Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Segmentation optimization in trajectory-based ship classification

Daniel Amigo [*], David Sánchez Pedroche, Jesús García, José Manuel Molina

Group GIAA, University Carlos III of Madrid, Spain

A R T I C L E I N F O

Keywords:
Trajectory segmentation
Kinematic behaviour
AIS data
Class imbalance
Ship classification

A B S T R A C T

The paper presents an analysis over eleven trajectory segmentation techniques applied to the study and experimentation of ship classification problems based only on kinematic information. Using the experimental framework introduced in previous works, it cleans, smooths and extracts trajectories from real-world Automatic Identification System (AIS) data. It also applies three balancing solutions to address the lack of an equal distribution among classes. In total, 196 classification experiments have been carried out, which have been presented with a multi-objective analysis to consider the imbalance problem and conflicting metrics (total and minority class accuracies). The results show a Pareto front with different viable solutions for the classification problem, without a dominant one over the rest. The segments generated in the best experiments (Pareto front) are analysed using specific metrics to compare their impact in the classification problem.

1. Introduction

Maritime surveillance systems are an essential element for the protection of the seas, ensuring the safety of maritime transport and security of citizens. The detection and monitoring of vehicles are a solved problem using multiple technologies. In contrast, more specific problems, such as classification of the type of ship or its current manoeuvre are far from being a solved problem, which are essential for decision-making in maritime surveillance systems. Technologies such as AIS provide information that allows the target identification and behaviour identification [1]. However, AIS is not totally reliable, as it is manually adjusted, and its technology is susceptible to manipulation.

The problem of this study is the classification of trajectories to obtain the type of ship based on kinematics data that model its behaviour. This is an extension of a previous study [2–4], where the problem was defined and main subprocesses identified. These first approaches concluded that it was necessary to specifically analyse the impact of each subprocess on the classification. Thus, the objective of this paper is to study the impact of segmentation on the final performance, observing whether there are benefits compared to the fixed-size segmentation initially proposed. To achieve it, more complex segmentation techniques, both classical and recent, are studied and analysed, generating variable size segments that can better adjust the ships' motion.

To move from the sensor measurements to a ship classification, it is required a framework that performs different processes on the data. Specifically the Ship Type Detection System (STDS hereafter) developed and detailed in previous works [2–4]. will be used. As those works conclude, it is necessary to perform a specific analysis on the segmentation sub-component, the aim of this work. Therefore, that sub-component is the only modification of the STDS used here.

A short overview of the STDS components is presented. The first component performs the data preparation. To clean real-world reports, an IMM (interacting multiple model) filter is used to reduce the noise by smoothing the target trajectory. The next step is the segmentation of trajectories, splitting the original track by applying different criteria (uniform length, shape or direction preserving…). Later, a process handles of the data imbalance as the ship types are not distributed in a homogeneous manner (neither in trajectories nor segments). Finally, the classification is performed by using different algorithms applied to track segments to predict the ship type. Specifically, the objective is to determinate the membership in the fishing class, which is the minority in the used dataset. Before applying the classification component, the feature extraction sub-component extracts representative features from each trajectory segment that summarizes the dynamic of the vessel.

Abbreviations: AIS, Automatic Identification System; BOPW, Before Opening Window; DAG, Directed Acyclic Graph; DOTS, DAG based Online Trajectory Simplification; DP, Douglas Peucker; IMM, Interacting Multiple Model; MRPA, Multi Resolution Polygonal Approximation; NOPW, Normal Opening Window; OPW, Opening Window; PED, Perpendicular Euclidean Distance; SED, Synchronized Euclidean Distance; SMOTE, Synthetic Minority Oversampling TEchnique; STDS, Ship Type Detection System; SVM, Support Vector Machine; TD, Top Down; TR, Time Ratio.

[*] Corresponding author.

E-mail addresses: damigo@inf.uc3m.es (D. Amigo), davsanch@inf.uc3m.es (D.S. Pedroche), jgherrer@inf.uc3m.es (J. García), molina@ia.uc3m.es (J.M. Molina).

https://doi.org/10.1016/j.jocs.2022.101568
Received 14 April 2021; Received in revised form 1 November 2021; Accepted 18 January 2022
Available online 31 January 2022

These variables are presented later. Although other variables related to the trajectory's context could provide useful information to classify them, as the global position or the coast distance, STDS seeks to avoid this type of information. This is because it aims to find a solution based on as little information as possible, focusing only on the track kinematics, which could be improved later by including the context information.

The experiments, on the one hand, compare various segmentation techniques with respect to the original segmentation (fixed length). The results show the trade-off between accuracy and imbalance of classification so there is not an absolute optimal solution, but makes it clear the multi-objective nature of the problem, and solutions show a Pareto front. On the other hand, the segmentation algorithms that form the Pareto front, are analysed specifically. Its generated segments are compared in two ways:

- Globally by using representative aggregate values for each segmentation algorithm and comparing between them.
- Specifically, by analysing two trajectories, one of each class, showing the practical difference between the initial segmentation approach and the more complex ones.

These analyses are performed by extracting specific metrics of the trajectory segmentation problem, such as the compression ratio and the residual of the points of the trajectory to the segment.

The complex and modern segmentation algorithms evaluated (DOTS, SQUISH-E) show how the segments are more representative and obtain a better result than the Uniform sampling technique with all segmentation metrics studied. Still, there is considerable room for improvement in the classification problem.

The main contributions of this paper can be summarised as follows:

- Identification and study of several relevant state-of-the-art segmentation algorithms.
- Implementation, testing and analysis of different segmentation algorithms together with several configuration parameters.
- Analysis and identification of the experiments that provide better results within the classification problem, obtaining a Pareto front better than the original results.

This paper is an expansion of [5], and it is organized as follows: In section II several methods in segmentation of trajectories of the literature are explained. In section III the framework process that performs the trajectory-based classification is explained. Section IV explains all the segmentation algorithms experiments that will be the input of the classifier. In section V results of the work are shown, first the classification output and later an extensive analysis of the segmentation results. Finally, the conclusions and perspectives for future works are presented in section VI.

2. State of the art

This state of the art looks specifically at the main problem addressed in this paper: trajectory segmentation and how it affects the trajectory classification problem. A basic problem of classification uses the available information to infer intelligence with diverse methods. In this domain, the available information is the vehicle's track, including position (latitude and longitude) with time. Such information is very limited for any classifier, so processing this data to infer additional knowledge is necessary to help the classifier to bias the inputs. This process is called feature extraction. For example, these recent studies [6–8] perform a feature extraction on the trajectory of the ship to determine its behaviour. This feature extraction is not adequate for a problem where long-duration trajectories or very heterogeneous mixture of trajectories appear.

As an alternative, feature extraction can be applied on each segment instead of the whole track to extract more precise information for the classifier. There are researchers [9] who perform a segmentation before classification, but they use their own segmentation technique very specific to their problem. Alternatively, this paper experiments with some classical and recent segmentation techniques to analyse how they influence the problem of classification trajectories. Note also that all these papers use context information, making them incomparable with the present proposal.

Vehicle trajectories, particularly in the maritime domain, are very long and complex, with too much information for a classifier to provide. Splitting them into shorter segments reduces their information and is useful in any learning problem. However, not all segmentations are adequate for each problem. For example, if it is desired to classify the curves, the segmenter must split the trajectory to maintain those curves, so that the classifier can learn them. In complex problems, like this one, it is unknown which type of segments is the best one to generate for a classifier.

The field of trajectory segmentation has several approaches [10]. One of them is to use compression algorithms, which identify the key-points of the trajectory and use them to generate the segments. Segments are generated according to different conditions, e.g., time gaps, trajectory shape or its semantic context. Also, they can be categorized according to whether they need the entire track (offline), or they can run in real time (online).

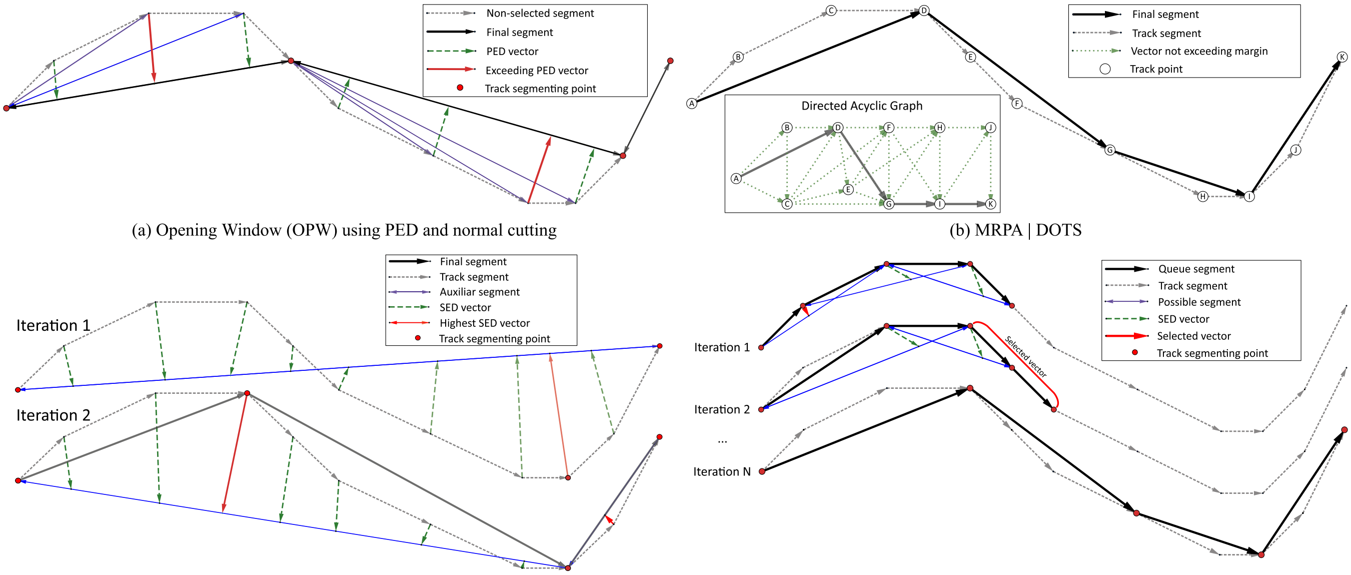
The simplest approach to segmentation is Uniform sampling, which cut the track into segments of uniform size [11] (the approach used in the previous works). This paper explores segmentation algorithms according to the trajectory shape, generating segments that minimize error with respect to the trajectory. Fig. 1 illustrates several segmentation algorithms achieving different outputs on the same track.

The classic algorithms for segmentation are:

- Opening Window (OPW) [12]: This process, also known as Sliding Window in the literature, generates variable size segments by setting the start of the segment and searching for the end. To find the end, as is shown in Fig. 1(a), it evaluates each following point calculating the error of the segment with respect each point in the window (between segment start and segment end). When the error exceeds a threshold, the current segment is closed. Then from this point a new segment is started, restarting the window. The same process will be performed until the trajectory end.
- Top-down [13]: It starts with a segment that covers the entire trajectory. Then it selects the trajectory point with the highest residual from the segment according to a specific error metric. That trajectory point is used to divide the segment. Two segments are then generated, and both evaluated with the same process, making recursively divisions as the two shown in Fig. 1(c). This process continues until the highest residual is below a predefined stop threshold.
- Bottom-up: The inverse process to Top-Down. It starts with small segments and the process calculates hypothetical segments that unifies each pair of real segments. The hypothetical segment with the smallest residual is transformed into a real one and the process iterates, creating new hypothetical segments. The process works in a recursive manner until all errors are over a predefined stop threshold, meaning that the segments cannot be unified anymore.

