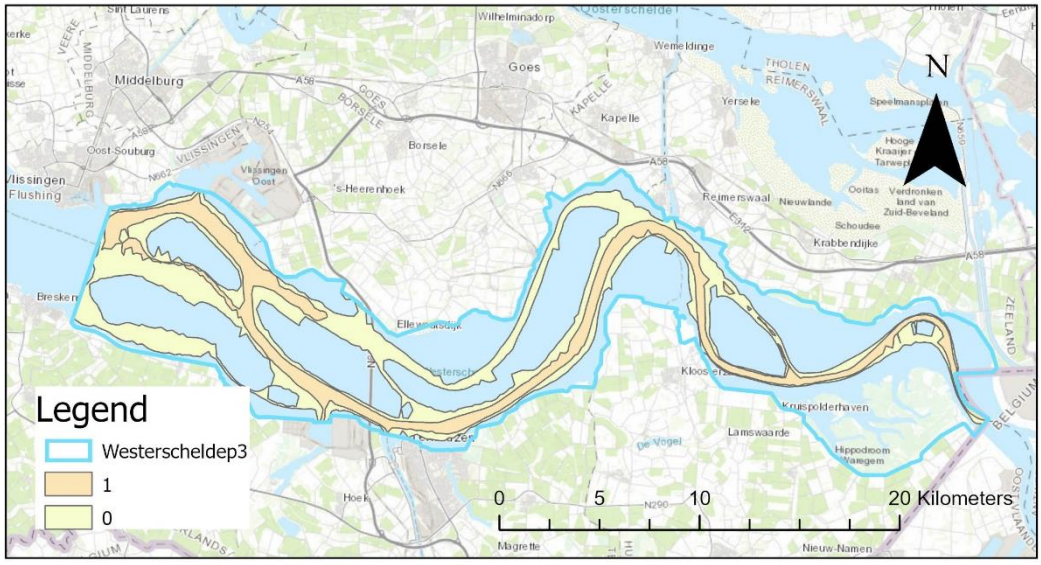


Figure 5.20: Visualisation pattern 2. K = 4

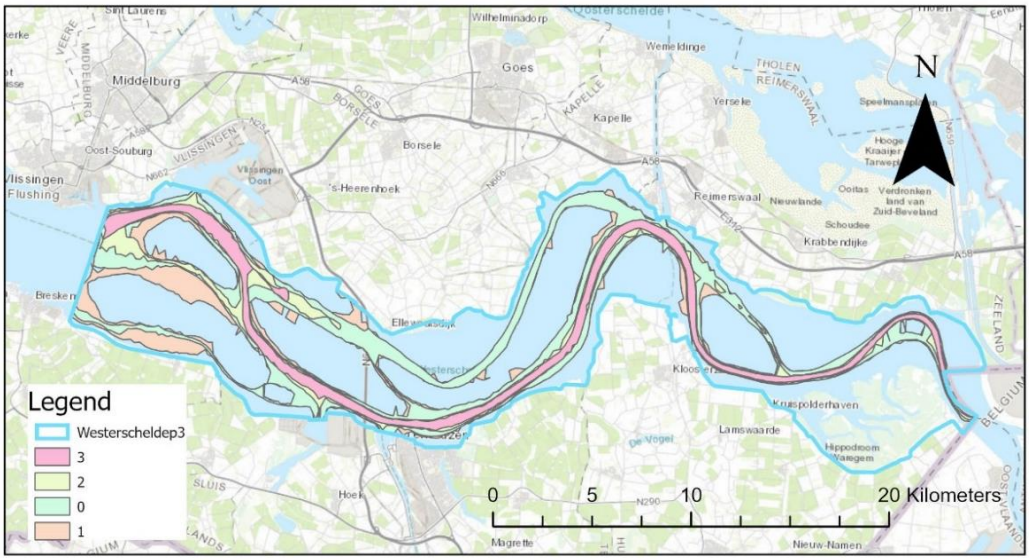
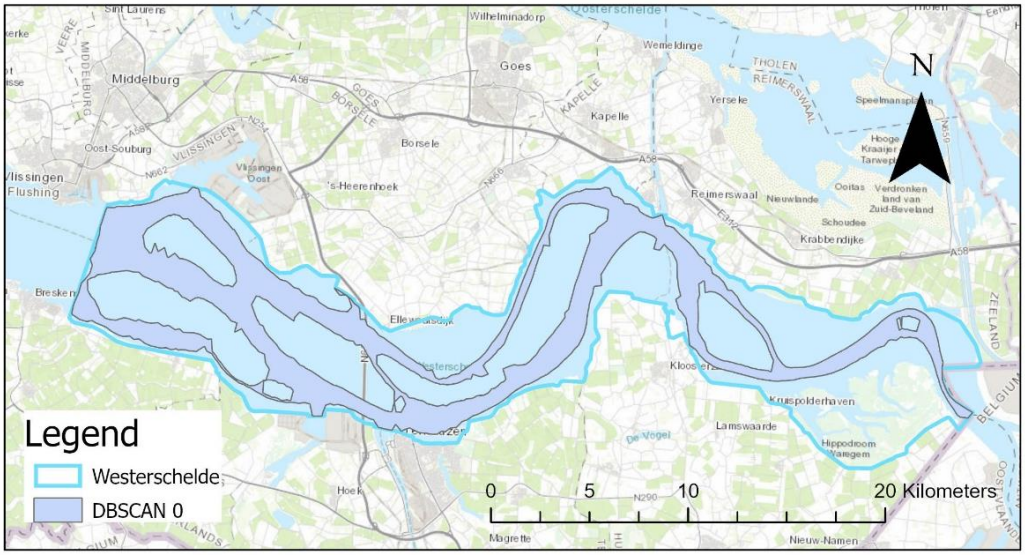
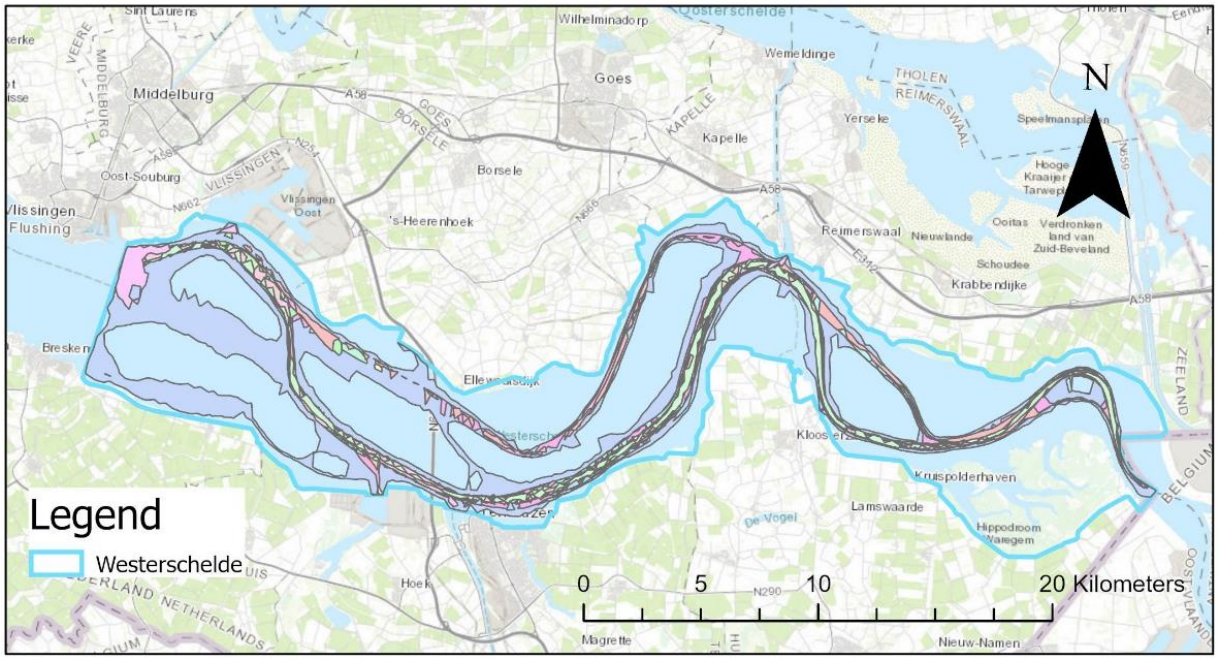


