GeoInformatica
https://doi.org/10.1007/s10707-022-00475-0

Optimizing vessel trajectory compression for maritime situational awareness

Giannis Fikioris[1] · Kostas Patroumpas[2] · Alexander Artikis[3,4] · Manolis Pitsikalis[5] · Georgios Paliouras[4]

Received: 19 July 2021 / Revised: 2 May 2022 / Accepted: 17 August 2022

Abstract
We present an open-source system that can optimize compressed trajectory representations for large fleets of vessels. We take into account the type of each vessel in order to choose a suitable configuration that can yield improved trajectory synopses, both in terms of approximation error and compression ratio. We employ a genetic algorithm that converges to a fine-tuned configuration per vessel type without any hyper-parameter tuning. These configurations can provide synopses that retain less than 10% of the original points with less than 20m approximation error in a real world dataset; in another dataset with 90% less samples than the previous one, the synopses retain 20% of the points and achieve less than 80m error. Additionally the level of compression can be chosen by the user, by setting the desired approximation error. Our system also supports incremental optimization by training in data batches, and therefore continuously improves performance. Furthermore, we employ a composite event recognition engine to efficiently detect complex maritime activities, such as ship-to-ship transfer and loitering; thanks to the synopses generated by the genetic algorithm instead of the raw trajectories, we make the recognition process faster while also maintaining the same level of recognition accuracy. Our extensive empirical study demonstrates the effectiveness of our system over large, real-world datasets.

Keywords AIS · Genetic algorithm · Maritime data analytics · Trajectory · Event recognition

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 in real-time through data exchange with other nearby vessels,

✉ Giannis Fikioris
gfikioris@cs.cornell.edu

Extended author information available on the last page of the article

Published online: 29 August 2022

coastal stations, and satellite constellations. More than 500,000 vessels worldwide can be tracked via AIS, thus supporting online maritime monitoring. Authorities can readily identify vessels that breach international regulations in navigation, prevent accidents, protect sensitive maritime zones, etc. AIS is also particularly important for shipping companies, enabling reduction of fuel consumption and improving vessel efficiency. The collected AIS raw tracking data is valuable, as it includes unique identification of vessels, their position, course, speed, etc. Although this data is not noise free, as AIS messages may be delayed, intermittent, or conflicting, it still offers capabilities for advanced processing, fusion, analysis and route scheduling by the shipping industry and maritime authorities.

However, managing in real-time the huge amounts of AIS data continuously emitted from numerous vessels worldwide is particularly demanding in terms of storage and processing. Typically, stakeholders 'decimate' the AIS data to be stored for analysis, by selectively or randomly dropping a significant percentage of the relayed AIS messages. This pre-processing stage ingests only a subset of the original AIS data, which can still provide an acceptable representation of the original trajectories of vessels without overwhelming the available system resources. The rationale is that successive locations emitted every few seconds from each vessel can hardly contribute additional knowledge about their actual motion patterns.

In [31], 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* indicating stops, turning points, changes in speed, etc. along each trajectory. In contrast to typical trajectory simplification, not only can this technique 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 seems indispensable when selecting 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. Further, this trajectory compression used to apply the same parametrized settings for all vessels, despite their differences in type, tonnage, length, etc., and hence in their motion patterns. Overall, choosing suitable compression parameters is a painstaking, time-consuming, data-dependent, and error-prone process.

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 is proprietary and is provided by MarineTraffic[1], one of the largest vessel tracking companies; it 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 and composite event recognition. 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 composite event recognition and real-time analytics. This system can be deployed against large-volume AIS data 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.

This paper is an extended and improved version of two previous works. In [11], we introduced a genetic algorithm that takes into account the vessel types and can provide a suitable configuration for the summarization parameters in order to yield improved trajectory synopses. This approach was further enhanced in [12] by avoiding hyper-parameter tuning, while also supporting incremental optimization and facilitating composite maritime event recognition. In this work, we study the efficiency of our compression engine for a wider range of vessel types. We also demonstrate in more detail the effect of such compression on recognizing complex events, by comparing the events identified in a non-summarized dataset against the events recognized from the trajectory synopses. Last, but not least, we introduce a new component that re-annotates the compressed trajectories according to user-specified values based on domain knowledge. We show that, compared with the original annotation assigned during the compression, this new annotation yields more accurate results in complex event recognition.