These algorithms need an error measure of the segment to make the comparison with the defined threshold. Typically, the segment error is calculated in relation to the trajectory by using the Perpendicular Euclidean Distance (PED) of each point. A big improvement is to use instead of PED, the Synchronized Euclidean Distance (SED) [12], which take into consideration track point timestamp with regard to the segment total time. Other improvements use more variables as direction variation or speed in an equal manner as the time [12,14].

Based on the previous classic approaches there are many other algorithms that seek a better performance when performing the



(a) Opening Window (OPW) using PED and normal cutting

(b) MRPA | DOTS

(c) Top-Down using SED, known as TD-TR

(d) SQUISH-E, with a queue size of 4 points

Fig. 1. Example of how several track segmentation algorithms work.

segmentation, being of interest the following ones:

- SQUISH-E [15]: It works by creating a fixed size queue with the first trajectory points. On each iteration, the process adds a new point to the queue. As it exceeds the size, the process finds the point of the queue to remove. As bottom-up does, hypothetical segments within the window are created. It selects the segment with the smallest SED residual to maintain the queue size. This SED value of the eliminated point is shared among the neighbours as a weight, to be considered in the following iterations. Fig. 1(d) shows this procedure, checking in trios the less relevant point and removing it from the queue. This algorithm uses two parameters for shaping the resulting segment: λ guarantees a compression ratio of the track (used to calculate the queue size), while μ indicates the maximum SED error.
- MRPA [16]: Which comes from Multi Resolution Polygonal Approximation. It works by approaching the track based on a bottom-up multiresolution approach, creating a DAG (Directed Acyclic Graph) with the multiple candidate curves (each potential set of segments) for different error tolerances. The DAG is represented in Fig. 1(b). These DAG approaches were previously common, but due to the high computational time required, a reduction of the search space is being explored, as MRPA does. The error criterion used is the accumulated variation of SED, LSSD. Then it selects the most suitable curve among those approximated in the DAG using an error tolerance.
- DOTS [17]: Which comes from DAG based Online Trajectory Simplification. This algorithm is like the MRPA algorithm, algorithm, but its main feature is the online execution capability, being able to obtain the segments in real time. It uses a DAG to describe potential segments of the trajectory, as can be shown in Fig. 1(b). To achieve the online operation of the algorithm it simplifies the DAG into a tree, which vertices are the location points of the track while each tree depth level represents the segments. On each iteration the same ISSD criterion is applied locally over the new track point to adjust edges between consecutive layers (the edges represent the segments with an accepted ISSD).
- All these segmentation algorithms are studied in this paper by analysing which ones work best for the ship type classification problem.

3. Ship-type determination using binary classification

This section provides a brief explanation of the STDS, summarizing its main subprocesses, starting from the input data up to the classification algorithms. It was fully detailed in [2–4].

To summarise, the STDS consists of data cleaning, including the use of a filter to reduce noise in the classifier inputs. With the cleaned data, the segmentation process is used (this process is the one expanded in this article), and, over the segments, the data balancing process is applied. Finally, classification is applied, although the input is not the whole segment, but a series of features extracted from the trajectory (see Fig. 2).

The first step is cleaning the raw data from sensors. In this case, the available data is from AIS sensor. It provides kinematic data of ships integrated with additional information such as the ship type, which is used here to train the classifier. Specifically, the chosen repository is the one provided by the Danish Maritime Authority [18], in which there is a recompilation of daily AIS contacts since 2006. Dealing with real-world raw data requires a strong pre-processing which is critical for final performance, removing inconsistencies, null, wrong, and noisy values. These problems are generated by malfunction of AIS transmitters and human errors. The measurement noise taken by the sensor can either be outliers, directly detectable evaluating the offset in GPS coordinates, or small noises that can be smoothed by a filtering algorithm. An IMM filter has been implemented to smooth the noise, configured with two Extended Kalman Filters as modes of prediction for ship trajectories: one for linear movements and low prediction noise and other to model the movements that would be considered noisy (speed variations, turns,.).

Prior to classification, its necessary a process to address the unbalance problem present in this domain due to the lack of an equal distribution among classes. For instance, long and frequent trajectories of cargo and passenger vessels populate the training data sets and bias the classification models towards these categories reducing the representation of other ones, like the fishing vessel category. To solve the problem, the system implements oversampling and undersampling techniques, which adjust the amount of data of each class by adding or removing instances [19]. The experimentation uses the original imbalanced dataset, and two balanced datasets: one using random undersampling, randomly removing instances of the majority classes, and another using the SMOTE algorithm [20], already used for track classification [7], oversampling the minority class by creating new artificial samples.

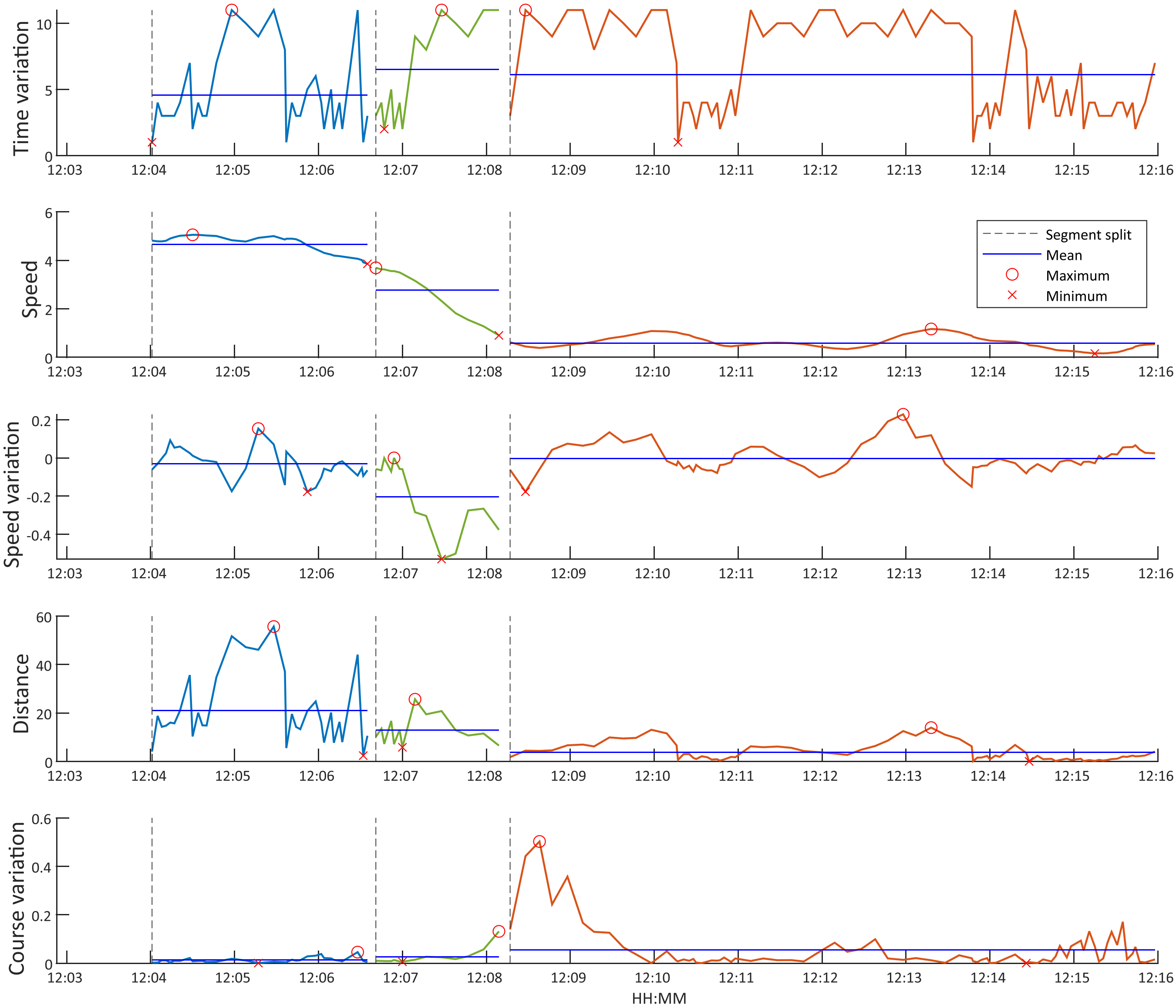


Fig. 2. Kinematic feature extraction example.

The classification is based on a set of features extracted from each segment as shown in Fig. 2. From the track points the process extract the following kinematic parameters:

- Time variation between measurements, considering the time gaps between two track points.
- Speed, characterizing the speed module between two track points.

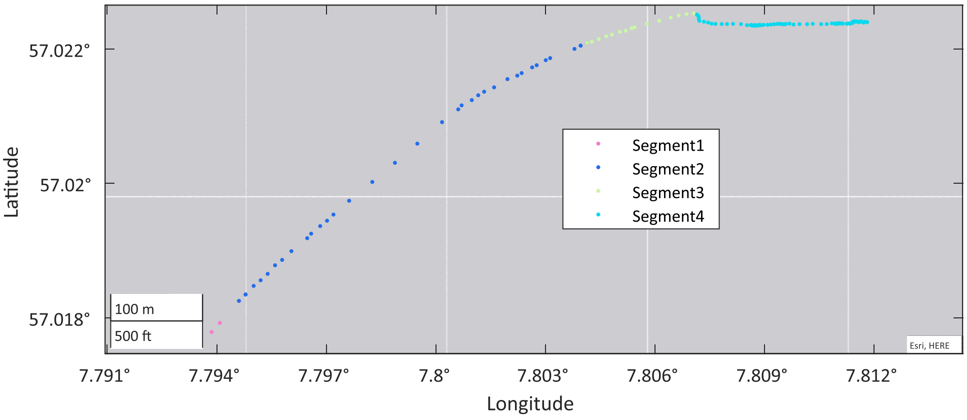


Fig. 3. Trajectory segmentation example.

- Speed variation, describing acceleration or deceleration produced between two track points.
- Distance, characterizing movement range and complexity between two track points.
- Course variation, describing the turnarounds between two track points.

Because the possible difference in the number of measures between segments, is necessary to make those kinematic variables suitable as a classification input. The following statistical measures are applied to aggregate and resume all the segment track points: the mean, maximum, minimum, mode, standard deviation and three quartiles. Also, the total time of the segment is included to support the time gaps variables.

To illustrate this process, the trajectory shown in Fig. 3 will be used. Fig. 2, alongside the kinematic parameters of each track point, the statistical values of minimum, maximum and mean have been remarked for each segment. It can be seen how the track point features have variation in time, while the statistical values acquire a single value that summarise the feature for the whole segment. These statistical values are the extracted features that will act as input for the classification algorithms.