Figure 5.21: Visualisation pattern 2 DBSCAN cluster 0



5.22: Visualisation pattern 2 DBSCAN all clusters

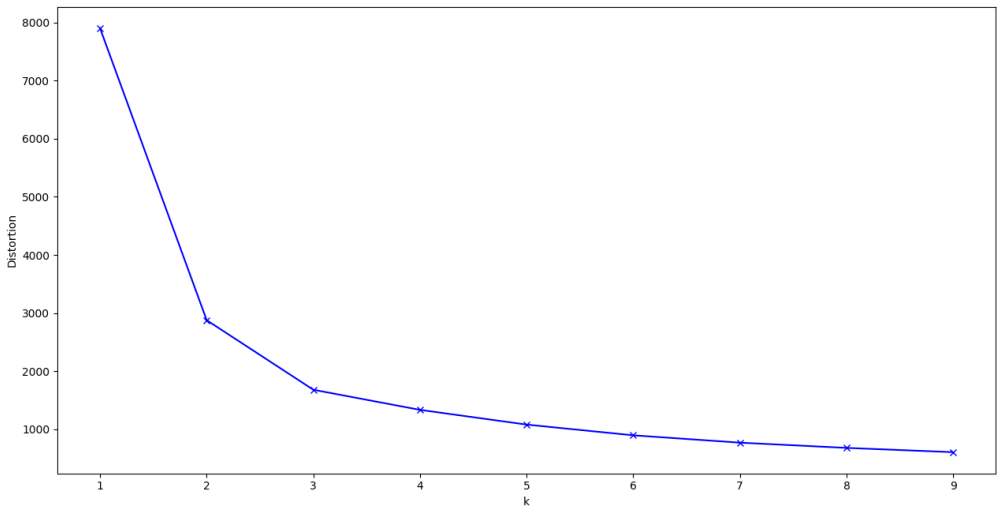


5.4. Pattern 3: SOG and distance to fairway

5.4.1. Clustering

For this pattern, the values or distance to the fairway are also normalized using the min-maxscaling method. The Elbow method is applied to find the best number of clusters for K-Means. The result can be found in Figure 5.23. In this figure, the line has a linear decrease at K = 5.

Figure 5.23: Elbow method pattern 3



The result of the silhouette index can be found in Table 5.7. The best number of clusters based on this score is 3, so the values chosen for K for this pattern are 3 and 5.

Table 5.7: Silhouette method pattern 3

Number of clusters	Average silhouette score
2	0.483
3	0.488
4	0.432
5	0.450
6	0.433
7	0.399

Figure 5.20 shows the results of K-Means with K = 3. As shown in this scatterplot, no real groups within speed exist. The biggest difference between Figure 5.24 and 5.25 is that one cluster is found more right in the plot. In Figure 5.24 and 5.25, the black dots are the centroids. For DBSCAN, the number of clusters is 11, with 204 outliers. One large cluster is found, as is shown in Figure 5.26, and some small clusters are identified too.

Figure 5.24: Result K-Means K = 3, pattern 3

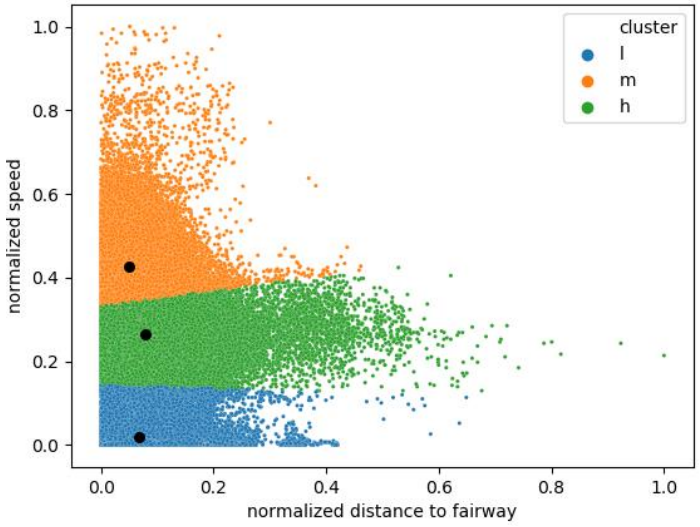


Figure 5.25: Result K-Means K = 5, pattern 3

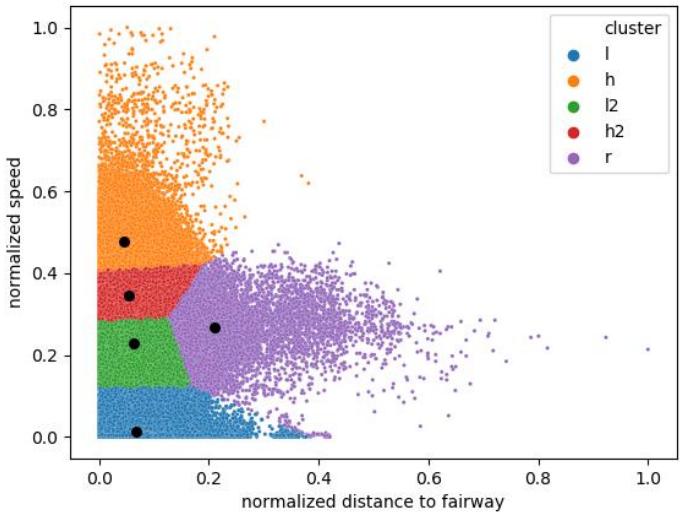
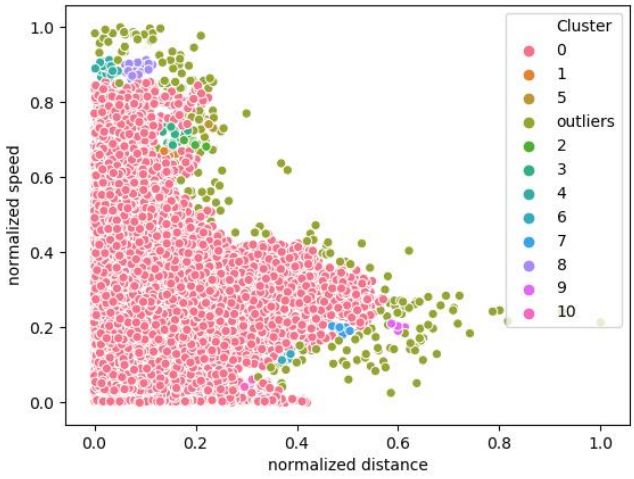


Figure 5.26: Results DBSCAN, pattern 3



5.4.2. Validation

For the validation of the third pattern, the same indices are used as for the other patterns. The results can be found in Table 5.8. For the Calinsky-Harabasz score, the highest score is the best one. This means that K = 3 is the best amount of clusters. This value for K is also used in the clustering method. K = 5 is used too, however, this number of clusters has a worse score for the Calinsky-Harabasz index compared to K = 3. The best score for the Davies-Bouldin index is K = 2, because this is the lowest value.

