Fine-Tuned Compressed Representations of Vessel Trajectories

Giannis Fikioris
NCSR Demokritos
Athens, Greece
gfikioris@iit.demokritos.gr

Kostas Patroumpas
Athena Research Center
Athens, Greece
kpatro@athenarc.gr

Alexander Artikis
University of Piraeus
NCSR Demokritos
Athens, Greece
a.artikis@unipi.gr

Georgios Paliouras
NCSR Demokritos
Athens, Greece
paliourg@iit.demokritos.gr

Manolis Pitsikalis
University of Liverpool
Liverpool, UK
E.Pitsikalis@liverpool.ac.uk

ABSTRACT

In the maritime domain, vessels typically maintain straight, predictable routes at open sea, except in the rare cases of adverse weather conditions, accidents and traffic restrictions. Consequently, large amounts of streaming positional updates from vessels can hardly contribute additional knowledge about their actual motion patterns. We have been developing a system for vessel trajectory compression discarding a significant part of the original positional updates, with minimal trajectory reconstruction error. In this work, we present an extension of this system, that allows the user to fine-tune trajectory compression according to the requirements of a given application. The extended system avoids the issues of hyper-parameter tuning, supports incremental optimization and facilitates composite maritime event recognition. Finally, we report empirical results from a comprehensive empirical evaluation against two real-world datasets of vessel positions.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**.

KEYWORDS

AIS, genetic algorithm, maritime data analytics, trajectory

ACM Reference Format:
Giannis Fikioris, Kostas Patroumpas, Alexander Artikis, Georgios Paliouras, and Manolis Pitsikalis. 2020. Fine-Tuned Compressed Representations of Vessel Trajectories. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412706

1 INTRODUCTION

As maritime traffic is steadily increasing to support global supply chains, merchandise trade and transportation, protection of the ocean environment and its resources becomes imperative, as well as safety and security in maritime navigation. Over the past two decades, the *Automatic Identification System* (AIS) has provided a powerful means to track vessels across the seas, thus supporting online maritime monitoring, as well as enabling reduction of fuel consumption to improve vessel efficiency. The collected AIS raw tracking data is valuable, as it includes unique identification of vessels, their position, course, speed, etc. However, processing in real-time large amounts of streaming AIS positional updates at global scale is a huge burden in terms of processing and storage. Typically, stakeholders 'decimate' the AIS data to be stored for analysis, by selectively or randomly dropping a significant percentage of the relayed AIS messages. The rationale is that successive locations emitted every few seconds from each vessel can hardly contribute additional knowledge about their actual motion patterns.

In [16], we proposed a maritime surveillance system for *trajectory detection* from AIS positions enabling online maintenance of compressed representations of their evolving traces. Instead of randomly discarding incoming AIS positions, this module judiciously picks selected *critical points* along each trajectory, e.g., indicating stops, turning points, changes in speed, etc. In contrast to typical trajectory simplification, not only can this module provide compressed, lightweight trajectories for further analysis, but it also annotates important *mobility events* through those chosen critical points. The resulting *trajectory synopses* considerably reduce the data volume to be stored, with a tolerable reconstruction error.

Nevertheless, such trajectory compression is very sensitive to *parametrization*. Every incoming AIS position must be checked against several spatio-temporal conditions, e.g., to identify any significant deviations in speed, heading, acceleration, etc. with respect to the known motion status of the respective vessel. If threshold values for such deviations are not suitable for the AIS data at hand, this can deteriorate the quality of the synopses or increase the amount of retained critical points. In fact, AIS data exploration with the advice of maritime domain experts seem indispensable for the selection of suitable parameter values. As this pre-processing strongly depends on data characteristics (e.g., frequency of positional updates, spatial extent of the monitored area, number and type of moving vessels, etc.), it must be repeated for each new dataset almost from scratch. Finally, this trajectory compression used to apply the same parametrized settings for all vessels, although they may differ in type, tonnage, length, etc., and hence in their motion patterns.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412706

In this work, we present a system to automatically adapt the parameter values of trajectory compression. First, we take into account the type of each vessel (passenger, cargo, fishing, etc.) in order to choose a suitable configuration that can yield improved trajectory synopses, both in terms of approximation error and compression ratio. Second, we employ a genetic algorithm that iterates over several combinations of the parameter values until converging to a fine-tuned configuration per vessel type. We use an optimization function that does not require hyper-parameter tuning, and thus we avoid the computational overhead and accuracy issues of this process. Third, our system supports incremental optimization, by training in data batches, and therefore continuously improves performance. Fourth, our system is integrated with a composite event recognition engine, to efficiently detect complex maritime activities, such as ship-to-ship transfer and loitering.

We report results from a comprehensive empirical evaluation on two real-world datasets. The first one is a publicly available dataset concerning vessel activity for 6 months around Brest, France. The second dataset contains vessel positions for 6 months across the entire Mediterranean Sea. Our tests confirm that compression efficiency is comparable or even better than the one with default parametrization, without resorting to a laborious data exploration. This also enhances composite event recognition, as it operates on fewer data points without compromising its predictive accuracy.

Our prototype system employs open-source software specialized in vessel trajectory summarization. This efficient, flexible, and robust software offers to AIS data stakeholders, such as vessel tracking companies, a powerful means to intelligently discard a large amount of originally relayed positions. Instead of the current practice that randomly eliminates positions to reduce the bulk of accumulated information, our proposed lightweight synopses can reliably reconstruct vessel trajectories with minimal error while keeping the least possible amount of incoming locations. Furthermore, annotations at those locations are valuable for more advanced processing, particularly in complex event recognition and real-time analytics. In the context of the INFORE project[1] that includes MarineTraffic[2], one of the largest vessel tracking companies, this system will be deployed against large-volume AIS data in the Mediterranean Sea and fused with other sources in order to enhance Situational Awareness in the maritime domain and to forecast critical events of interest in real-world conditions.