Note that these segments are contiguous, belonging a trajectory point to two segments. For simplicity, as some kinematic parameters require two measurements to be calculated, it has been decided to eliminate the first measurement of each segment in the figure, generating a gap between them.

The classification problem considered in this work is predicting when a vessel is of fishing type and when it is not, i.e., a binary classification problem. Common classification algorithms in binary problems as the Support Vector Machine (SVM) and the decision tree algorithm are chosen, looking to keep the importance on the segmentation problem by using simple and well-known techniques but able to perform it.

To evaluate the results obtained by the classification it is necessary to consider two main factors, the accuracy of the general classification and the specific accuracy on the minority class (fishing), which is affected by the imbalance in the training process. Therefore, along with the classification accuracy, the F-measure metric [21], along with the sensitivity and precision values, is considered to assess both effects.

The simultaneous evaluation of both metrics prevents the domination of the classification accuracy by the effect of majority class. Besides, the presence of these two metrics makes the problem multi-objective, allowing to observe the Pareto’s front when displaying the results from different algorithms and their parameters.

Apart from the classification result, it is possible to evaluate the quality of the segments generated by the different segmentation algorithms to be used. To evaluate the difference between a trajectory and its segments is possible to use accuracy metrics that define the similarity between two trajectories (the real one, and the one formed by the segments).

PED and SED metrics, used within the segmentation algorithms to measure error of each segment, are metrics that could evaluate the segmentation result, as the distance between two trajectories is a measure of the accuracy of the segmentation process.

Also, the PED and SED allow the projection of the trajectory point into the segment, being possible to compare the projection movement parameters with the trajectory point movement parameters.

4. Trajectories segmentation experiments

This section presents the different experiments to be carried out using the track segmentation algorithms. Each algorithm has different parameters to set its functionality depending on the problem. In this case, as the configuration of each algorithm is not trivial with respect to its impact on the classification, different experiments are performed, varying from each of the parameters, allowing an analysis of the impact of each of them. A summary of the variations of each algorithm is shown in Table 1 and a detailed explanation of the 196 experiments tested in this paper is given below.

The base case used in STDS uses a Uniform sampling of 50 measurements (around 9 min). For comparison, tests with 10 and 20 measurements are performed as well.

Opening window (OPW) has the following variants from its base implementation:

- The cut-off criterion: whether it occurs at the point where the window has exceeded the error (NOPW), and whether it is done at the previous point (BOPW) [12].
- Error evaluation functions: PED or SED ("_TR", meaning Time-Ratio [12]), illustrated in Fig. 1. Three error values (20, 30 and 50 m) are tested with each function to divide the segment if the error is over the value.
- To ensure that the segments are generated with a minimum length, favouring the classification. A minimum segment size its tested with the values 0, 10, 20 and 50 measurements.

Top Down algorithm has variations for the error evaluation function, marked as “DP” (Douglas Peucker algorithm [13]) when it uses PED and as “TD_TR” when it uses SED [12]. These variations use the same error and minimum segment size values as OPW.

Bottom Up has no relevant variations according to the error function, as only the PED error function has been used in the literature.

SQUISH-E only uses the SED error function, with the same three error values already listed as μ value. In addition, it has the compression parameter λ, testing 1, 5 and 10 values.

Finally, both DOTS and MRPA only vary on the error values, using 100 and 500 as values for its accumulative SED variation.

To evaluate the segmentation performed by each experiment, the following metrics are calculated on each generated segment:

- Mean, minimum, maximum, and standard deviation for PED and SED error applied over:
- Position difference between each track point and the projection point in the segment.

Table 1
Segmentation algorithms variations.

Base algorithm	Variations	Subvariations	Error function	Error value (metres)	Segment size (points)
Uniform sampling	–	–	–	–	10, 20, 50
OPW	OPW	BOPW	PED, SED	20, 30, 50	10, 20, 50
	OPW_TR	NOPW	PED, SED	20, 30, 50	10, 20, 50
		BOPW_TR	PED, SED	20, 30, 50	10, 20, 50
		NOPW_TR	PED, SED	20, 30, 50	10, 20, 50
TopDown	DP	DP	PED, SED	20, 30, 50	10, 20, 50
	TD_TR	TD_TR	PED, SED	20, 30, 50	10, 20, 50
BottomUp	–	–	PED	20, 30, 50	–
SQUISH-E	–	–	SED	20, 30, 50	1, 5, 10
DOTS	–	–	ISSD	100, 500	–
MRPA	–	–	ISSD	100, 500	–

- Speed module difference between each track point and the projection point in the segment.
- Angle difference between each track point and the projection point in the segment.
- The number of segments of the trajectory and the average that are within each segment, allowing the analysis over the segment's features. Related with those, the compression rate allows the measure of how compressed a trajectory after the segmentation is.

Fig. 4 represents how the angle difference and speed are calculated for each point. α and β are the angle difference for the second and third trajectory points, while *ΔSpeedF* is the speed modulus difference for F point between the track point and the projected one (*F′*).

5. Results analysis

5.1. Classification results

The performed experimentation is applied over three days in July 2017 from AIS contacts off the coast of Denmark. In total, more than 30 million contacts are available as system inputs. After the cleaning process, there are 7 million AIS measurements, divided into 39077 different trajectories. These trajectories are the inputs of the segmentation stage, which results in the number of segments shown in Fig. 5. It also shows a demonstration of the imbalance problem, being possible to see the difference between the fishing class and the remaining instances (non-fishing).

As mentioned, to analyse the results of the different experiments carried out, the accuracy and F-measure are displayed together as a multi-objective problem, considering the total accuracy and the problem imbalance problem at the same time. In Fig. 6, it can be seen the distribution of values of the accuracy and F-measure corresponding to different variations of the classification and balancing algorithms. The Pareto front is formed for those non-dominated solutions, i.e., those with no other solutions with higher values in the two metrics simultaneously. In it, this front is formed by the solutions appearing in the upper-right corner.

It can be appreciated how the SVM has results that are usually better with respect to accuracy, but in return it may have a worse performance when considering the class imbalance. That effect is produced because it is a boundary-based algorithm and has a trend to misclassify the minority class if it has a low impact in the total accuracy. This is especially noticeable in the imbalanced classification, which shows in many cases a zero value for F-measure (i.e., all samples of the minority class misclassified).

The decision trees have more moderate results, which do not stand out so much in the accuracy but in return they get better results in the F-measure. However, the front is clearly dominated by the SVM with balanced data sets, these although still have executions that demonstrate little success in the problem of the imbalance but also have the executions located in the front.

Also, as can be seen in the points highlighted in white in Fig. 6, the original segmentation algorithm (Uniform sampling) used in previous works is behind the marked Pareto front (blue line). This implies that more advanced segmentation algorithms can provide better results, although it does not detract from the fact that some configurations of these algorithms have worse results. This only implies that it is important to find the correct configuration of the algorithms to use.

The most notable results are the SVMs that operate on a balanced data set using SMOTE, although the random undersampling also have Pareto front executions. To put the results in perspective, Fig. 7 shows all the segmentation algorithms executed by SVM applied on the SMOTE balanced data set. It not only shows the results of the accuracy but also the results for the F-measure which is not so positive since the most complex segmentations usually have slightly lower results in that metric.

There is no case that stands out especially from the rest, since when talking about a multi-objective problem between unbalance metrics and classification accuracy there is no algorithm that is especially good in both.

Being a point to emphasize that the best algorithms in one of the objectives clearly obtain their improvement when getting worse in the other one, an example would be the SQUISH-E with 20 error value and 5 compression parameter that obtains the best accuracy although its metrics are far below other algorithms. There is also the opposite case with the opening window algorithm, in which the best F-measure show an accuracy 20 points below that obtained by the specified SQUISH-E.

Regarding the higher complexity of the segmentation algorithms, we can see how generally the segmentation algorithms that give better results when performing the compression of trajectories (SQUISH-E, MRPA, DOTS) do not ensure a better result within the proposed classification problem. Most of their executions seem to have good accuracy but not all of them good results in the F-measure used for the imbalance problem. In fact, one of the results belonging to the front and that therefore could be considered as one of the best, is obtained by the most basic segmentation algorithm, the Uniform sampling with a size of 50.

Another aspect to consider is that the parameters introduced in the different segmentation algorithms influence the results variation, since the different executions of the same algorithm show very different results. For example, with the SQUISH-E algorithm, is possible to observe different results: one with the best accuracies, other with very poor results and another clearly within the Pareto front, achieving one of the best values within the two objectives with an accuracy close to 90% and balancing metrics only about 10 points below the best. Even if there is no absolute solution that meets the two proposed objectives, there is a set of solutions located on the Pareto front that are valid solutions, being better in one or the other objective.

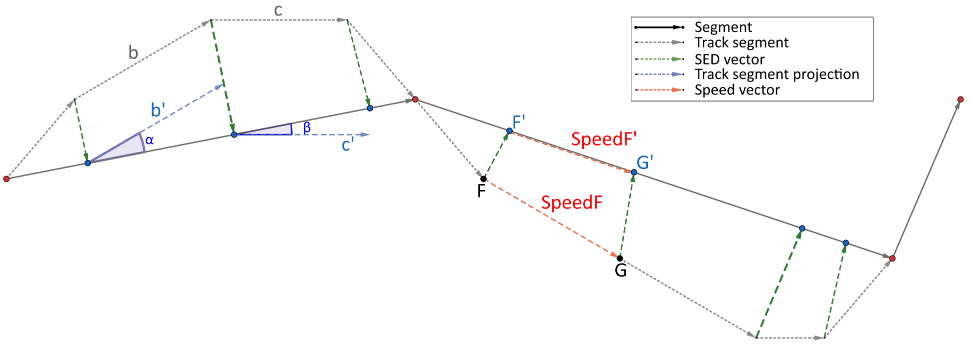


Fig. 4. Metrics example.

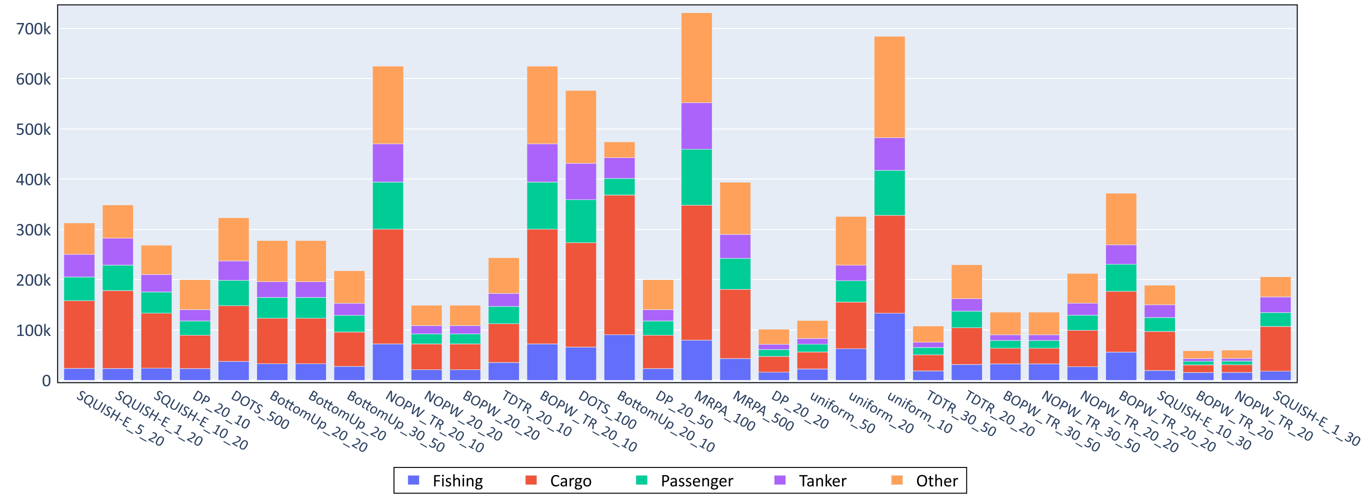


Fig. 5. Number of segments of the main ship types.