Table 5.8: Validation indices pattern 3

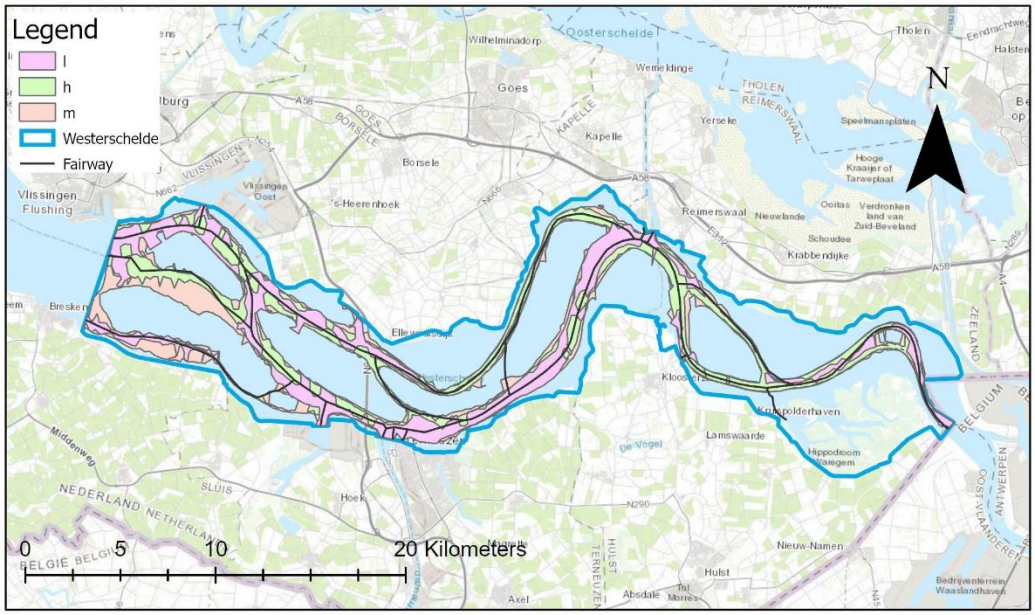
Number of clusters	Calinsky-Harabasz index	Davies-Bouldin index
2	467845.05	0.57
3	495736.92	0.73
4	439245.76	0.85
5	422351.30	0.78
6	418294.58	0.79

The ARI index results for this pattern were: for K = 3: 0.988 and for K = 5: 0.994. It is interesting to see that the score for K = 5 is higher than for K = 3, because K = 5 did not have good scores in the Calinsky-Harabasz and Davies-Bouldin index. For the DBSCAN clustering for this pattern, the Davies-Bouldin score is 1.14, which is a strange value, because it should be between 0 and 1. The Calinsky-Harabasz index for the DBSCAN clustering for this pattern is 192.73, which is very low compared to the values of the K-Means clustering. The score for the silhouette-index for DBSCAN is -0.29, which indicates that the result is not well clustered.

5.4.3. Visualisation

The spatial distribution for the result of the K-Means clustering for this pattern can be found in Figure 5.27 – 5.29. For K = 3, the m cluster is spread over the whole area, the h cluster only follows the two most northern tracks in the western part of the area. As was clear from Figure 5.27, the clusters for K = 3 are mainly based on the speed differences. This means that ships with a higher speed will take the most northern tracks in the western part of the research area.

Figure 5.27: Results K-Means with K = 3, pattern 3



To see the differences for the results of K-Means with K = 5, the results are shown in two maps (Figure 5.28 and Figure 5.29). Based on the spatial distribution in Figure 5.28, it is clear that the l2 cluster can be found in more tracks than the l cluster. This means that ships with a lower speed do not use for example the most northern track in the middle. Two groups that can be compared based on speed and distance are the l and the r cluster. The l cluster has a low speed and a low distance to the fairway. The r cluster has a higher speed and a higher distance to the fairway. The highest distance to the fairways can be found in the most western and the most eastern part of the research area, this are the points from the r cluster.

In Figure 5.29, the two groups h and h2 are compared. These groups have differences based on speed. This means that, based on this visualization, ships with a low speed have more tracks.

Figure 5.30 shows the results of the DBSCAN cluster. It is clear that only the largest cluster (cluster 0) has a clear track on the waterway. In Figure 5.31, the zoom on the western part of the research area is shown. In this part, some very small polygons are visible of other clusters.