The remainder of this paper proceeds as follows. Section 2 surveys related work. Section 3 outlines the types of mobility events annotated in the trajectory synopses for vessels. Section 4 analyzes the suggested methodology involving a genetic algorithm for fine-tuning the parameters used in trajectory compression. Section 5 reports results from a comprehensive empirical study against two real-world AIS datasets. Finally, Section 6 summarizes the paper.

2 RELATED WORK

Our approach on trajectory synopses over streaming AIS positions involves a kind of *online* path simplification. Offline techniques like [12, 14] cannot apply, since complete trajectories must be available in advance. In contrast, we consider evolving trajectories where fresh locations are received online. Samples retained in the synopses should keep each compressed trajectory as much closer to the original one, as in fitting techniques [2, 13], which minimize approximation error. The one-pass approach in [13] discards points buffered in a sliding window until the error exceeds a given threshold. The notion of safe areas in [19] keeps samples that deviate from predefined error bounds regarding speed and direction. Dead-reckoning policies like [23] and mobility tracking protocols in [7] are usually employed on board of the moving objects, hence they do not seem applicable for AIS position reports. More recently, the focus on online trajectory simplification is focused on error bounds. For example, the ageing-aware approach in [10] uses a bounded quadrant mechanism to chose samples for the summary. Local distance checks and optimizations for higher compression are employed in [9]. A novel spatiotemporal cone intersection technique involving Synchronous Euclidean Distance is applied in [8] for more aggressive compression. A survey and empirical study of various trajectory simplification techniques is available in [24].

Although such generic online methods can provide reliable summaries over vessel trajectories, they entirely lack support for mobility-annotated features in the retained samples. The maritime surveillance platform in [16] aims at tracking vessel trajectories and also recognizing composite events, such as dangerous vessel activity. Its trajectory compression module applies a sliding window over the streaming positions, and periodically reports annotated "critical" points (stop, turn, speed change, etc.) to be retained in each vessel's synopsis. Empirical results show that less than 5% of the raw data suffice to offer reliable trajectory approximations. Further enhanced as a stream-based application [17], this has led to the *Synopses Generator* framework that detects mobility events with richer semantics (multiple annotations per location, more refined conditions) with minimal latency in modern cluster infrastructures [22]. With extra rules, the reported mobility events can also act as notifications to timely trigger detection of more *complex events* [4], analogous to those applied by CEP-traj [21] and RTEC [1, 18]. Still, all this requires careful parametrization of its various conditions, which we aim to fine tune in this work using a genetic algorithm.

Machine learning techniques have been applied against AIS data for various objectives. Indicatively, classifying vessels by type from raw AIS trajectories is suggested in [11]. Employing a recurrent neural network, the deep-learning scheme in [15] supports various tasks in maritime traffic surveillance, such as detection of abnormal behavior, trajectory reconstruction, vessel type identification, etc. In another direction, the tool proposed in [6] extracts features from large volumes of AIS data streams and employs a trained classifier to identify typical vessel activities related to fishing patterns (trawling, longlining). However, none of these techniques aims at online trajectory simplification as our proposed method.

3 VESSEL TRAJECTORY COMPRESSION

Next, we outline how the open-source Synopses Generator module detects mobility events along vessel trajectories and the parameters involved in such online processing of AIS positions. Details about the applied trajectory summarization can be found in [16, 17].

Upon arrival of a fresh AIS position, four attributes are extracted: *MMSI* (Maritime Mobile Service Identity) is used as vessel identifier,

[1] https://www.infore-project.eu/

[2] MarineTraffic.com

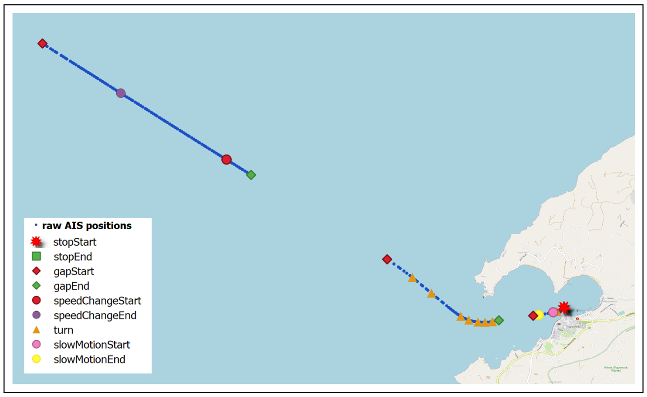


Figure 1: Critical points detected along a vessel trajectory.

Longitude and *Latitude* coordinates of the reported position, as well as its *Timestamp*. To reduce inherent *noise* in this data (such as duplicate or delayed messages, invalid coordinates, etc.) [5], several single-pass heuristics can be applied online, effectively discarding up to 20% of the original AIS positions without harming vessel monitoring and reconstruction of their trajectories [16].

As the resulting noiseless positions stream in, they are chronologically buffered in memory, one sequence per vessel. This sequence represents the *most recent portion* of each evolving trajectory composed of m 'clean', time-ordered locations (e.g., $m = 5$). Based on them, the *mean velocity* $\overrightarrow{v}_m$ of each vessel is estimated, as well as several derived spatiotemporal features (distance, travel time, overall change in heading, etc.). Since frequency of updates may differ among vessels, a maximum *historical timespan* ω is set to discard obsolete locations from the buffer, e.g., those arrived one hour before current timestamp τ_{now}.

The Synopses Generator applies single-pass heuristics for detecting mobility events with suitable parametrization (Table 1). Each such event is captured by one, two, or multiple *critical points* that are retained in the synopsis of each vessel trajectory. In particular:

- *Stop* indicates that a vessel remains stationary over a period of time by checking whether its instantaneous speed v_{now} is lower than a threshold v_{min} (e.g., 0.5 knots). The first and last locations in this subsequence are annotated as critical points and are kept in the synopsis. In case a fresh location is found more than D meters away the previous one, the stop event ends even if $v_{now} < v_{min}$.
- *Slow motion* is detected when a vessel moves at a speed less than a threshold v_θ (e.g., < 5 knots) over some time interval. The first and last point in this sub-trajectory are marked as critical.
- *Change in Heading*: If current heading deviates more than an angle $\Delta\theta$ (e.g., $> 4^o$) from mean velocity $\overrightarrow{v}_m$, a turning point is detected. Since vessels generally make smooth turns, multiple such points may be successively issued as critical ones.
- *Speed change* occurs when the rate of change for speed exceeds a given threshold α (e.g., 25%) with respect to its mean speed v_m over a recent time interval. The two locations marking the duration of this event are kept as critical.
- *Communication gaps* indicate that a vessel has not reported its location recently, e.g., in the past ΔT =10 minutes. The locations marking loss of contact and its restoration are the critical ones.