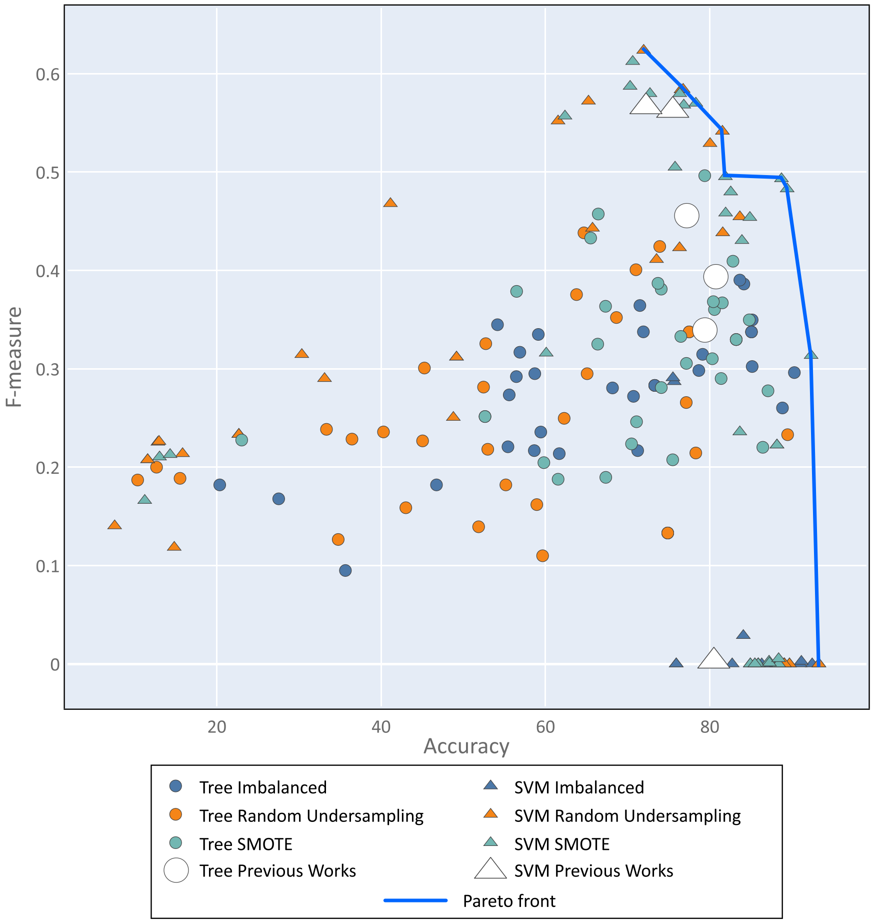


Fig. 6. Classification results for the different proposed variations.

5.2. *Segmentation algorithms study*

To evaluate specifically the segmentation algorithms two analysis are proposed.

On the one hand, a series of metrics are extracted from the segmented trajectories, making a comparison between the segmentation algorithms of the Pareto front and between the classification problem classes, fishing and not-fishing.

On the other hand, a series of representative trajectory are studied to show differences between the simplest segmentation, Uniform sampling, and an advanced segmentation, SQUISH-E.

As metric evaluation, the PED and SED distances are used comparing the point motion information with its projection in the segment. In specific 3 motion parameters are evaluated:

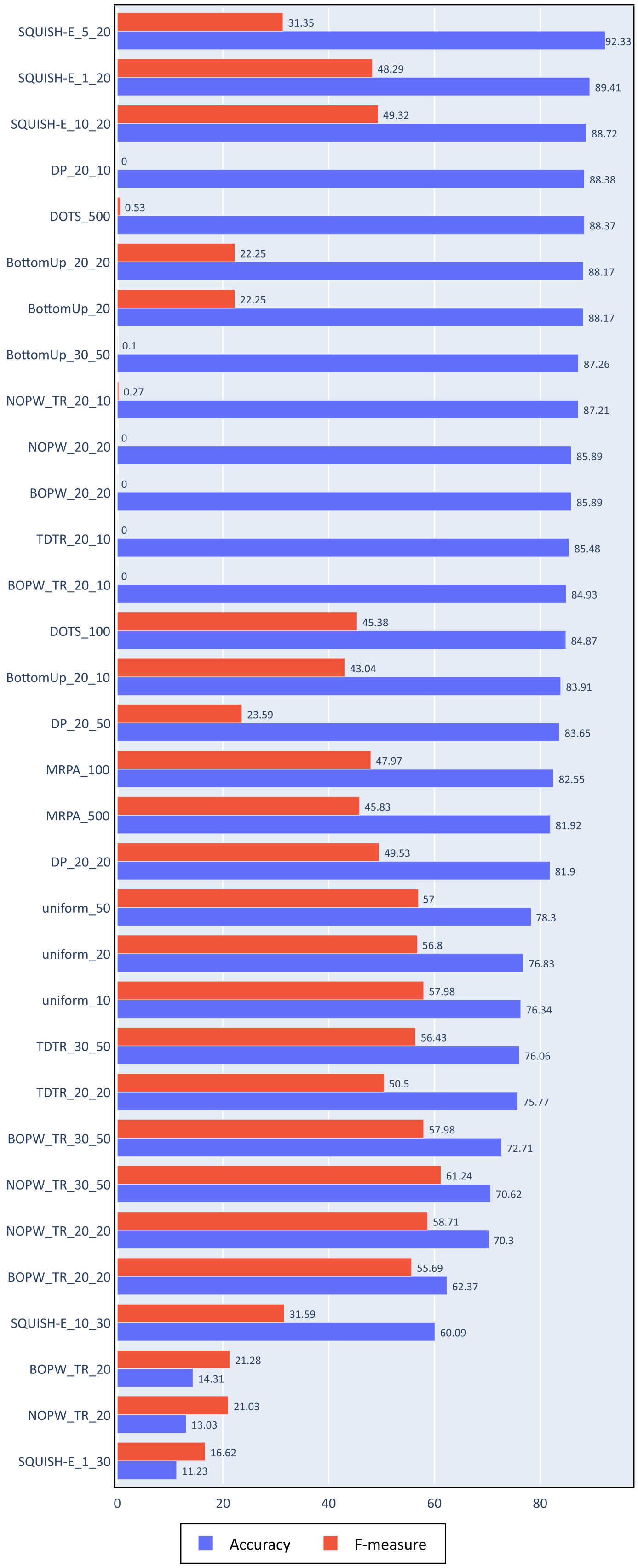


Fig. 7. SVM classification result for the segment variation in SMOTE balanced dataset.

- The position through the maximum error of both distances, PED and SED, has the distance between the trajectory point and its projection into the segment is a measure of the position difference.
- The mean angle difference between the trajectory points and the SED projection segment point. PED results are not shown because are similar to SED ones.
- The mean speed different between the trajectory point and its projection in the segment. Both PED and SED are analysed as there are differences.

In addition, the average number of points per segment and the compression rate are also analysed.

Although other metrics have been studied to assess segmentation, as it can be seen in the explanation of the previous section, in this section we only show the ones indicated as they have shown the most relevant information.

To analyse those metrics, a series of box-plot diagrams are used. Allowing the comparison between the segmentation algorithms results and a clear view of the metric distribution along all the texted trajectories.

For all the analysis at first the general segmentation results are exposed, then the individual results for each class of the classification stage, fishing and not fishing, are analysed. Also, the analysis is explained over the segmentation algorithms implementations that appear int the Paretoś front of the classification.

5.3. General analysis

5.3.1. PED / SED distance

With respect to the distance metrics, this analysis uses the maximum distance error, as it represents the worst case of each segment. The first thing that can be appreciated in Fig. 8 is that PED clearly have smaller values than SED. This is normal since PED distance measures the closest point of the segment perpendicularly while SED deviates from the perpendicular according of the track point time.

That said, it is possible to look at the differences of the compared algorithms. Uniform sampling shows little error despite cutting without any intelligence, although this is due to cutting every few points. This means that the error cannot grow sufficiently. The opposite is the case with OPW-TR, which, as it has a high minimum number of measurements, shows more error than the rest of the algorithms compared. It is important to note that the versions of the algorithms chosen for comparison are those within the Pareto front, and that other versions of the algorithms would show a completely different segmentation result.

Looking at the algorithms that cut most intelligently, SQUISH-E shows a scale where the error increases with different implementations of the algorithms. This scale is logical since the error increases as the compression ratio increases, which implies that as more segments are made with more points, these segments accumulate a higher error.

The graph-based algorithms, DOTS and MRPA, show a lower error, the main difference being that these two algorithms use an accumulation of the error when performing the segmentation, which implies that they segment earlier than SQUISH-E, reaching a lower error. A remarkable feature is that these two algorithms do not show outliers that deviate to a large extent with respect to the rest of the measures, which is the case with the rest of the segmentation algorithms.

Comparing the algorithms for both classes, fishing, and non-fishing focusing on SED distance (Fig. 9), the results are in line with the general results. It is noteworthy that the error of Fig. 8 the fishing class is lower than that of the non-fishing class, especially with OPW-TR whose maximum SED is greatly increased with respect to that obtained for fishing vessels. This is because non-fishing vessels are a conglomerate of different vessels, which implies a greater variation of trajectories and consequently a greater error. In addition, the trajectories of this class are generally longer than those of the fishing class, which can also lead to an increase in error due to the required compression ratio increase.

5.4. Number of points per segment and compression rate

In Fig. 10 is possible to see how Uniform sampling entries produce a constant segment size, as its segments are created on a fixed size. The OPW-TR has a minimum number of points per segment, which means that its segments are larger and consequently there is a higher compression ratio than in the rest of the algorithms.

Is remarkable, the advanced algorithms also in Fig. 10 show a similar average, with approximately 10 points per segment. This value is the one used by the Uniform sampling entries on the Pareto front, so it can be assumed that a segment size around this value seems to give good classification results.

If the compression rate of each of the classes in Fig. 11 is observed, the advanced algorithms that are influenced by the size of the segment to be generated have a higher value for the fishing class, especially the SQUISH-E algorithm. One effect of this feature is that the fishing SQUISH-E tends to have measurements centred on a zone that allows a higher compression rate. Within the SQUISH-E algorithm, it can also be observed that variations of the algorithm in fishing generate measurements that are more similar to each other than in non-fishing.