Figure 5.28: Results K-Means with K = 5, only clusters r, l and l2, pattern 3

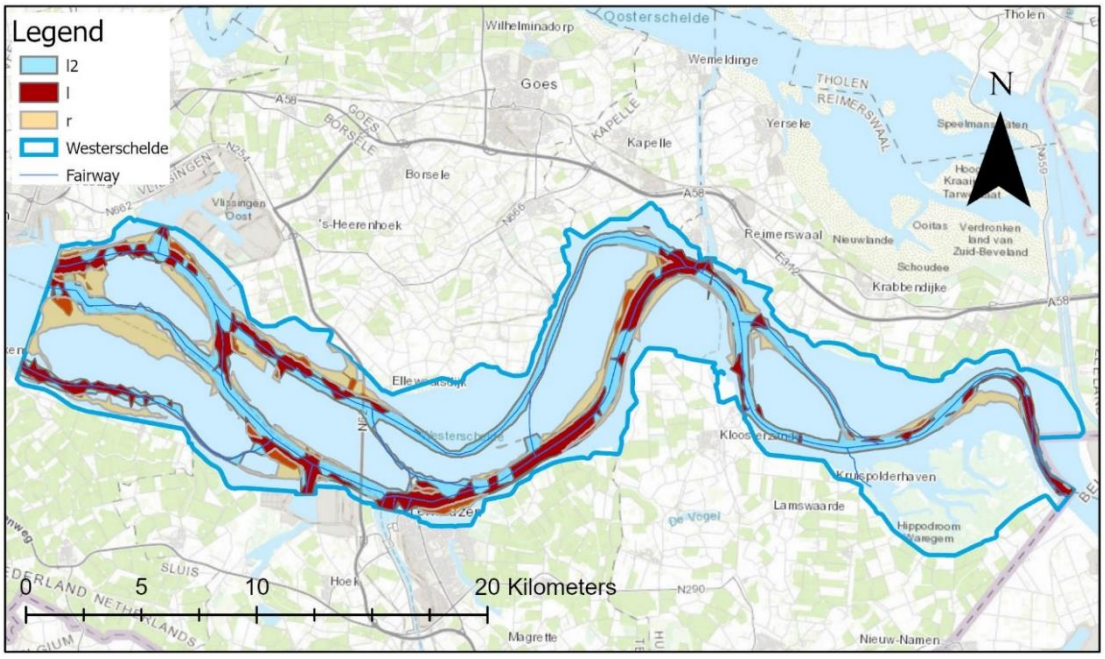


Figure 5.29: Results K-Means with K = 5, only clusters h2 and h, pattern 3

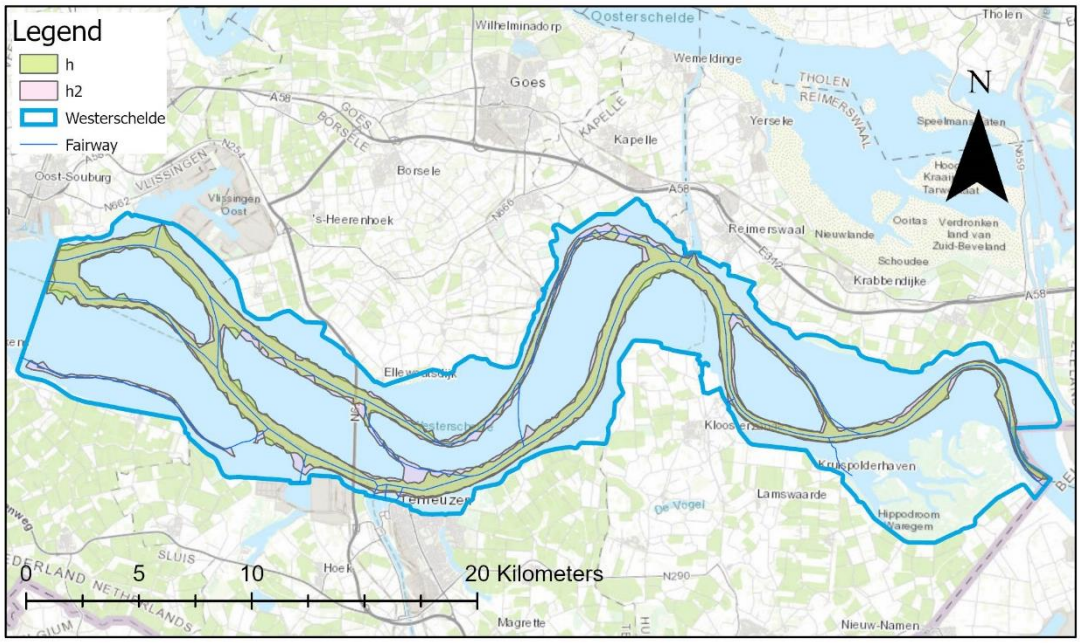


Figure 5.30: Results DBSCAN, pattern 3

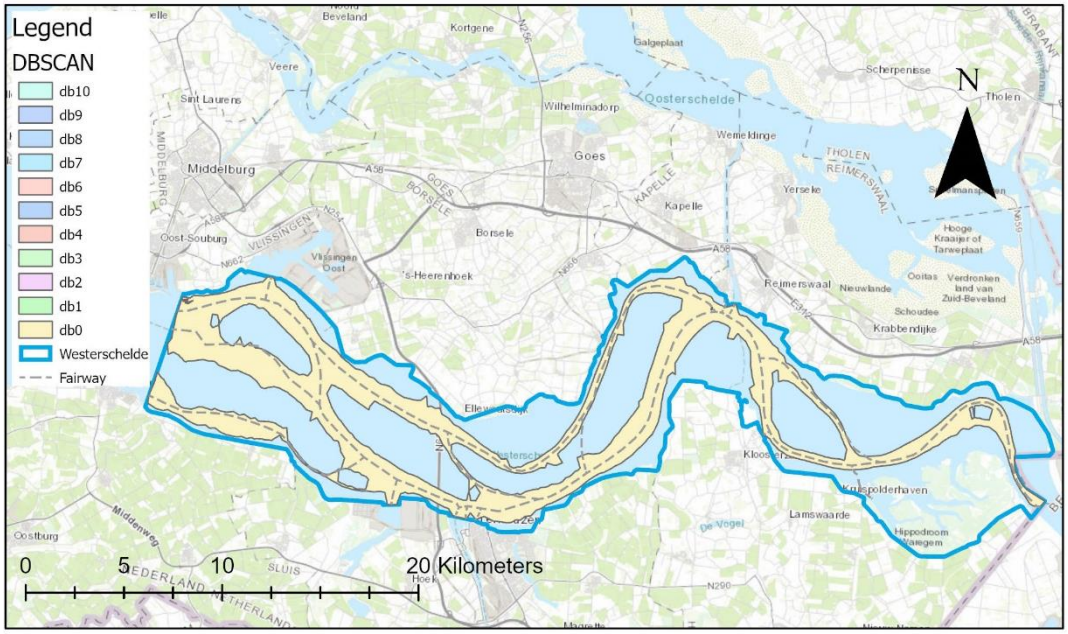
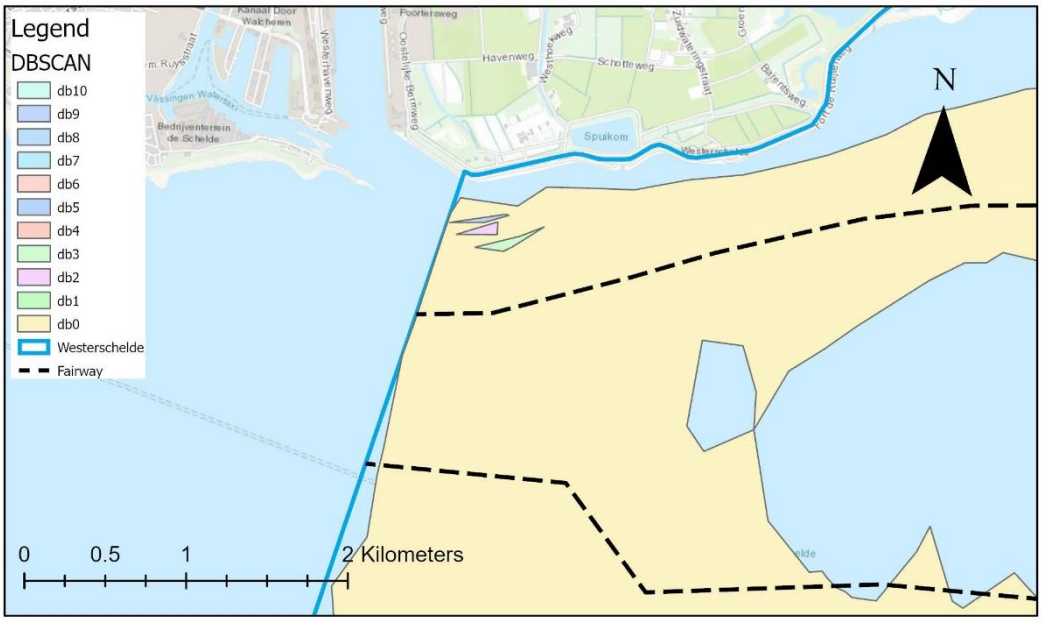


Figure 5.31: Results DBSCAN (zoom), pattern 3



5.5. Results summary

In table 5.9, a summary of the results for K-Means is given. This is done to have an overview of the chosen numbers of K and the results of the validation. In this table the scores for DBSCAN are not included, because no good validation method was found. The values for the Elbow method and the silhouette index are the values used for clustering. The ARI value is the number of K used which had the best performance. This table shows that for pattern 1 and 2, the silhouette index, the Davies-Bouldin index and the Adjusted Rand index give the best performance to the same amount of clusters. Only for the third pattern, the Elbow method and the ARI give the same K. Based on this table, it is also clear that most of these validation could have been done before clustering to get more information about the optimal K. Only the ARI needs information after clustering is done.

Table 5.9: Summary of results for K-Means

Pattern	Elbow method (used)	Silhouette index (used)	Calinsky-Harabasz index	Davies-Bouldin index	Adjusted Rand Index
1	5	2	6	2	2
2	4	2	3	2	2
3	5	3	3	2	5

6. Conclusion and discussion

6.1. Conclusion

In this research the use of clustering non-spatial variables to detect spatial patterns in AIS data is examined. This is done using the research question: *To what extent can clusters based on ship behaviour conditions in AIS data be used to detect spatial ship behaviour patterns?*

Before answering this research question, the sub-questions need to be answered first. This will be done per sub-question in this section.

SQ1. What are the behaviour patterns to detect and how to define them?

Answer: Based on literature, three patterns are found, which are tried to find using the clustering methods. The first pattern was based on the size of the ship, the second patterns was related to the speed and the size of the ship and the third pattern was related to the speed and the place on the waterway. These patterns were a combination of: 1. Length and beam, 2. Length, beam and speed over ground and 3. Speed over ground and distance to the fairway.