Figure 1 illustrates the types of detected critical points along an example vessel trajectory. It is important that detection of such mobility events keeps in pace with the streaming AIS positions, ideally identifying within milliseconds whether a fresh location is critical or not, and thus incrementally maintaining the trajectory synopses. Note that the actual course of a vessel can be approximated from its synopsis using time-based interpolation to estimate positions not retained as critical. Overall, the Synopses Generator can compress drastically the data volume, sometimes keeping even less than 1% of the raw AIS positions with tolerable error in the resulting approximation [16].

4 ADAPTING COMPRESSION PARAMETERS

Trajectory compression is very sensitive to *parametrization*. Table 1 presents the parameters and their default values, which have been set with the valuable advice of domain experts, specifically for an AIS dataset concerning vessel activity in Brest, France [20]. However, in any other AIS dataset, different values may be needed, depending on the geographic area, the sampling frequency, the types of monitored vessels, etc. Moreover, this kind of trajectory summarization applies the same parametrized conditions over all vessels in the data, irrespective of their type, tonnage, length, etc. This approach lacks flexibility and cannot cope with the varying mobility patterns of the vessels. For example, a larger ship takes turns more smoothly, while a Fishing or Tug Boat can make sharper turns. This would allow more relaxed (i.e., greater) angle threshold for larger ships without harming the quality of their synopses. On the other hand, having a stricter angle threshold for smaller ships would entail a more accurate approximate route.

To address these issues, we developed a system that computes, for each vessel type, the optimal compression parameter values from the ranges shown in the penultimate column of Table 1. The system keeps as few AIS positions as possible, i.e., it minimizes the *compression ratio*, while at the same time minimizes the *approximation error* in the resulting trajectory synopses. The compression ratio is the percentage of locations kept as critical points in the synopses over the noise-free raw locations for all vessels. The parameters that we optimize do not affect the noise reduction filters; thus, the number of noiseless positions is fixed for a given dataset of AIS messages. Typically, the approximation error is quantified with the Root Mean Square Error (in meters).

4.1 Genetic Algorithm

Our system employs a Genetic Algorithm (GA) to optimize the trajectory compression parameter values *per vessel type*. GAs can

Table 1: Parameters of vessel trajectory synopses.

Symbol	*Parameter*	*Value range*	*Default value*
$\Delta\theta$	Angle threshold (^{o})	2 . . . 25	4
m	Buffer size (locations)	3 . . . 50	5
ΔT	Gap Period (seconds)	200 . . . 5000	1800
ω	Historical timespan (seconds)	300 . . . 5000	3600
v_{min}	No speed threshold (knots)	0.05 . . . 2	0.5
v_θ	Low speed threshold (knots)	0.05 . . . 8	5
α	Speed ratio	0.01 . . . 0.8	0.25
D	Distance threshold (meters)	2 . . . 100	50

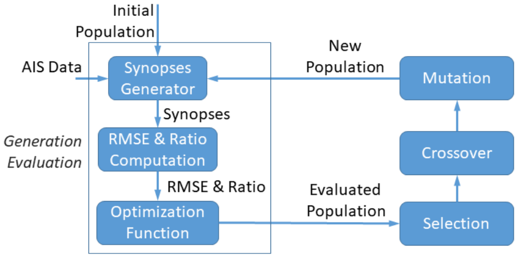


Figure 2: Fine-tuning trajectory compression parameters.

solve optimization problems that are often intractable because of their large space of parameters. In each generation, GA keeps a population of *individuals*. In our case, each individual is a tuple that contains specific values for the trajectory compression parameters (see Table 1). The individuals of the initial population may be picked using e.g. a uniform distribution. Then, the fitness of each individual is computed. To do this, the Synopses Generator (see Section 3) is instructed to compute the synopsis according to the parameter values of the individual. Then, the Compression Ratio and the RMSE of the resulting synopsis are used for calculating the fitness of the individual; the details of the fitness/optimization function will be discussed in the following section. Figure 2 illustrates our optimization process. The operators of selection, crossover and mutation are successively applied. We use *Tournament Selection* which repeatedly and randomly picks three individuals and selects the fittest, until the desired number of total individuals has been picked. Selection is followed by crossover; we adopt *single-point crossover*, with a probability of 0.4. Finally, we employ *Gaussian Mutation*, which adds random Gaussian noise to each value of an individual with some probability. The mutation probability is set to 0.8 since the Gaussian Mutation restricts mutation. The probability that each trait of an individual would be mutated is set to 0.5. The steps of generation evaluation, selection, crossover and mutation are repeated for a fixed number of times or until convergence.

4.2 Optimization Function

In earlier work [3], we instructed the GA to minimize

$$(RMSE + r)^n \times Ratio \tag{1}$$

in order to minimize both *RMSE* and *Ratio*. In order to set values to hyper-parameters r and n, we had to train the GA for various value combinations and choose the combination with *RMSE* and *Ratio* values below certain, user-specified thresholds. Hyper-parameter tuning may result in sub-optimal values for the hyper-parameters, as the data used in this process may have different statistical properties from the remaining dataset. Moreover, choosing the thresholds for n and r requires knowledge of the given dataset, since there is no guarantee that these thresholds can be satisfied.