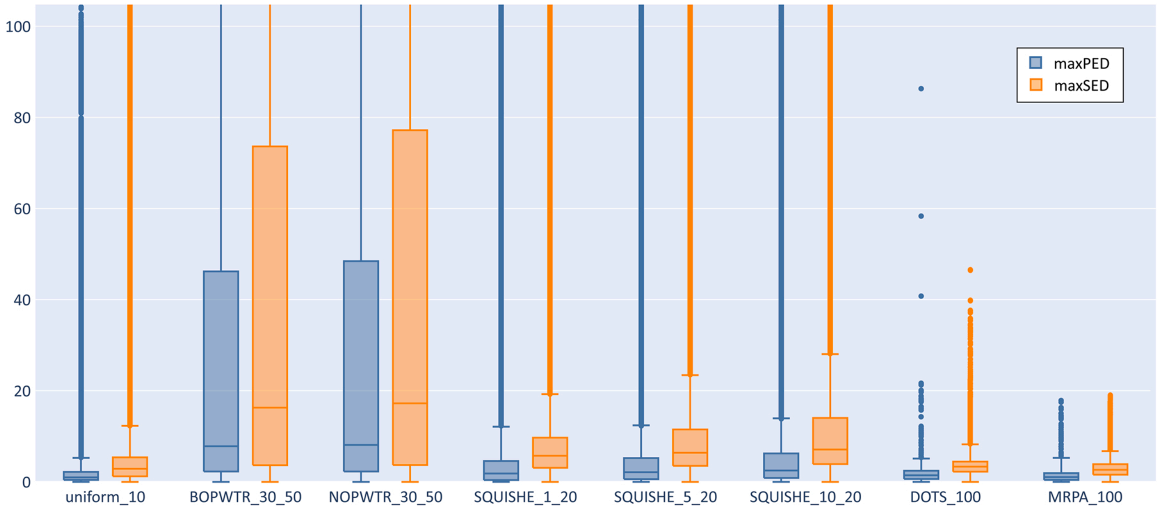


Fig. 8. Maximum PED and SED for all segments.

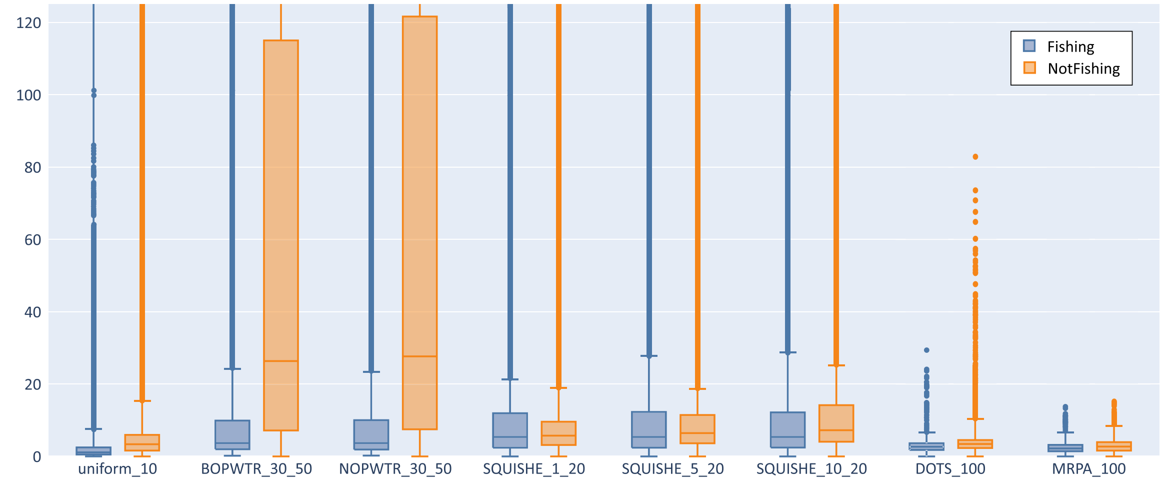


Fig. 9. Maximum SED distance for fishing and non-fishing segments.

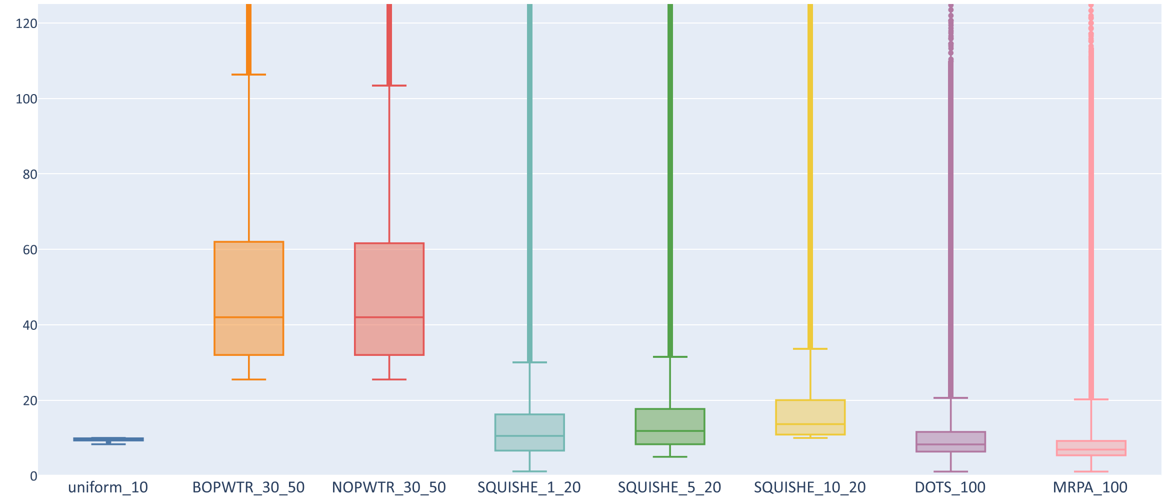


Fig. 10. Average number of points per segments.

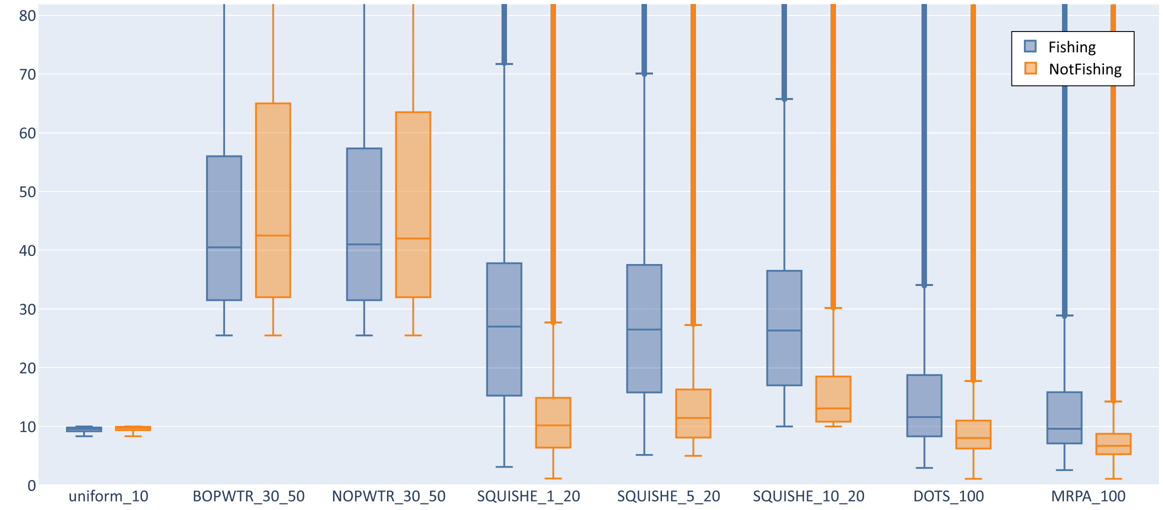


Fig. 11. Compression rate for fishing and non-fishing segments.

5.5. Average angle difference

In general, it can be seen in Fig. 12 that the most intelligent segmentation algorithms, SQUISH-E, MRPA and DOTS, have smaller angle differences. This indicates that these algorithms consider the dynamics of the ship when performing the segments, since the increase of the error when there are turns indicates the existence of a new segment.

Looking at the difference between classes in Fig. 13, fishing vessels have a greater difference in angle, which may be indicative of a greater variation in their trajectories.

5.6. Average speed difference

Finally, with respect to the variation of speed, the most remarkable thing is that the input data belong to ship trajectories, which implies little variation due to the movement of the ships. However, between SED and PED (Fig. 14) SED has a higher value. This is again because the SED measurement is more informed as it is adjusted with respect to the time of the point, instead of being set perpendicularly.

Again, if the classes are observed (Fig. 15), fishing has a smaller speed difference than non-fishing, this indicates that the distance travelled will be shorter for fishing vessels and that fishing vessels have a slower movement.

5.7. Segmentation illustrative examples

5.7.1. Fishing ship

One common situation is the one shown in Fig. 16. As it can be seen, the trajectory shows a fishing ship entering the port which implements two rectilinear movements at the approach, with a little course variation in between, and a manoeuvre segment in the port. An advanced algorithm could take this information and make more informed segments.

In the left, the original and simplest Uniform sampling algorithm, is applied. Its segments are less informed as the cuts are applied without taking into consideration the ship movement. In contrast, In the right, an advanced segmentation as SQUISH-E algorithm creates three segments, one for each defined movement.

Analysing the features extracted of the segments, is possible to see kinematic parameters like the course variation represented in Fig. 18 that show an increasing average value as the segment increase the manoeuvre. The speed modulus represented in Fig. 17 shows a decreasing value as it approaches the port, well divided by both algorithms.

As it can be seen in Fig. 17, the features also show variation although this segmentation technique show more segments with the same measures instead of one segment that summarized the movement.

This does not necessarily have to be a bad quality of the segmentation, because as can be seen the manoeuvring part is more detailed with these shorter segments. On the other hand, however, more noise is introduced into the classification problem. The solution is to find algorithms that maintain a balance between detailing the parts of greatest interest and summarising those that do not provide information.

It is noteworthy that the Uniform sampling algorithm loses information at the end of the segments, when the last one does not reach the minimum size set, in this case, 10 measurements.

5.8. Non-fishing ship

Other interesting movement is the one of Fig. 19, which show a long-distance movement in the middle of the sea for a non-fishing ship (in specific is a cargo ship). As it can be seen, it shows also three differentiate movements: two rectilinear movements separated from a manoeuvre movement in between.

As it can be seen an advanced algorithm like SQUISH-E creates three segments (see right), and Uniform sampling algorithm gets much more segments (see left).

If the features are analysed (see Fig. 20 and Fig. 21) is possible to see how features like speed variation or course variation are quite well represented and partitioned in SQUISH-E, with the exception of the middle manoeuvre movement that show a variation that differentiates the segment.