SQ2. Which conditions are relevant to find spatial ship behaviour patterns?

Answer: The first pattern is based on the values for bow, stern, port and starboard in AIS data. The second pattern is based on the same variables as the first pattern, but speed over ground, retrieved from the AIS data, but also calculated by the researcher, is added as third variable. The third pattern is based on speed, the same values as the second pattern, and the calculated distance to the fairway. For data about the fairways, other data than AIS is used.

SQ3. Which clustering algorithm is suitable for clustering AIS data and how to apply this algorithm?

Answer: For all patterns, different clustering algorithms are applied and different validation methods are compared. Based on the results, the best clustering algorithm was the K-Means algorithm. This can be applied using clustering modules for python.

SQ4. How to visualise the clusters found to find spatial behaviour patterns?

Answer: The best way to visualise the patterns found is using the aggregate points tool in ArcGIS Pro.

After applying the methodology for all patterns the conclusion to the research question is clear. For the first pattern, clear tracks were found for different clusters. For the second patterns also clear tracks were found, however, the clusters found were mainly based on length and beam and not on length, beam and SOG. For DBSCAN, many clusters were found for this pattern, which did not all have a clear location on the map. For the third pattern, different locations on the waterway for the different clusters could be found, but these clusters are mainly based on SOG and not on distance to the waterway. This means that clustering based on AIS data is possible and will result in different clusters and patterns, but it does not always add value.

6.2. Discussion

In this section, the results will be discussed in more detail and will be compared with literature. The research process will be discussed too.

Before clustering, for each pattern, the data is pre-processed to use the right data for clustering. Clustering is done based on the K-Means algorithm and DBSCAN. This resulted in different clusters and

spatial distributions. For the patterns chosen, the results from the K-Means algorithm seem to be the best. The results of DBSCAN do not always show clear patterns, because of the high amount of clusters (pattern 2) or some clusters are only located on a very small area, which will not result in a clear spatial distribution (pattern 3). When having such a large amount of clusters as the result of DBSCAN for pattern 2, it is not clear which cluster represents which values, for example: does this cluster represent a high or low length? The clusters are validated using some internal indices and one external index, which should be based on known labels. For the dataset, no labels exist, which means that this external index is based on data retrieved from previous clustering. This is no real external validation, but by using this method, the results for smaller and larger datasets are compared. For K-Means, some validation indices were used. For DBSCAN, these indices are used too, however the results of validation for DBSCAN are not reliable, because most measures are developed for spherical clusters (Moulavi et al., 2014). Based on the results of the validation, to find the best amount of clusters for K, the best method might be the Silhouette method. Compared to the results of the Davies-Bouldin Index and the ARI index, the amount of clusters retrieved from the silhouette method performs better.

Based on the clusters some spatial patterns are detected. For the first pattern, it is clear that ships with a high length and beam only follow one track in the Westerschelde. The lower the length and beam, the larger the area the ships navigate in in the Westerschelde. In Chapter 2.2, it was clear from literature that length and beam will result in different patterns on the waterway, according to Zhou et al. (2019). These authors did not cluster the data themselves and did not show the spatial patterns. Based on this research, it is clear that indeed different clusters can be detected based on differences in length and beam and different spatial patterns could be detected. For the second pattern, the variable SOG was added. This was based on the articles of Tu et al. (2018) and Lo Duca et al. (2017). These researchers explain that speed can be used for route estimation and predicting purposes. In this research the effect of speed on the clusters based on length and speed is examined. When comparing the results of the visualisation for the first pattern and the second pattern for K-Means, no clear differences are found. This means that, when adding speed as a third dimension, it will not add value. Tu et al. (2018) and Lo Duca et al. (2017) use speed in combination with course, which is not done in this research. This might cause the differences in results between this research and the two articles. For the third pattern, different patterns can be found for the different clusters. However, this is mainly caused by the differences in speed, as can be concluded from Figure 5.24 and 5.25. This means that this research does not agree with the conclusion of Zhou et al. (2019), who said that a relationship exists between speed and place on the waterway. However, they used a straight waterway and the Westerschelde has many curves, which might cause the difference.

As a conclusion to this research can be said that clustering based on AIS data is possible and spatial patterns can be found based on these clusters, but it does not always add value. For example for the last pattern, the same spatial patterns might have been detected by creating polygons only based on different classes for speed. Adding a third dimension did not add value to this research and when adding more dimensions, visualising the result in a scatterplot might be hard. Then, the meaning of a specific cluster is not clear: does it represent a high or low score et cetera. When using AIS data for clustering or other analysis tasks, it is important to be aware of the data quality (Harati-Mokhtari et al., 2007). Some variables might have many wrong values or values which mean that data is not available, but that are represented as a number, like 102.3 for speed. With such values, it is possible to do calculation, but a speed of 102.3 knots is impossible, so the user of the data needs to know the values for wrong data.

One of the aims of this research was to find patterns that might be used as input for ship behaviour prediction. Some patterns found, especially the first pattern, can be used as input for behaviour prediction, because it is clear where ships with a specific length and beam will sail. The exact track for ships in the cluster with the highest values is the most clear one, because only one track on the Westerschelde is found for this ship type. This means that this pattern can be used for prediction. It can also be used for navigation purposes, because for a captain of a ship in that category, the track to choose is clear. The other patterns do not result in clear spatial patterns based on clustering and will be less important for prediction. However, based on the last pattern, one small track is used only by ships with a medium or high speed. This is the track in the middle in the western part of the Westerschelde. When a ship is sailing there, it is not expected that the speed will decrease in such a way that the ship should belong to the cluster with the low speed values. However, to find this result, no clustering was needed.

6.3. Recommendations

Based on the research done and its limitations, recommendations can be given for future research on AIS data. First, based on this research can be concluded that patterns can be found based on clustering. However, this research has some limitations. For this research a limited time span for the data (7 days), only two clustering algorithms, a relatively small research area and three patterns including only four different variables were included. For future research the impact of a larger time span for the data can be examined: will this result in the same patterns? Also other clustering algorithms can be used, like hierarchical clustering: will the result be the same? For this research the performance of the clustering is validated afterwards. For future research, it might be better to calculate indices like the Davies-Bouldin index and the Calinsky-Harabasz index before clustering, because they give information about the best number of clusters for K. For validating the DBSCAN result, the DBCV method can be used, which is not done in this research because of technical failure.

It is also possible to apply the method used for other areas, for example the open sea and compare the results. Ships have more straight tracks in such areas than in the Westerschelde, so results might be different. It could also be interesting to add other sources. For this research it was the initial idea to add sources about the tide. In areas like the Westerschelde, the tide might influence the speed (Kornacki, Mazurek and Smolarek, 2009). This is not done, because only AIS data used for this research. Using tidal data would have extended this research too much in terms of time, because for each moment in time, data about the tide at that moment for a specific location in the research area was needed. For future research it might be interesting to find patterns based on speed and time to find the impact of tides. This can also be used as input for prediction.

8. Bibliography

Baars, M. (2004). *Moving objects in a geo-DBMS Structuring, indexing, querying and visualizing moving point objects in a geo-DBMS context*.

Berkhin, P. (2002). Survey of Clustering Data Mining Techniques. *Grouping Multidimensional Data: Recent Advances in Clustering*. https://doi.org/10.1007/3-540-28349-8_2

Bholowalia, P., & Kumar, A. (2014). EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. In *International Journal of Computer Applications* (Vol. 105).

Bin Mohamad, I., & Usman, D. (2013). Standardization and Its Effects on K-Means Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, *6*(17), 3299–3303.