To avoid such issues and the computational overhead of hyper-parameter tuning, we define the following optimization function:

$$Ratio + ReLU(RMSE - \theta) \tag{2}$$

ReLU is the the *Rectified Linear Unit* function, that is, $ReLU(x) = \max(0, x)$, whereas θ is a threshold for *RMSE*. Intuitively, *RMSE* must be below a certain value so that a synopsis can be characterized as reliable and not deviating from the original trajectory. This is the role that *ReLU* serves: if *RMSE* exceeds θ and thus the synopsis is considered less reliable, then minimizing function (2) leads to effectively minimizing *RMSE*, since *Ratio* is always less than 1 and *RMSE* typically takes much higher values. In contrast, as long as *RMSE* is less than θ, the minimization of function (2) leads to the minimization of *Ratio*. Thanks to function (2), we can set the desired value for *RMSE*, and get synopses with a similar quality and the smallest possible *Ratio*. In Section 5, we will present an empirical evaluation of the use of this function.

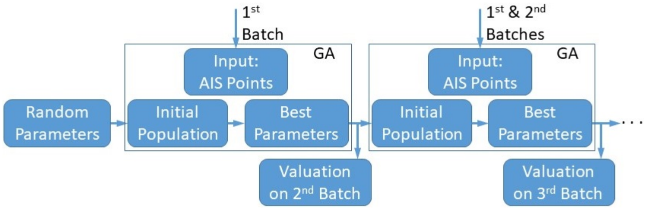


Figure 3: Incremental Optimization.

4.3 Incremental Optimization

Our system supports incremental optimization, by training in steps, where each step concerns a data batch. At the i-th step, the initial population of GA is set to the best individuals that were computed at the previous training step, i.e. the individuals computed using data up to the $i-1$-th batch. Recall that each individual is a tuple of compression parameter values. Then, training is performed using all data seen so far, i.e. data batches 1 to i. Figure 3 shows the processing flow. The individuals, i.e. parameter values, computed at the i-th step are evaluated on the i+1-th data batch.

4.4 Composite Event Recognition

Composite event recognition engines [4] are an integral part of maritime monitoring systems to support safe shipping via online detection of events like ship-to-ship transfer, loitering and tugging. Such composite events are defined as spatio-temporal patterns continuously matched on AIS data streams and static information, like protected zones or port areas. To achieve online performance, composite event recognition engines rely heavily on trajectory summarization. Our system employs the 'Run-Time Event Calculus' (RTEC), an open-source[3] composite event recognition engine with formal, declarative semantics successfully deployed in the maritime domain [1, 18]. As will be shown in the empirical analysis, trajectory summarization affects the predictive accuracy of RTEC only very slightly. Indeed, when RTEC operates on compressed trajectories resulting from the GA optimization, it performs more efficient maritime event recognition, compared to consuming compressed trajectories generated under the default parametrization.

5 EMPIRICAL ANALYSIS

5.1 Experimental Setup

We used two real-world datasets to evaluate our system. The first one is a publicly available dataset [20] with AIS messages over 6

[3] https://github.com/aartikis/RTEC

Table 2: Tested AIS datasets: BR (*top*) and MS (*bottom*).

	Vessel type	*AIS Messages*	*Vessel Count*
BR	Passenger Ships	4,792,487	17
	Unknown	3,466,765	115
	Fishing Boats	3,288,577	161
	Tug Boats	1,411,761	15
	Cargo Vessels	1,198,228	184
	Military	802,045	12
	Entire Dataset	19,035,631	5,055
MS	Cargo Vessels	48,646,162	5,804
	Fishing Boats	36,696,167	3,634
	Pleasure Crafts	18,157,399	6,057
	Sailing Vessels	18,157,399	8,062
	Passenger Ships	16,250,325	1,424
	Tankers	14,466,046	1,747
	Entire Dataset	221,772,127	35,546

months (October 2015 to March 2016) in the area of Brest, France (hereafter 'BR dataset'). The second dataset spans 6 months (March to August 2016) across the Mediterranean Sea ('MS dataset', for short), courtesy of MarineTraffic, our partner in the INFORE project and owner of the largest network of AIS base stations globally. Table 2 lists the total size of each dataset, the number of AIS messages and vessel counts for the six vessel types with the most messages. In our analysis, we will focus on these six vessel types.

As already noted, AIS data stakeholders like MarineTraffic discard a large amount of originally relayed positions to reduce the bulk of accumulated information. This is the case with the MS dataset, which has an average sampling rate ten times lower than the BR dataset. Thus, its *Ratio* is expected to be much higher, as well as *RMSE* values, since many critical points containing significant information may have been discarded by the data provider.

Our system is open-source software[4] built on Python, Scala, and Prolog. More specifically, GA is implemented in Python 3 using the Deap framework[5], the Synopses Generator in Scala on top of Apache Flink[6], and RTEC was written in Prolog and tested under Yap 6.2[7]. Our experiments were conducted on a Linux server with Intel® Xeon® CPU E5-2630 v2 @ 2.60GHz and 256GB RAM.

5.2 Genetic Algorithm

We present the *RMSE* and *Ratio* values achieved by training the GA for various values of θ, i.e. the threshold for *RMSE* in the optimization function (see Eq. (2)). For comparison, we also present the *RMSE* and *Ratio* values for synopses produced under the default compression parameter values as specified in the last column of Table 1, which were picked with the help of domain experts for the BR dataset. Moreover, we present the *RMSE* and *Ratio* values of the synopses generated by our earlier optimization function, i.e. Eq. (1).

We performed 6-fold cross validation on the six months of the BR dataset (approx. 14M AIS messages), and on two months of the MS dataset, March and June (approx. 53M AIS messages for both), in order to compare seasonal variations in the results. In the interest of space, Figure 4 displays some indicative results. To facilitate understanding, we omitted results achieved using the default parameter values on the MS dataset. As default parameter values increase *RMSE* significantly, we report the these results in Table 3. Concerning the use of the earlier optimization function, i.e. Eq. (1), hyper-parameter tuning in the BR dataset was constrained to achieve an *RMSE* value between 15m–30m and a *Ratio* value between 10%– 30%. In the more challenging MS dataset due to the sparser update frequency of vessels, hyper-parameter tuning was performed by constraining *RMSE* to 80m and *Ratio* to 50%.