This excessive segmentation that Uniform sampling algorithm generates, increases a possible bad performance, introducing to the classification process multiple equal features that could create misinformation or overfitting to the training dataset.

6. Conclusions and perspectives

In the study, the impact of segmentation on the classification results have been analysed, being possible to appreciate as the most advanced algorithms usually provide better results in accuracy objective. However, the segments provided by these algorithms do not ensure good results in the second objective proposed, which is related to the performance with the minority class, due to the high imbalance in the data set. That said, the results show a Pareto front with different solutions that work for the two objectives imposed within the multi-objective problem.

As a conclusion, it is very important the quality of the segments

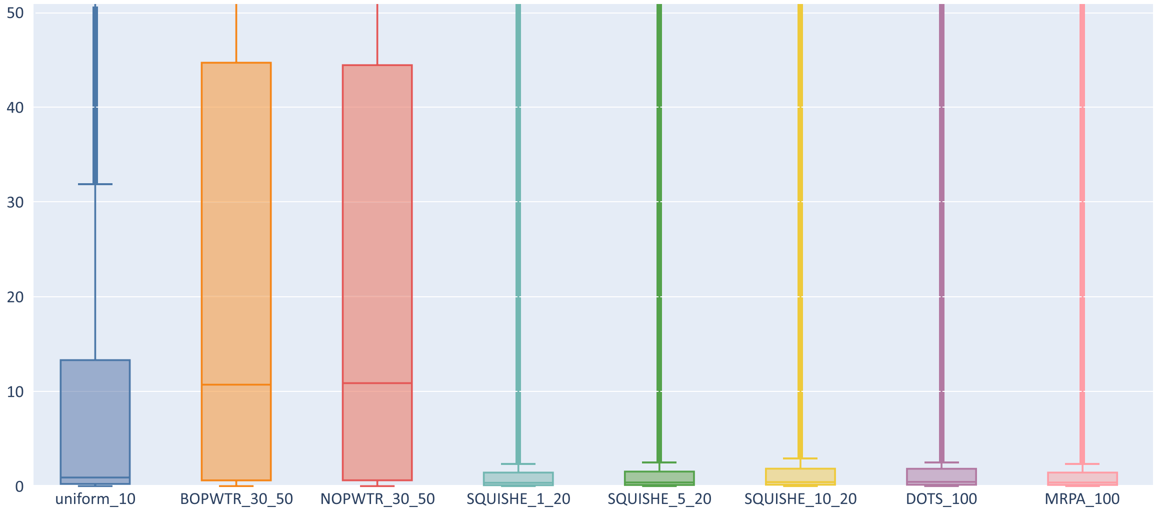


Fig. 12. Average angle difference in SED for all segments.

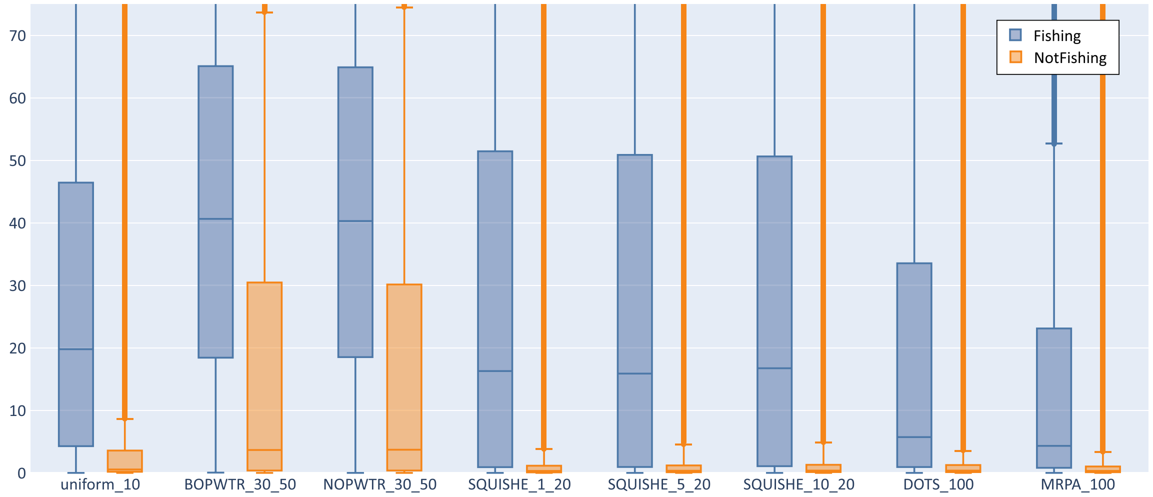


Fig. 13. Average angle difference in SED for fishing and non-fishing.

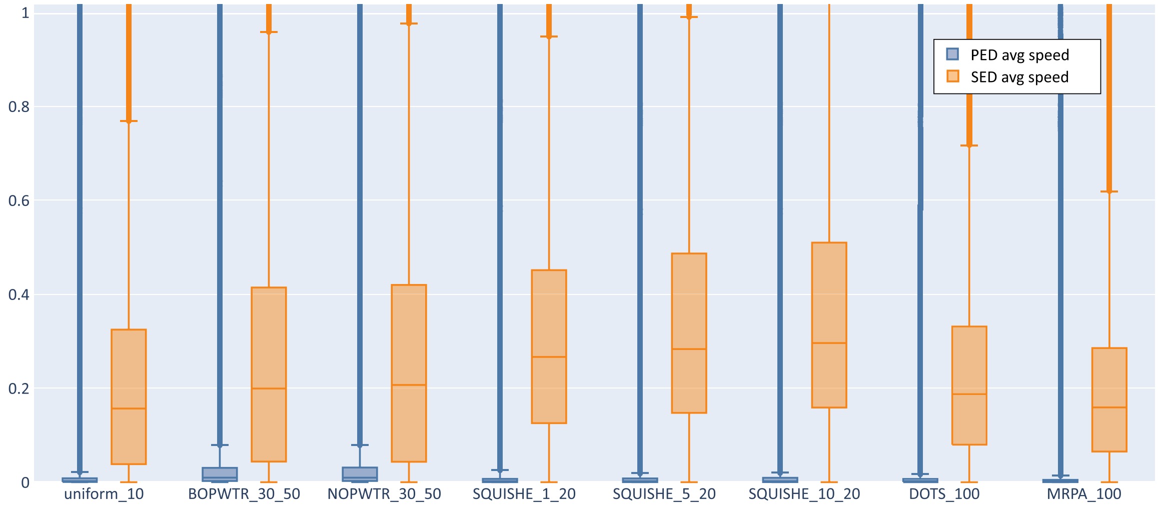


Fig. 14. Average speed difference in PED and SED for all segments.

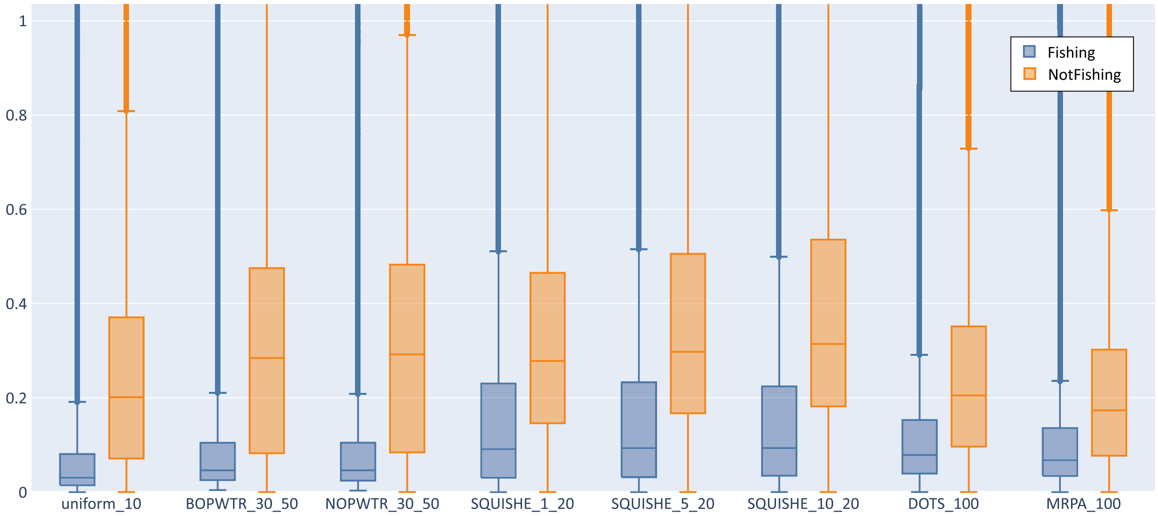


Fig. 15. Average speed difference in SED for fishing and non-fishing.

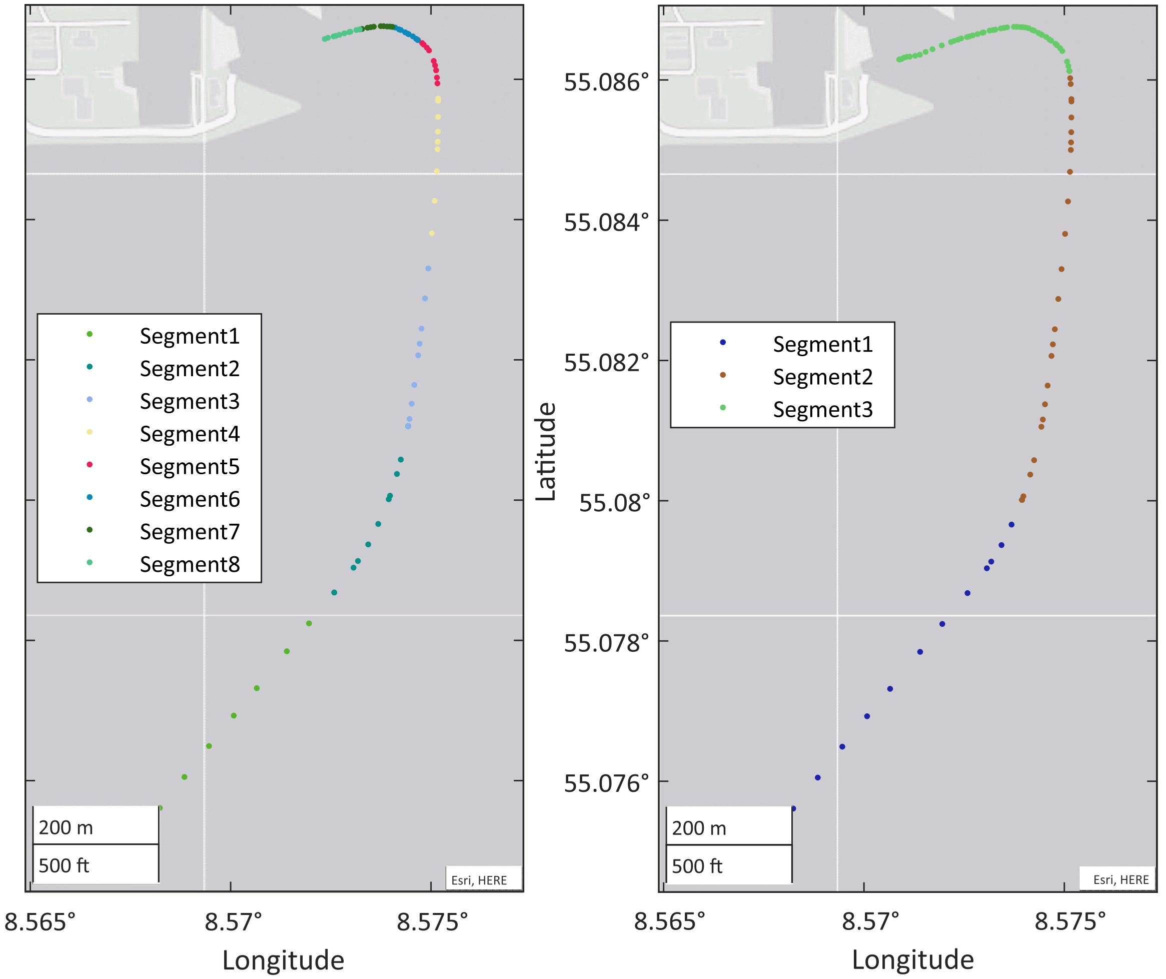


Fig. 16. Fishing segments with Uniform (left) and SQUISH-E (right).