Bruno, P. (2019, 22 March). Measuring a Vessel's beam. Retrieved from https://www.thoughtco.com/measuring-a-vessels-beam-2292976

Centrale Commissie voor de Rijnvaart [CCR] (2011). *Informatieblad Inland AIS.* Retrieved from https://www.ccr-zkr.org/files/documents/workshops/wrshp181011/Leaflet_Inland_AIS_nl.pdf

CCR (2015). *Informatie met betrekking tot de verplichting tot uitrusting met Inland AIS-apparaten, Inland ECDIS-apparaten of daarmee vergelijkbare visualiseringssystemen.* Retrieved from https://www.ccr-zkr.org/files/documents/ris/brochureAIS_nl.pdf

Choudhary, S. (n.d.). Davies Bouldin Index. Retrieved from https://www.hackerearth.com/problem/approximate/davies-bouldin-index/

Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., & Mathieu, C. (2017). Hierarchical Clustering: Objective Functions and Algorithms. *Journal of the ACM (JACM)*, *66*(4).

Cotteleer, A. (2019, 28 February). AIS data sources. Retrieved from https://mods.marin.nl/display/MIOD/AIS+Data+Sources

Du, K., Monios, J., & Wang, Y. (2019). Green Port Strategies in China. *Green Ports* (pp. 211–229). https://doi.org/10.1016/b978-0-12-814054-3.00011-6

Encyclopeadia Brittannia. (2010, 17 September). Shoal. Retrieved from https://www.britannica.com/science/shoal

ESRI (2020, February 20). Aggregate Points (Cartography). Retrieved from https://pro.arcgis.com/en/pro-app/tool-reference/cartography/aggregate-points.htm

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Retrieved from www.aaai.org

Fluit, A. (2011). AIS information quality report of static AIS messages: "AIS Information Quality Report" Region: HELCOM. Retrieved from http://efficiensea.org/files/mainoutputs/wp4/efficiensea_wp4_13.pdf

Gopal, S., Patro, K., & Kumar Sahu, K. (2015). *Normalization: A Preprocessing Stage*. ArXiv. DOI:10.17148/IARJSET.2015.2305

Gove, R. (2017, 26 December). Using the elbow method to determine the optimal number of clusters

for K-Means clustering. Retrieved from https://bl.ocks.org/rpgove/0060ff3b656618e9136b

Hanyang, Z., Xin, S., & Zhenguo, Y. (2019). Vessel Sailing Patterns Analysis from S-AIS Data Dased on K-means Clustering Algorithm. *2019 4th IEEE International Conference on Big Data Analytics, ICBDA 2019*, 10–13. https://doi.org/10.1109/ICBDA.2019.8713231

Harati-Mokhtari, A., Wall, A., Brooks, P., & Wang, J. (2007). Automatic identification system (AIS): Data reliability and human error implications. *Journal of Navigation*, *60*(3), 373–389. https://doi.org/10.1017/S0373463307004298

Helpdesk Water (2018, 21 August). Scheepvaart. Retrieved from https://www.helpdeskwater.nl/onderwerpen/gebruiksfuncties/scheepvaart/

International Maritime Organization [IMO] (n.d.). AIS transponders. Retrieved on 8 October 2019 from http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx

Jassal, R. (2016, 1 April). Speed through water or speed over ground, which one to use? Retrieved from https://www.myseatime.com/blog/detail/speed-over-water-or-speed-over-ground-which-one-to-use

Kornacki, J., Mazurek, J., & Smolarek, L. (2009). Analysis of the Influence of Current on the Manoeuvres of the Turning of the Ship on the Ports Turning-Basins. *Marine Navigation and Safety of Sea Transportation*, 365.

Lane, R. O., Nevell, D. A., Hayward, S. D., & Beaney, T. W. (2010). Maritime anomaly detection and threat assessment. *13th Conference on Information Fusion, Fusion 2010*. https://doi.org/10.1109/icif.2010.5711998

Lei, P. R. (2019). Mining maritime traffic conflict trajectories from a massive AIS data. *Knowledge and Information Systems*. https://doi.org/10.1007/s10115-019-01355-0

Li, H., Liu, J., Wu, K., Yang, Z., Liu, R. W., & Xiong, N. (2018). Spatio-Temporal Vessel Trajectory Clustering Based on Data Mapping and Density. *IEEE Access*, *6*, 58939–58954. https://doi.org/10.1109/ACCESS.2018.2866364

Liu, Z., Wu, Z., & Zheng, Z. (2019). A novel framework for regional collision risk identification based on AIS data. *Applied Ocean Research*, *89*, 261–272. https://doi.org/https://doi.org/10.1016/j.apor.2019.05.020

Lo Duca, A., Bacciu, C., & Marchetti, A. (2017). A K-nearest neighbor classifier for ship route prediction. *OCEANS 2017 - Aberdeen*, *2017-Octob*, 1–6. https://doi.org/10.1109/OCEANSE.2017.8084635

MarineTraffic (2017, 23 September). What kind of information is AIS-transmitted. Retrieved from 205426887-What-kind-of-information-is-AIS-transmitted-

Moulavi, D., Jaskowiak, P. A., Campello, R. J. G. B., Zimek, A., & Sander, J. (2014). Density-Based Clustering Validation. *Proceedings of the 14th SIAM International Conference on Data Mining (SDM)*.

Ng, R. T., & Han, J. . (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge & Data Engineering*, (5), 1003–1016.

Ou, Z., & Zhu, J. (2008). AIS Database Powered by GIS Technology for Maritime Safety and Security. *Journal of Navigation*, *61*(4), 655–665. https://doi.org/10.1017/S0373463308004888

Pallotta, G., Vespe, M., & Bryan, K. (2013). Traffic knowledge discovery from AIS data. *Proceedings of the 16th International Conference on Information Fusion*, 1996–2003.

Parimala, M., Lopez, D., & Senthilkumar, N. C. (2011). A Survey on Density Based Clustering Algorithms for Mining Large Spatial Databases. In *International Journal of Advanced Science and Technology* (Vol. 31).

Reddy, C. K., & Vinzamuri, B. (2019). A Survey of Partitional and Hierarchical Clustering Algorithms. In *Data Clustering* (pp. 87–110). https://doi.org/10.1201/9781315373515-4

Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of Computers and Communications*, *5*(1), 27–34.

Retsch, J. (2018, 24 April). What is the difference between IMO and MMSI number? Retrieved from https://help.fleetmon.com/en/articles/2010884-what-is-the-difference-between-imo-and-mmsi-number

Reynolds, A. P., Richards, G., & Rayward-Smith, V. J. (2006). Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms. *Journal of Mathematical Modelling and Algorithms*, *5*, 475–504. https://doi.org/10.1007/s10852-005-9022-1

Rhodes, J. D., Cole, W. J., Upshaw, C. R., Edgar, T. F., & Webber, M. E. (2014). Clustering analysis of residential electricity demand profiles. *Applied Energy*, *135*, 461–471. https://doi.org/10.1016/j.apenergy.2014.08.111

Rijkswaterstaat (2018, 1 September). Nationaal Wegen Bestand (NWB) - Vaarwegen - vaarwegvakken (RWS) https://data.overheid.nl/dataset/58741-nationaal-wegen-bestand--nwb----vaarwegen---vaarwegvakken--rws-

Rijkswaterstaat (2014, 8 December). KRW oppervlaktewaterlichamen RWS 2014 vlakken (Kaderrichtlijn Water)(RWS). Retrieved from https://data.overheid.nl/dataset/58723-krw-oppervlaktewaterlichamen-rws-2014-vlakken--kaderrichtlijn-water--rws-