Table 3: Results with default parametrization on MS dataset.

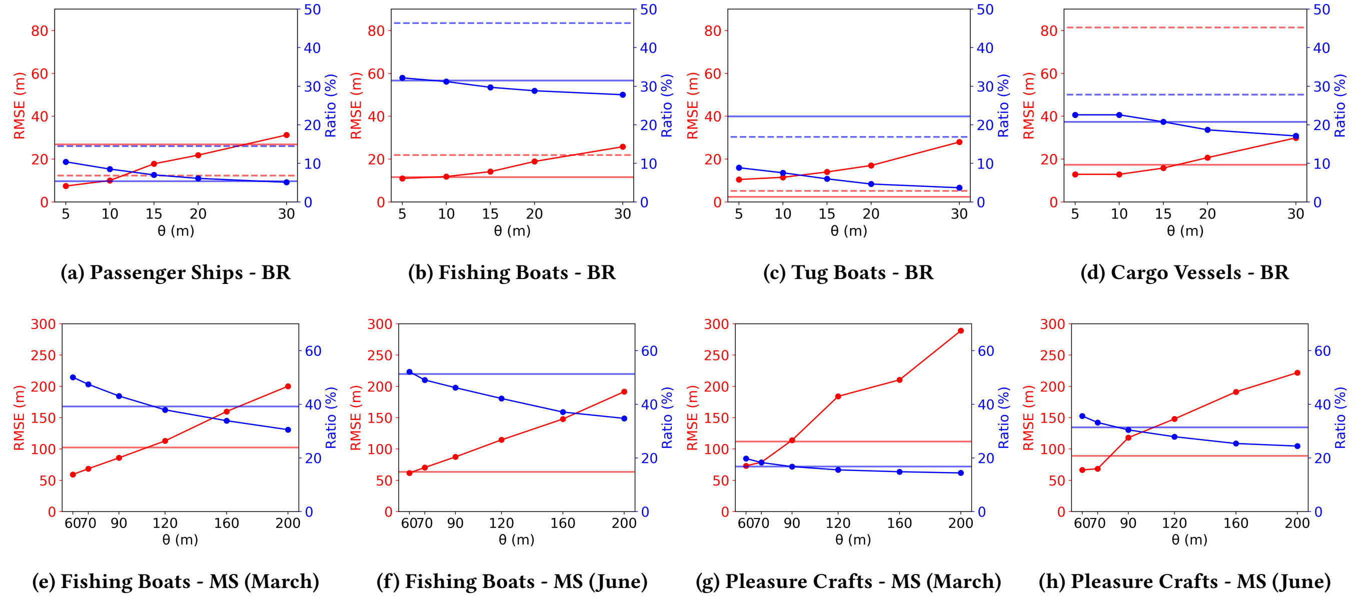
	RMSE		*Ratio*	
Vessel type	*March*	*June*	*March*	*June*
Cargo Vessels	1,099m	728m	15%	18%
Fishing Boats	107m	79m	52%	53%
Pleasure Crafts	367m	220m	29%	21%
Sailing Vessels	318m	188m	36%	31%
Passenger Ships	710m	662m	42%	43%
Tankers	846m	567m	19%	23%

For the BR dataset, Figure 4 shows that the results of the default parameters are satisfactory. However, as it should be expected, using different parameters for each ship type yields better results. In all the plots in the top row (but one), picking the desired value for θ yields better results, both in terms of *RMSE* and *Ratio*. The only exception are Tug Boats, where the *RMSE* is too low; by increasing the *RMSE* by a small value, which makes no practical difference in the approximation error, we manage to achieve half the value for *Ratio*. For the MS dataset, the results from the default parameters in Table 3 immediately show that their performance is unsatisfactory; *RMSE* values of over 300m lead to huge inaccuracies between the original trajectories and their synopses with the exception of Fishing Boats, which obtain tolerable *RMSE* values.

However, with a proper choice for θ, GA can give a similar *RMSE* and a lower *Ratio*. Notice in Figure 4 that when GA minimizes the earlier optimization function (Eq. (1)), performance is good but 'unpredictable'. The resulting synopses are always close to satisfying the thresholds set during hyper-parameter tuning, but the final *RMSE* and *Ratio* are not always what we would prefer. This can be seen for Tug Boats in the BR dataset, where although the result is more than acceptable, its *RMSE* is a lot less than the threshold set. Instead, the synopses generated by function (2) manage to keep an acceptable error and a lower *Ratio*. In the MS dataset where the same thresholds were used for all vessel types to tune the hyper-parameters of our earlier methodology, we also see unpredictability: although keeping the same thresholds, the results for each ship type are different, because the hyper-parameters were picked to satisfy the thresholds on a different set of data. In contrast, when GA minimizes function (2), the *RMSE* is almost always close to the value of θ, and given that, the *Ratio* takes the best possible value.

In the MS dataset, we observe that trajectory compression for Fishing Boats is less effective compared to that for Pleasure Crafts; to achieve the same quality in terms of *RMSE*, more AIS locations must be kept in the synopses for Fishing Boats (higher *Ratio*) than for Pleasure Crafts. Additionally, we see that Fishing Boats have similar behavior in different seasons. In the results of the MS dataset, the curves are almost identical between Figure 4e and Figure 4f.