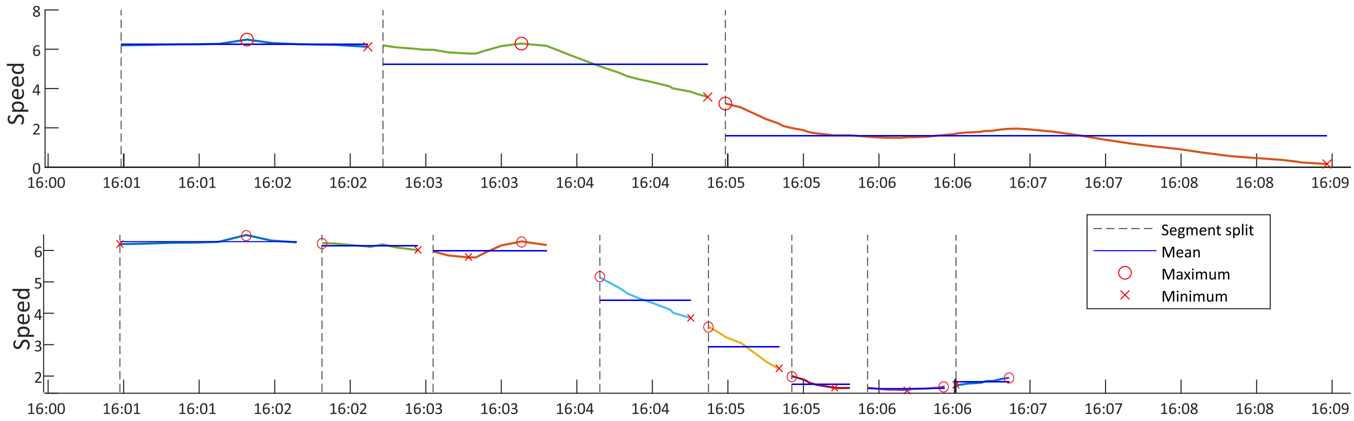


Fig. 17. Speed comparison between Fishing segments with SQUISH-E (top) and Uniform (bottom).

within the proposed process since there are trajectories with more measurements than others which create more segments with certain segmentation algorithms, affecting the classification. Also, by classifying segments it is possible to introduce noise with non-representative segments to its class (e.g., a ship departing from a port), or overfitting a specific class to a non-representative type of dynamics. Finding the kind of segments that each segmenter generates, as well as those prioritised by each classifier, is further research that should be carried out to assure that a proper solution is adopted.

The SVM algorithm has demonstrated that it has the capacity to obtain good results for the classification, however it has a clear tendency towards the trivial solution, harming the minority class to obtain good results when maximizing the majority class.

Both classification algorithms are representative and responsive to the analysed balancing algorithms. The main point of improvement for the future will be to test new classification algorithms, that achieve a better separation of instances, particularly those that can benefit most from the segments. Also, the application of the proposed method can

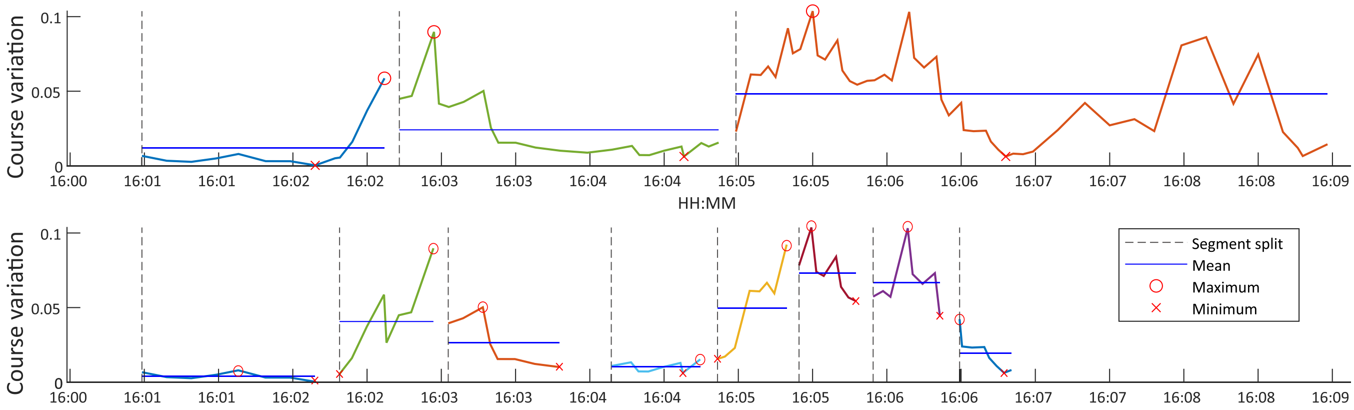


Fig. 18. Course variation comparison between Fishing segments with SQUISH-E (top) and Uniform (bottom).

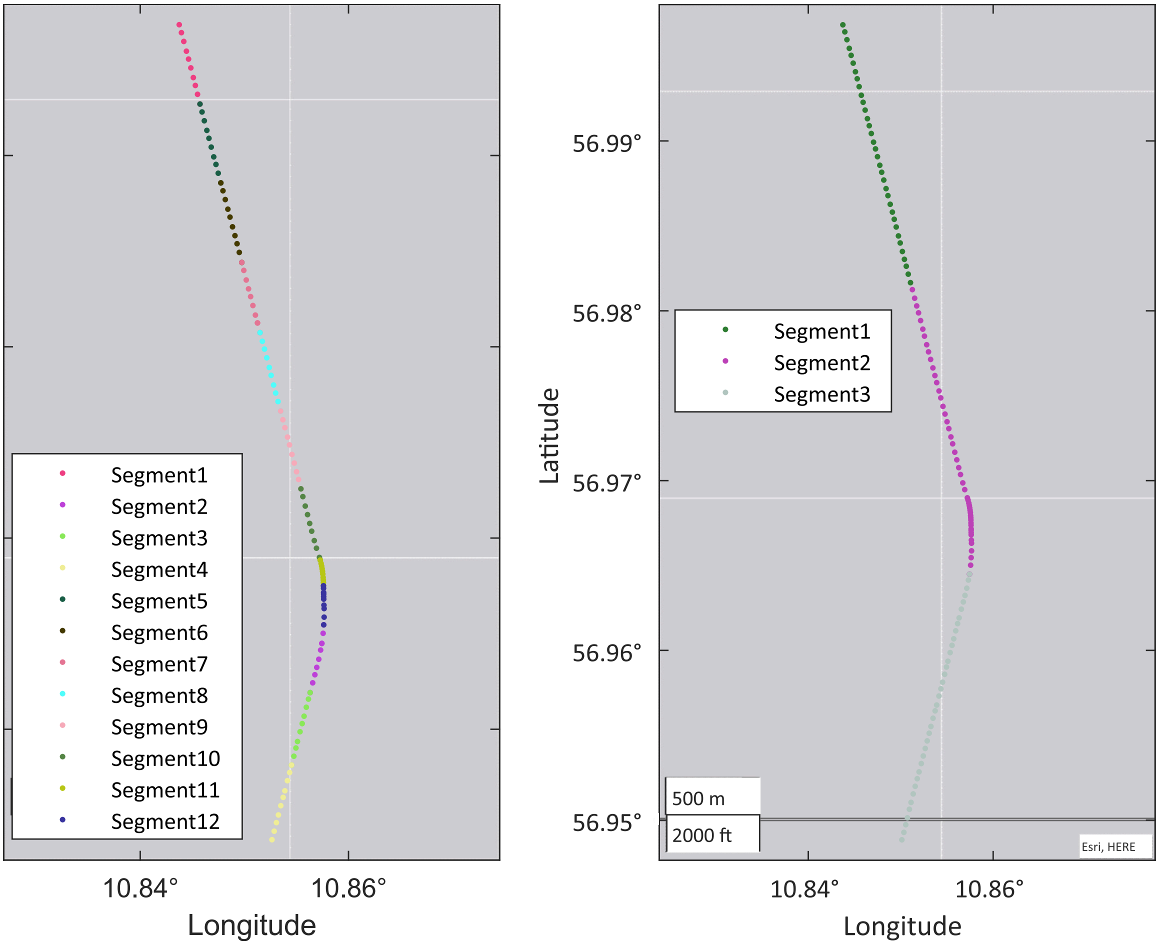


Fig. 19. Example non-fishing trajectory with SQUISH-E algorithm.

approach other similar problems where classification is performed based on kinematic information of trajectories. For example, a classification oriented on pedestrian traffic could ensure safety (pickpocket identification), or the application in air traffic can allow flying mode identification thanks to the track segments adaptability.

From the point of view of the segmentation algorithms, it has been observed that the more advanced algorithms show more informed segments of the ship motion dynamics. However, it is also possible to obtain good results with less advanced algorithms if they are properly configured.

The type of movement made in the trajectories also has a strong influence on the result, as long movements with little variation can damage some segmentation algorithms, generating more segments than desired.

The error in the accuracy of the segmentation algorithms is lower in the more advanced ones, although a low error can be obtained with simpler algorithms. In addition, the average compression ratio of the Pareto front algorithms is in the same area, indicating that the most informed segments for the problem need a compression ratio in this area.