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., … Lin, C. T. (2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664–681. https://doi.org/10.1016/j.neucom.2017.06.053

Scheldemonden. (n.d.). Scheldemonden. Retrieved on 3 December, 2019 from https://scheldemonden.loodswezen.nl/

Scikit-Learn (2020a, February 21). 2.3 Clustering. Retrieved from https://scikit learn.org/stable/modules/clustering.html

Scikit-Learn (2020b, February 21). Selecting the number of clusters with silhouette analysis on KMeans clustering. Retrieved from https://scikit learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Sheng, P., & Yin, J. (2018). Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. *Sustainability*, *10*(7), 2327. https://doi.org/10.3390/su10072327

Silveira, P. A. M., Teixeira, A. P., & Soares, C. G. (2013). Use of AIS data to characterise marine traffic patterns and ship collision risk off the coast of Portugal. *Journal of Navigation*, *66*(6), 879–898. https://doi.org/10.1017/S0373463313000519

Svanberg, M., Santén, V., Hörteborn, A., Holm, H., & Finnsgård, C. (2019). AIS in maritime research. *Marine Policy*, *106*, 103520. https://doi.org/https://doi.org/10.1016/j.marpol.2019.103520

Swarndeep, S. J., & Pandya, D. S. (2016). An overview of partitioning algorithms in clustering techniques. *International Journal of Advanced Research in Computer Engineering & Technology*

(IJARCET), *5*(6).

Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., & Huang, G. Bin. (2018). Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey from Data to Methodology. *IEEE Transactions on Intelligent Transportation Systems*, *19*(5), 1559–1582. https://doi.org/10.1109/TITS.2017.2724551

United States Coast Guard Navigation Center (2019, 17 April). CLASS A AIS POSITION REPORT (MESSAGES 1, 2, AND 3). Retrieved from https://www.navcen.uscg.gov/?pageName=AISMessagesA

Van der Zee. (2015, 31 March). Scheefheid. Retrieved from: https://hulpbijonderzoek.nl/onlinewoordenboek/scheefheid/

Wang, X., Smith, K., & Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery*, *13*(3), 335–364. https://doi.org/10.1007/s10618-005-0039-x

Wang, Z., Claramunt, C., & Wang, Y. (2019). Extracting Global Shipping Networks from Massive Historical Automatic Identification System Sensor Data: A Bottom-Up Approach. *Sensors*, *19*(15), 3363. https://doi.org/10.3390/s19153363

Wu, D., Taheri, E., & Kolmanovsky, I. (2018). Complex interplanetary trajectories design with low-thrust based motion primitives. *Space Flight Mechanics Meeting, 2018*. https://doi.org/10.2514/6.2018-0215

Yang, D., Wu, L., Wang, S., Jia, H., & Li, K. X. (2019). How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications. *Transport Reviews*, *39*(6), 755–773. https://doi.org/10.1080/01441647.2019.1649315

Yilou, P., & Dejun, C. (2018). A Research on Dynamic Identification of Main Route of River Estuary Segment Based on AIS. *2018 3rd IEEE International Conference on Intelligent Transportation Engineering, ICITE 2018*, 209–213. https://doi.org/10.1109/ICITE.2018.8492701

Zhou, Y., Daamen, W., Vellinga, T., & Hoogendoorn, S. P. (2019). Ship classification based on ship behavior clustering from AIS data. *Ocean Engineering*, *175*, 176–187. https://doi.org/10.1016/J.OCEANENG.2019.02.005

9. Appendix

The appendices to this research contain the PostgreSQL queries and the Python scripts used.

Appendix I: PostgreSQL queries

Pattern 1

Select different static mmsi for 1 day:

- select distinct mmsi from static10;

Combine stat en dyn:

create table wsstatdyn10

as

SELECT distinct static10.mmsi as mmsistatic, static10.bow, static10.stern, static10.port, static10.starboard, ws10dynamic.mmsi as mmsidyn

FROM static10

INNER JOIN ws10dynamic

ON static10.mmsi = ws10dynamic.mmsi;

---

alter table wsstatdyn10

add column lengt integer,

add column beam integer;

---

update wsstatdyn10

  set lengt = bow + stern,

  beam = port + starboard;

---

delete from wsstatdyn10

  where lengt=0

  or beam=0;

Check which mmsi appears more than once:

with counting as (select mmsistatic as mmsicount, count (*) from wsstatdyn10

group by mmsistatic

```
having count(*)>1)


select counting.mmsicount, wsstatdyn10.bow, wsstatdyn10.stern, wsstatdyn10.port,
wsstatdyn10.starboard, wsstatdyn10.lengt, wsstatdyn10.beam from wsstatdyn10
inner join counting on wsstatdyn10.mmsistatic = counting.mmsicount;
```

Same lengt hand beam are ok:

```
create table wsstatdyn102
as
SELECT distinct wsstatdyn10.mmsistatic, wsstatdyn10.lengt, wsstatdyn10.beam fROM wsstatdyn10;
```

```
with counting as (select mmsistatic as mmsicount, count (*) from wsstatdyn102
group by mmsistatic
having count(*)>1)


select counting.mmsicount, wsstatdyn102.lengt, wsstatdyn102.beam from wsstatdyn102
inner join counting on wsstatdyn102.mmsistatic = counting.mmsicount
order by mmsicount;
```

Check length and beam:

```
select distinct callsign, shipname from static10
where mmsi = 244670369;
```

Final table:

```
create table mmsisize102
(mmsi integer, lengt integer, beam integer);


Insert into mmsisize102 (mmsi, lengt, beam)
SELECT mmsistatic as mmsi,
   MAX(lengt) AS lengt,
   MAX(beam) AS beam
```

```
FROM wsstatdyn102
GROUP BY wsstatdyn102.mmsistatic;
```

Minmax-scaling

```
Alter table mmsisize103min
add column minmaxlength real,
add column minmaxbeam real;
-----------------------------------------------------------------
UPDATE mmsisize103min mm
  SET minmaxlength = 1.00 * (mm.lengt - x.minlength) / x.lengthRange
FROM
  (
    SELECT lengt,
        min(lengt) OVER () AS minlength,
        max(lengt) OVER () - min(lengt) OVER () AS lengthRange

    FROM mmsisize103min
  ) x ;
-----------------------------------------------------------------------------------------------
UPDATE mmsisize103min mm
  SET minmaxbeam = 1.00 * (mm.beam - x.minbeam) / x.beamRange
FROM
  (
    SELECT beam,
        min(beam) OVER () AS minbeam,
        max(beam) OVER () - min(beam) OVER () AS beamRange

    FROM mmsisize103min
  ) x ;
```

Update table after clustering:

```
alter table minmaxsize10
add column cluster2 integer,
```

```
add column clustertot2 varchar;

-----------------------------------------

UPDATE mmsisizetot

SET    cluster2 = clustertot2.cluster2

FROM   clustertot2

WHERE  mmsisizetot.mmsi = clustertot2.mmsi;

-----------------------------------------------------------

UPDATE mmsisizetot

SET    clustername5 = clustertot5.clustername5

FROM   clustertot5

WHERE  mmsisizetot.mmsi = clustertot5.mmsi;

-------------------------------------------------------------------------------------------------------------------
```

All days to one table

```
CREATE TABLE statictotal

AS

  SELECT * FROM static10

    UNION

  SELECT * FROM static11

 UNION

  SELECT * FROM static12

    UNION

  SELECT * FROM static13

Union

  SELECT * FROM static14

    UNION

  SELECT * FROM static15

Union

Select * from static16;
```