[4] https://github.com/GiannisFikioris/Genetic-Algorithm-for-Synopses-Generator
[5] https://deap.readthedocs.io
[6] https://flink.apache.org/
[7] https://en.wikipedia.org/wiki/YAP_(Prolog)



(a) **Passenger Ships - BR**

(b) **Fishing Boats - BR**

(c) **Tug Boats - BR**

(d) **Cargo Vessels - BR**

(e) **Fishing Boats - MS (March)**

(f) **Fishing Boats - MS (June)**

(g) **Pleasure Crafts - MS (March)**

(h) **Pleasure Crafts - MS (June)**

Figure 4: *RMSE* (red lines) and *Ratio* (blue lines) for different values of θ. Top plots: BR dataset; bottom plots: MS dataset. Horizontal dashed (continuous) lines concern the use of the default parameter values (earlier optimization function Eq. (1)**).**

Pleasure Crafts exhibit much different behavior. In Figure 4g the *RMSE* values are much higher than the specified threshold θ. This is not due to erroneous training, as in that phase synopses with the desired error were found. This means that Pleasure Crafts tend to exhibit higher *RMSE* values in the test sets, most likely due to variance in mobility patterns. Something similar happens in Figure 4h (different season), although at a much smaller scale.

5.3 Incremental Optimization

We evaluated the process of incremental optimization, as presented in Section 4.3, against the MS dataset. The threshold θ for *RMSE* in Eq. (2) was set to 60m. We split the dataset into six data batches, each concerning a month of AIS messages. Consequently, the system performed five training steps. In the first step, GA was trained on the AIS messages of March and tested on the data of April. In the second step, the initial population was set to the best individuals of the previous training step, while the training data consisted of the AIS messages of March and April; the evaluation was carried out on the data of May. The next steps were performed similarly. Note that the first training step includes 15 generations, while the remaining four training steps include 10 generations. This is because in the i-th training step, where i>1, the initial population consists of the best individuals of the previous step, while in the first training step the initial population is randomly chosen.

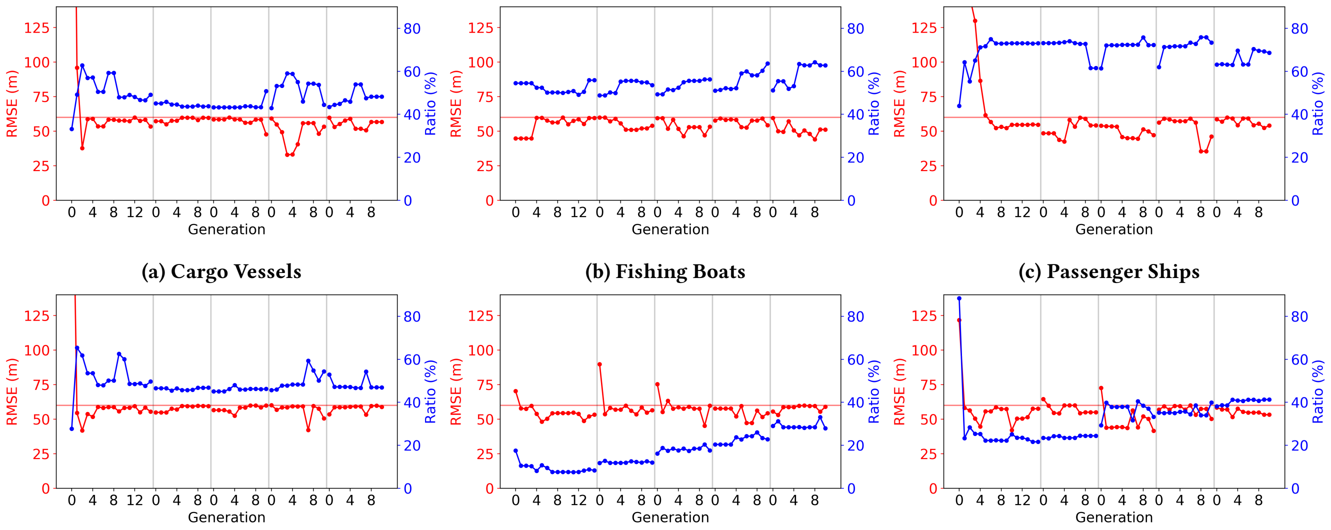
Figure 5 displays the performance of incremental optimization on the training sets. We show the *RMSE* and *Ratio* values of compressed trajectories computed using the best individual (i.e., compression parameter values) of each generation. Overall, there is continuity across the training phases; the *RMSE* and *Ratio* values of the last generation of a training phase are very close to the corresponding values of the first generation of the following phase. Pleasure Crafts slightly deviate from this behavior: *RMSE* is higher than the threshold of 60m in the beginning of some training phases. This is due to variability in the movement patterns of Pleasure Crafts in different months. Additionally, Figure 5 illustrates that parameter values that satisfy the *RMSE* threshold of θ = 60m are easy to find, and they are computed early in the training phase.

Figure 6 presents the performance of incremental optimization on the test sets. For example, the displayed *RMSE* and *Ratio* values of August concern trajectory compression on the AIS messages of August using the best individual produced by incremental training over all previous months (March to July). By and large, the good performance of the parameter values during training carries over to the test sets. For all but one vessel types, we attain *RMSE* close to the threshold of 60m, whereas *Ratio* does not deviate from the values achieved during training. Interestingly, for Sailing Vessels (Figures 5f and 6f), more original AIS messages must be kept in the synopses during the summer months, when mobility is increased, in order to maintain an acceptable trajectory reconstruction error.

Concerning Pleasure Crafts, we notice a similar behavior to that of Sailing Vessels, where the value of *Ratio* keeps increasing as more months are examined (see Figures 5e and 6e). Even though the performance in the training phase is more than satisfactory, with the lowest *Ratio* values across all ship types (Figure 5e), in the early testing phases we see an unusually high value of *RMSE* (Figure 6e). This is consistent with the findings reported in Section 5.2. As more training data gets processed, the *RMSE* values gradually converge to the desired θ value (see, again, Figure 6e).