This work raises potential areas for future work, both in the field of trajectory classification and segmentation. The most obvious one is to

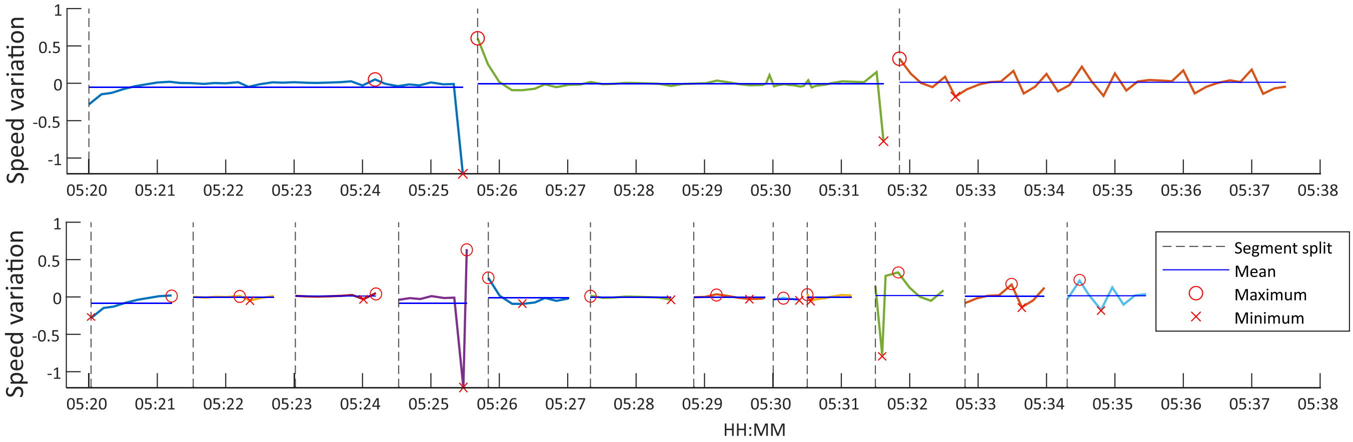


Fig. 20. Speed variation comparison between Non-fishing segments with SQUISH-E (top) and Uniform (bottom).

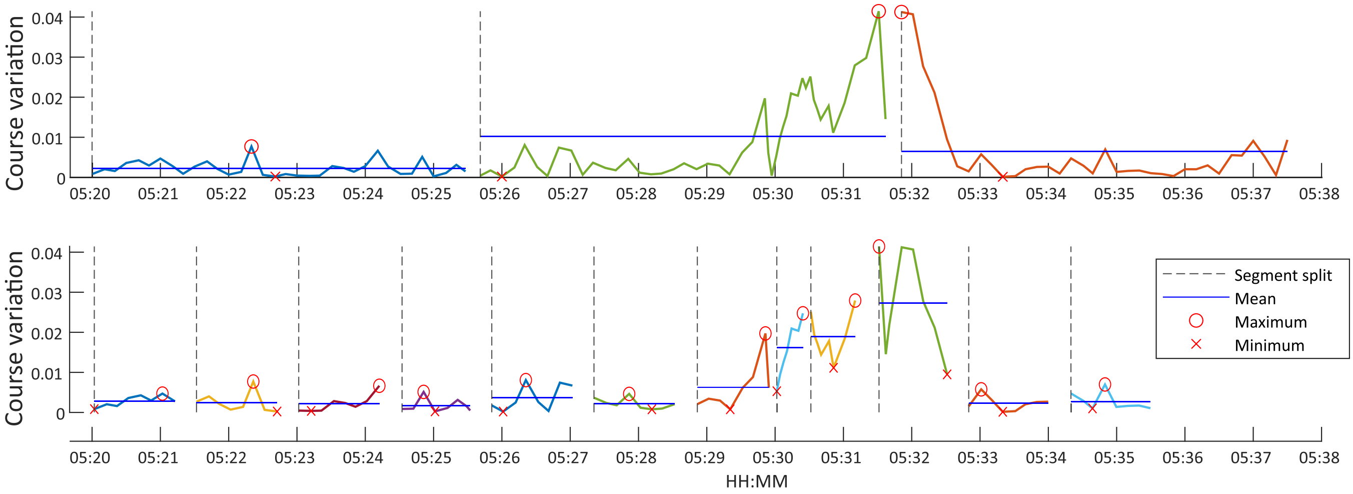


Fig. 21. Course variation comparison between Non-fishing segments with SQUISH-E (top) and Uniform (bottom).

analyse in more detail, as in this work, other components of the framework, such as data cleaning or classification processes, to improve the overall result. Additionally, in segmentation, to study other algorithms focused on vehicle dynamics rather than on the trajectory shape, or to automatically find the ideal parameters of each segmenter for the problem.

CRediT authorship contribution statement

Daniel Amigo: Visualization, Writing – review & editing, Writing – original draft, Data curation, Resources, Formal analysis, Validation, Software, Methodology, Conceptualization. **David Sánchez:** Visualization, Writing – original draft, Data curation, Formal analysis, Validation, Software. **José Manuel Molina:** Funding acquisition, Project administration, Supervision, Investigation, Validation, Conceptualization. **Jesús García:** Funding acquisition, Project administration, Supervision, Investigation, Formal analysis, Validation, Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by public research projects of Spanish Ministry of Economy and Competitivity (MINECO), reference TEC2017–88048-C2–2-R and by the Madrid Government under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17), and in the context of the V PRICIT (Regional Program of Research and Technological Innovation).

References

[1] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, G.B. Huang, Exploiting AIS data for intelligent maritime navigation: a comprehensive survey from data to methodology, IEEE Trans. Intell. Transp. Syst. 19 (5) (2018) 1559–1582, https://doi.org/10.1109/TITS.2017.2724551.

[2] D. Amigo, D. Sánchez Pedroche, J. García, and J.M. Molina, "AIS trajectory classification based on IMM data," in *2019 22th international conference on information fusion (FUSION)*, Ottawa, ON, Canada, Jul. 2019, pp. 1–8.

[3] D. Sánchez Pedroche, D. Amigo, J. García, and J.M. Molina, "Context information analysis from IMM filtered data classification," in 1st Maritime Situational Awareness Workshop (MSAW), Lerici, Italy, Oct. 2019.

[4] D. Sánchez Pedroche, D. Amigo, J. García, J.M. Molina, Architecture for trajectory-based fishing ship classification with AIS data, Sensors 20 (13) (2020) 3782, https://doi.org/10.3390/s20133782.

[5] D. Amigo, D. Sánchez Pedroche, J. García, and J.M. Molina, "Segmentation optimization in trajectory-based ship classification," in *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO)*, Burgos, Spain, 2020, p. 10.

[6] P. Kraus, C. Mohrdieck, and F. Schwenker, "Ship classification based on trajectory data with machine-learning methods," in *2018 19th International Radar Symposium (IRS)*, Bonn, Jun. 2018, pp. 1–10. doi: ⟨10.23919/IRS.2018.8448028⟩.

[7] T. Zhang, S. Zhao, J. Chen, Research on Ship Classification Based on Trajectory Association, in: C. Douligeris, D. Karagiannis, D. Apostolou (Eds.), Knowledge Science, Engineering and Management, vol. 11775, Springer International Publishing, Cham, 2019, pp. 327–340, https://doi.org/10.1007/978-3-030-29551-6_28.

[8] S. Ichimura and Q. Zhao, "Route-Based Ship Classification," in *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST),* Morioka, Japan, Oct. 2019, pp. 1–6. doi: ⟨10.1109/ICAwST.2019.8923540⟩.

[9] K. Sheng, Z. Liu, D. Zhou, A. He, C. Feng, Research on ship classification based on trajectory features, J. Navig. 71 (1) (2018) 100–116, https://doi.org/10.1017/S0373463317000546.

[10] Y. Zheng, Trajectory data mining: an overview, ACM Trans. Intell. Syst. Technol. 6 (3) (2015) 1–41, https://doi.org/10.1145/2743025.

[11] W.R. Tobler, Numerical map generalization, Mich. Inter-Univ. Community Math. Geogr. (1966).

[12] N. Meratnia, A. Rolf, Spatiotemporal compression techniques for moving point objects (no), Lect. Notes Comput. Sci. (2004), https://doi.org/10.1007/978-3-540-24741-8.

[13] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a line or its caricature, Can. Cartogr. 10 (1973) 112–122, https://doi.org/10.3138/FM57-6770-U75U-7727.

[14] N. Meratnia, R.A. de By, A new perspective on trajectory compression techniques, Int. Soc. Photogramm. Remote Sens. ISPRS (2003) 8.

[15] J. Muckell, P.W. Olsen, J.-H. Hwang, C.T. Lawson, S.S. Ravi, Compression of trajectory data: a comprehensive evaluation and new approach, GeoInformatica 18 (3) (2013) 435–460, https://doi.org/10.1007/s10707-013-0184-0.

[16] Minjie Chen, Mantao Xu, P. Franti, A fast O(N) multiresolution polygonal approximation algorithm for GPS trajectory simplification, IEEE Trans. Image Process 21 (5) (2012) 2770–2785, https://doi.org/10.1109/TIP.2012.2186146.

[17] W. Cao, Y. Li, DOTS: An online and near-optimal trajectory simplification algorithm, J. Syst. Softw. 126 (2017) 34–44, https://doi.org/10.1016/j.jss.2017.01.003.

[18] Danish Maritime Authority, "AIS Data." dma.dk/SikkerhedTilSoes/Sejladsinformation/AIS/Sider/default.aspx.

[19] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, Sep. 2017, pp. 79–85. doi: ⟨10.1109/ICACCI.2017.8125820⟩.

[20] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357, https://doi.org/10.1613/jair.953.

[21] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera, Learning from Imbalanced Data Sets, Springer International Publishing, Cham, 2018, https://doi.org/10.1007/978-3-319-98074-4.

Daniel Amigo is a PhD student at Universidad Carlos III de Madrid, in Computer Science and Technology. In 2019 he completed a double Master's degree in Computer Engineering and Computer Science and Technology and a bachelor's degree in Computer Engineering in 2017 also at Universidad Carlos III de Madrid. He is involved in several related research projects on traffic tracking and surveillance. His current lines of research are the compression and segmentation of trajectories, the geolocation of real-world objects and its conversion into virtual environments for simulation tools.

David Sánchez Pedroche has a pre-doctoral contract at the Universidad Carlos III de Madrid. He joined the Department of Computer Science at Universidad Carlos III de Madrid as part of the Applied Artificial Intelligence Group (GIAA) in 2018. His current research focuses on the application of data fusion techniques, radar data processing and detection systems, air and maritime traffic management, and UAV intelligence. He graduated in Computer Engineering in 2017 with a double Master's degree in Computer Engineering and Computer Science and Technology in 2019, both from Universidad Carlos III de Madrid.

Jesús García is full professor at the Universidad Carlos III de Madrid. He joined the Computer Science Department of the Universidad Carlos III de Madrid in 1993. Currently he coordinates the Applied Artificial Intelligence Group (GIAA). His main research interests are computational intelligence, sensor and information fusion, machine vision, traffic management systems and autonomous vehicles. Within these areas, including theoretical and applied aspects, he has co-authored more than 10 book chapters, 60 journal papers and 180 conference papers.

José M. Molina is full professor at the Universidad Carlos III de Madrid. He joined the Computer Science Department of the Universidad Carlos III de Madrid in 1993. Currently he coordinates the Applied Artificial Intelligence Group (GIAA). His current research focuses on the application of soft computing techniques (NN, Evolutionary Computation, Fuzzy Logic and Multiagent Systems) to radar data processing, air traffic management, e-commerce and ambient intelligence. He has authored up to 100 journal papers and 200 conference papers. He received a degree in Telecommunications Engineering in 1993 and a PhD degree in 1997 both from the Universidad Politécnica de Madrid.