SAME QUERIES ARE APLLIED AS FOR ONE DAY

Join dynamic and static with cluster results

```
ALTER TABLE dynamicws
```

```
ADD COLUMN lengt integer,
add column beam integer,
add column minmaxlength real,
add column minmaxbeam real,
add column cluster5 integer,
add column clustername5 character varying,
add column cluster2 integer,
add column clustername2 character varying;
```

```
UPDATE dynamicws
SET lengt = mmsisizetot.lengt,
beam = mmsisizetot.beam,
minmaxlength = mmsisizetot.minmaxlength,
minmaxbeam = mmsisizetot.minmaxbeam,
cluster5 = mmsisizetot.cluster5,
clustername5 = mmsisizetot.clustername5,
cluster2 = mmsisizetot.cluster2,
clustername2 = mmsisizetot.clustername2
FROM
mmsisizetot
where dynamicws.mmsi = mmsisizetot.mmsi;
```

```
create table dynamicwskmnonull as (select * from dynamicwskm);
```

```
delete from dynamicwskmnonull
where lengt is null;
```

Add dbscan clusters

```
alter table dynamicwskmnonull
add column dbcluster character varying;
```

```
UPDATE dynamicwskmnonull
SET dbcluster = dbclusterp1.dbcluster
FROM
dbclusterp1
where dynamicwskmnonull.mmsi = dbclusterp1.mmsi;
```

Split 70/30%

```
update mmsisizetot
set random=
Case when random() < 0.7 then 70
else 30
end;
```

```
update dynamicwsclnonull
set random =  mmsisizetot.random
FROM mmsisizetot
WHERE dynamicwsclnonull.mmsi = mmsisizetot.mmsi;
```

```
INSERT INTO dynamicclkmpart
SELECT *
FROM dynamicwsclnonull
WHERE random = 30;
```

Add tables with clustering result. Import values to main table:

```
alter table dynamickmpart
add column clname230 character varying,
add column clname530 character varying;
```

```
update dynamickmpart
set clname530 =  cl5km30.clustername5
FROM cl5km30
```

```
WHERE dynamickmpart.id = cl5km30.id;

-----------------------------------------------------------------
```

Pattern 2

```
create view mov_obj_succ as

select t1.*,  t2.fid AS next_fid, t2.ts AS next_ts, t2.longitude AS next_long,

          t2.latitude AS next_lat

  FROM dynamicwsp3nonull t1,

   dynamicwsp3nonull t2

where t1.mmsi=t2.mmsi and t2.ts=(select min(ts) from dynamicwsp3nonull where ts>t1.ts and
mmsi=t1.mmsi);

---------------------------------------------

COPY (SELECT * FROM mov_obj_succ) TO 'C:\Users\public\timep3.csv' DELIMITER ',' CSV HEADER;

------------------------------
```

Table with speed and differences, select best speed:

```
alter table speedwithoutnull

add column bestspeed real;

-----------------------------------------------------

update speedwithoutnull

set bestspeed = case when speed < 31 and speeddif between -5 and 5 then speed

end;

-----------------------------------------------------------

update speedwithoutnull

set bestspeed = case when speed < 31 and speeddif between -5 and 5 then speed

                                                    when speed < 31 and speedknots < 31 then
speedknots

                                                    end;

-----------------------------------------------------------

update speedwithoutnull

set bestspeed = case when speed < 31 and speeddif between -5 and 5 then speed

                                                    when speed > 31 and speedknots < 31 then
speedknots
```

when speed < 31 and speeddif < -15 or mmsi=376665000 or mmsi=246656000 or mmsi=205652000 or mmsi=246586000 or mmsi=246587000 then speeddif

end;

Remove NULL values:

```
delete from speedwithoutnull

where bestspeed is null;

----------------------------------------

update speedwithoutnull

set speed= speed * -1 where speed <1;

-----------------------------------------------

Alter table speedwithoutnull

add column minmaxspeed real;

-----------------------------------------------

alter table speedwithoutnull

add column id serial primary key;

-----------------------------------------------

UPDATE speedwithoutnull dw

  SET minmaxspeed = 1.00 * (dw.speed - x.minspeed) / x.speedRange

FROM

  (

    SELECT id, speed,

         min(speed) OVER () AS minspeed,

         max(speed) OVER () - min(speed) OVER () AS speedRange


    FROM speedwithoutnull group by id

  ) x

        where dw.id=x.id;
```

CHANGE NAME BESTSPEED IN SPEED AND REMOVE SPEED COLUMN

```
ALTER TABLE speedwithoutnull

add column minmaxlength real,
```

```
add column minmaxbeam real;

-----------------------------------------------------------------------------------------------

UPDATE speedwithoutnull

set

minmaxlength = mmsisizetot.minmaxlength,

minmaxbeam = mmsisizetot.minmaxbeam,

FROM

mmsisizetot

where speedwithoutnull.mmsi = mmsisizetot.mmsi;

-----------------------------------------------------------------------------------------------------------------------------------

COPY (SELECT * FROM speedsizewithoutnull) TO 'C:\Users\public\speedsize.csv' DELIMITER ',' CSV
HEADER;

-------------------------------------------------------------------------

ALTER TABLE speedwithoutnull

add column lon double precision,

add column lat double precision,

add column ts timestamp with time zone;

-----------------------------------------------------------------------------

UPDATE speedwithoutnull

set

lat = speedlonlatts.lat,

lon = speedlonlatts.lon,

ts = speedlonlatts.ts

FROM

speedlonlatts

where speedwithoutnull.fid = speedlonlatts.fid;

-------------------------------------------------------------------------

update speedwithoutnull

set cl2 = cl2kmp2.cl2

from cl2kmp2

where speedwithoutnull.id=cl2kmp2.id;
```

```
-----------------------------------------------------------------

alter table speedwithoutnull

add column random integer;

-------------------------------------------------

update speedwithoutnull

set random=

Case when random() < 0.7 then 70

else 30

end;

-----------------------------------------

INSERT INTO p2part30

SELECT *

FROM speedwithoutnull

WHERE random = 30;

---------------------------------------

Update p2part30

set cl4 = cl4kmp2.cln4

FROM cl4kmp2

WHERE p2part30.id = cl4kmp2.id;

-----------------------------------------------

Update p2part30

set cl230 =  p2name230.p2230

FROM p2name230

WHERE p2part30.id = p2name230.id;

-------------------------------------------------------
```

Pattern 3

```
ALTER TABLE speedwithoutnull

add column neardist real;

-----------------------------------------------------------------------------

UPDATE speedwithoutnull

set
```

```
neardist = distfairway.neardist
FROM
distfairway
where speedwithoutnull.fid = distfairway.fid;
-----------------------------------------------------------------------
ALTER TABLE speedwithoutnull
add column minmaxdist real;
-------------------------------------------------------------
alter table speedwithoutnull
add column id serial primary key;
-------------------------------------------------------------
UPDATE speedwithoutnull dw
   SET minmaxdist = 1.00 * (dw.neardist - x.mindist) / x.distRange
FROM
   (
      SELECT id, neardist,
            min(neardist) OVER () AS mindist,
            max(neardist) OVER () - min(neardist) OVER () AS distRange

      FROM speedwithoutnull group by id
   ) x
          where dw.id=x.id;
-----------------------------------------------------------------------------------------------
Update p2part30
set cl5p330 =  cl5p330.cln530
FROM cl5p330
WHERE p2part30.id = cl5p330.id;
```

Visualisation

```
alter table speedwithoutnull
add column cl5p3 character varying,
```

```
add column cl3p3 character varying;

-----------------------------------------------------

Update speedwithoutnull

set cl5p3 =  cl5p3.cln5

FROM cl5p3

WHERE speedwithoutnull.id = cl5p3.id;

-------------------------------------------------
```