5.4 Composite Event Recognition

Next, we assess the effects of synopses produced under GA optimization on composite maritime event recognition. The first column



(a) Cargo Vessels

(b) Fishing Boats

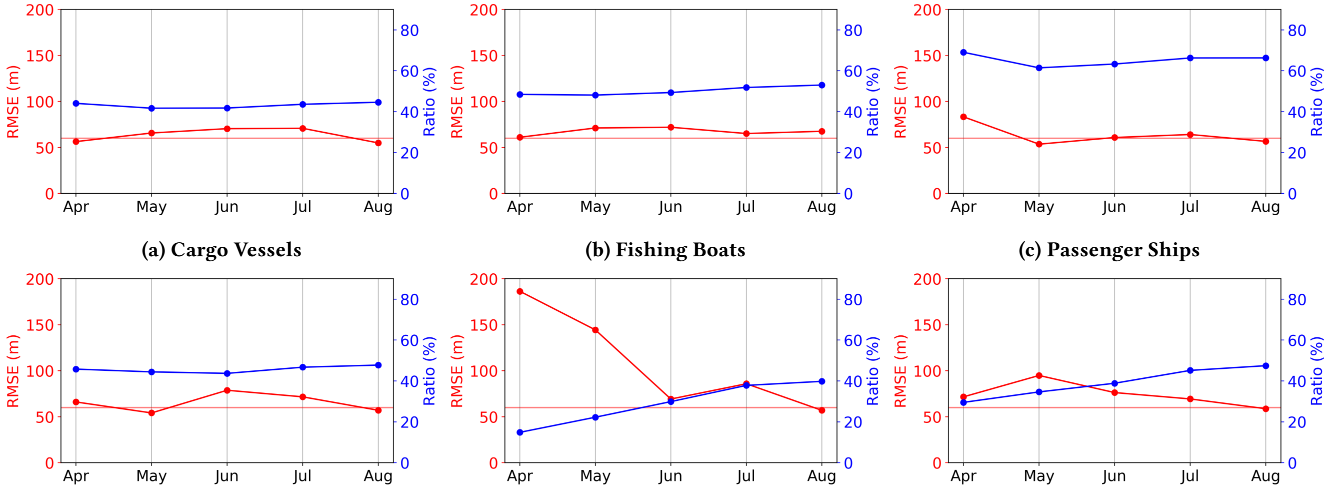
(c) Passenger Ships

(d) Tankers

(e) Pleasure Crafts

(f) Sailing Vessels

Figure 5: Performance of Incremental Optimization on the *training sets* over generations for the MS dataset. *RMSE* (red) and *Ratio* (blue) are shown for the best individuals (parameter values). The vertical lines separate the training steps and the horizontal line indicates the *RMSE* threshold θ. For tankers, we omitted the first *RMSE* value to avoid re-scaling the plot.



(a) Cargo Vessels

(b) Fishing Boats

(c) Passenger Ships

(d) Tankers

(e) Pleasure Crafts

(f) Sailing Vessels

Figure 6: Performance of Incremental Optimization on the *test sets* for the MS dataset.

in Table 4 lists the maritime events under investigation, which are all durative. For instance, ‘ship-to-ship transfer’ is detected when two vessels are stopped in the open sea closely to each other for at least some time interval. Detection of such events requires fusing the trajectories of different vessels, as well as computing the spatio-temporal relationships between them [18].

The composite event patterns are not restricted to the top-6 vessel types of our datasets. so, in these tests, we had to use all vessel types. We restricted attention to BR dataset, because it is possible to achieve a good summarization of the entire dataset by using the optimized parameter values of the top-6 vessel types. Some vessel types that are not in the top-6 have similar movement patterns to a ship type in the top-6; in these cases, we used the optimized parameter values of the latter for compressing the trajectories of the former. Less than 8% of the dataset includes vessel types with different movement patterns than those in the top-6. For this small part of the dataset, we used the default parameter values. In the optimization function of GA, the *RMSE* threshold θ was set to 15m.

Table 4 presents the predictive accuracy of RTEC when operating on the compressed dataset. The ground truth consists of the

Table 4: Accuracy of Composite Event Recognition.

Composite Event	*Precision*	*Recall*	*F_1-score*
Drifting Vessel	0.56	0.88	0.69
Search & rescue operations	0.98	0.99	0.99
Loitering	1.0	1.0	1.0
Piloting	1.0	1.0	1.0
Ship-to-ship transfer	1.0	1.0	1.0
Tugging	0.99	1.0	1.0
Vessel under way	0.99	0.78	0.87

composite event intervals that were computed when RTEC consumed the original (i.e., uncompressed) dataset. For example, the set of False Positives (resp. False Negatives) expresses the seconds in which a composite event is recognized when consuming the compressed (resp. uncompressed) dataset but not detected when consuming the uncompressed (resp. compressed) dataset. Table 4 shows that our system achieves perfect scores for most composite events. For the remaining events, there are some False Positives and False Negatives affecting Precision and Recall. False Positives arise when RTEC stops recognizing a composite event later when consuming the compressed dataset than when consuming the uncompressed one. Similarly, False Negatives arise when RTEC starts recognizing an event later when consuming the compressed dataset than when consuming the uncompressed one.

Table 5: Efficiency of Composite Event Recognition.

RTEC Input	*Average Time (sec)*	*Standard Deviation (sec)*	*Worst Time (sec)*
BR dataset	41.1	22.1	131.3
BR synopses [Def]	10.1	7.3	55.9
BR synopses [GA]	7.2	4.0	24.5

To support online recognition, RTEC operates with a sliding window. Table 5 shows the cost for recognizing all events displayed in Table 4, when RTEC operates with a 24-hour window over different input: the original (uncompressed) BR dataset, its synopses under default parameterization ('BR synopses [Def]'), and those optimized by GA ('BR synopses [GA]'). As expected, operating on synopses offers very significant performance gains. Moreover, the synopses derived using the GA are more succinct and thus lead to considerably more efficient composite event recognition, as opposed to those produced under the default parameter values.

6 SUMMARY AND FUTURE WORK

We presented a system to fine-tune the parameter values used for compressing vessel trajectories. Taking into account the characteristics of each vessel type, this system computes suitable parameter values that yield improved trajectory synopses, both in terms of approximation error and compression ratio. Furthermore, it supports incremental optimization, by training in data batches, and thus continuously improves its performance. A comprehensive empirical evaluation on two real-world datasets confirmed that compression efficiency is at least as good as the one with default parametrization, without relying on laborious data exploration. The resulting more succinct synopses also improve the recognition of composite maritime events without compromising its predictive accuracy. In future work, we plan to employ relabeling of critical points with extra rules and filters. For instance, ignoring turning points caused by small deviations, or making *low speed ratio* intervals narrower by superseding the assigned small values for *speed ratio*, may enable even more improved accuracy in complex event recognition.

ACKNOWLEDGMENTS

This work has received funding from the EU Horizon 2020 RIA program INFORE under grant agreement No 825070. We would also like to thank MarineTraffic for providing the MS dataset.

REFERENCES

[1] A. Artikis, M. J. Sergot, and G. Paliouras. 2015. An Event Calculus for Event Recognition. *IEEE Trans. Knowl. Data Eng.* 27, 4 (2015), 895–908.

[2] H. Cao, O. Wolfson, and G. Trajcevski. 2006. Spatio-temporal Data Reduction with Deterministic Error Bounds. *VLDB Journal* 15, 3 (2006), 211–228.

[3] G. Fikioris, K. Patroumpas, and A. Artikis. 2020. Optimizing Vessel Trajectory Compression. In *MDM*. 281–286.

[4] N. Giatrakos, E. Alevizos, A. Artikis, A. Deligiannakis, and M. N. Garofalakis. 2020. Complex event recognition in the Big Data era: a survey. *VLDB Journal* 29, 1 (2020), 313–352.

[5] C. Iphar, A. Napoli, and C. Ray. 2015. Detection of false AIS messages for the improvement of maritime situational awareness. In *OCEANS*. 1–7.

[6] I. Kontopoulos, K. Chatzikokolakis, K. Tserpes, and D. Zissis. 2020. Classification of vessel activity in streaming data. In *DEBS*. 153–164.

[7] R. Lange, F. Dürr, and K. Rothermel. 2011. Efficient real-time trajectory tracking. *VLDB Journal* 20, 5 (2011), 671–694.

[8] X. Lin, J. Jiang, S. Ma, Y. Zuo, and C. Hu. 2019. One-pass trajectory simplification using the synchronous Euclidean distance. *VLDB Journal* 28, 6 (2019), 897–921.

[9] X. Lin, S. Ma, H. Zhang, T. Wo, and J. Huai. 2017. One-Pass Error Bounded Trajectory Simplification. *PVLDB* 10, 7 (2017), 841–852.

[10] J. Liu, K. Zhao, P. Sommer, S. Shang, B. Kusy, J.-G. Lee, and R. Jurdak. 2016. A Novel Framework for Online Amnesic Trajectory Compression in Resource-Constrained Environments. *IEEE Trans. Knowl. Data Eng.* 28, 11 (2016), 2827–2841.

[11] H. Ljunggren. 2018. Using Deep Learning for Classifying Ship Trajectories. In *FUSION*. 2158–2164.

[12] C. Long, R. C.-W. Wong, and H.V. Jagadish. 2014. Trajectory Simplification: On Minimizing the Direction-based Error. *PVLDB* 8, 1 (2014), 49–60.

[13] N. Meratnia and R.A. de By. 2004. Spatiotemporal compression techniques for moving point objects. In *EDBT*. 765–782.

[14] J. Muckell, P.W. Olsen, J.-H. Hwang, C.T. Lawson, and S.S. Ravi. 2014. Compression of Trajectory Data: A Comprehensive Evaluation and New Approach. *Geoinformatica* 18, 3 (2014), 435–460.

[15] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet. 2018. A Multi-Task Deep Learning Architecture for Maritime Surveillance Using AIS Data Streams. In *DSAA*. 331–340.

[16] K. Patroumpas, E. Alevizos, A. Artikis, M. Vodas, N. Pelekis, and Y. Theodoridis. 2017. Online event recognition from moving vessel trajectories. *GeoInformatica* 21, 2 (2017), 389–427.

[17] K. Patroumpas, N. Pelekis, and Y. Theodoridis. 2018. On-the-fly mobility event detection over aircraft trajectories. In *ACM SIGSPATIAL*. 259–268.

[18] M. Pitsikalis, A. Artikis, R. Dreo, C. Ray, E. Camossi, and A.-L. Jousselme. 2019. Composite Event Recognition for Maritime Monitoring. In *DEBS*. 163–174.

[19] M. Potamias, K. Patroumpas, and T. Sellis. 2006. Sampling Trajectory Streams with Spatiotemporal Criteria. In *SSDBM*. 275–284.

[20] C. Ray, R. Dréo, E. Camossi, A.-L. Jousselme, and C. Iphar. 2019. Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance. *Data in Brief* 21 (2019). https://doi.org/10.5281/zenodo.1167595

[21] F. Terroso-Saenz, M. Valdés-Vela, E. den Breejen, P. Hanckmann, R. Dekker, and A. F. Skarmeta-Gómez. 2015. CEP-traj: An event-based solution to process trajectory data. *Information Systems* 52 (2015), 34–54.

[22] G. A. Vouros, C. Doulkeridis, G. Santipantakis, A. Vlachou, N. Pelekis, H. Georgiou, Y. Theodoridis, K. Patroumpas, E. Alevizos, A. Artikis, G. Fuchs, M. Mock, G. Andrienko, N. Andrienko, C. Ray, C. Claramunt, E. Camossi, A.-L. Jousselme, D. Scarlatti, and J. M. Cordero. 2018. Big Data analytics for time critical maritime and aerial mobility forecasting. In *EDBT*. 612–623.

[23] O. Wolfson, A.P. Sistla, S. Chamberlain, and Y. Yesha. 1999. Updating and Querying Databases that Track Mobile Units. *Distrib. Parallel Dat.* 7, 3 (1999), 257–287.

[24] D. Zhang, M. Ding, D. Yang, Y. Liu, J. Fan, and H. T. Shen. 2018. Trajectory Simplification: An Experimental Study and Quality Analysis. *PVLDB* 11, 9 (2018), 934–946.