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, Sect. 6 summarizes the paper.

2 Related work

Trajectory Simplification Our approach on trajectory synopses over streaming AIS positions resembles well-known methods for path simplification. Clearly, *offline* techniques working in batch mode like the seminal Douglas-Peucker algorithm [9] require that the

[1] https://www.marinetraffic.com/

complete trajectory be available in advance. Algorithms like [24] take several passes over the sequence of stored locations to select which points to keep in the compressed trajectory. Although this paradigm may yield a reduced approximation error, offline processing generally incurs increased cost. Other algorithms like [1] emphasize only on spatial features and essentially address simplification of large polylines; temporal information in trajectories is entirely overlooked, although it plays a significant role especially for maritime monitoring.

In contrast, when positional data is streaming (as is our case with AIS locations emitted from vessels), trajectory simplification algorithms must work in *online* mode. Such trajectories are evolving and the complete motion history may be neither stored nor indexed. In contrast, such online techniques should examine the data in one pass, i.e., each incoming position must be processed as it arrives. Some online methods like *STTrace* [34] apply an upper bound on the memory footprint of the synopsis to restrict the approximation size, i.e., a small number k of points must be kept per moving object in an in-memory buffer. Initially, each fresh location is accepted in the buffer. Once its capacity is reached and a fresh position can no longer be accommodated, it discards from the buffer the location that changes the least the existing trajectory approximation, i.e., incurs the minimal error with respect to *Synchronous Euclidean Distance* (SED). Of course, samples retained in such synopses should keep each compressed trajectory as much closer to the original one, as in fitting techniques [6, 26], which minimize approximation error. The one-pass approach in [26] discards points buffered in a sliding window until the error exceeds a given threshold. The notion of safe areas in the threshold-based method suggested in [34] keeps samples that deviate from predefined error bounds regarding speed and direction.

Under a similar error-based principle, *SQUISH-E* (Spatial QUalIty Simplification Heuristic) [27] drops samples by employing a priority queue in order to achieve a target compression ratio. It supports two operation modes: *SQUISH-E(λ)* minimizes SED error while targeting a compression ratio λ, and *SQUISH-E(μ)* aims to maximize compression ratio ($\lambda = 1$) while keeping SED-estimated error below an upper bound μ. In empirical tests [27, 44] the latter mode seems to cope better, since it can generally remove more redundant points as long as the increased SED does not exceed the specified error bound μ. Dead-reckoning policies like [43] and mobility tracking protocols in [18] can also provide trajectory summaries with accuracy (error) bounds. However, their basic premise is that summarization runs locally on board of each moving object, relaying only positions that signify changes in their course. This can hardly be the case with AIS tracking data, as shipping companies and authorities need to track the whereabouts of vessels as frequently as possible.

The *Bounded Quadrant System* (BQS) algorithm [21] maintains a rectangular bounding box as well as two bounding lines for each of its quadrants using a window over the most recent (not yet compressed) trajectory portion. Once a new point arrives, it can be readily dropped or retained by checking with a given error tolerance ϵ (in distance units) against convex hulls formed by this box and bounding lines. A fast, relaxed version (*FBQS*) adds a line segment to the compressed trajectory and starts a new window once the convex hull cannot bound all locations in the current window. Further, its *amnesic* extension (*ABQS*) [22] aims to maximize the trajectory information retained in a fixed storage space and employs varying error tolerance depending on the age of the recorded segments. This aging-aware scheme is similar in spirit to a time-decaying approximation of streaming trajectories developed in [35] by gradually evicting older samples and offer greater precision for the most recent trajectory segments.

The one-pass, error-bounded algorithm *OPERB* suggested in [20] applies a novel local distance checking method and involves several optimizations in order to achieve

higher compression. It approximates the buffered points with a directed line and it checks if a fresh location fits with this line. If it has a distance more than a given error bound ζ from the directed line, a new segment is added to the approximation and a new starting point for the directed line is considered. A more aggressive variant of this algorithm allows "patches" by interpolating new points when moving objects have sudden changes in their paths or relay updates intermittently. A novel spatiotemporal cone intersection technique involving Synchronous Euclidean Distance is applied in [19] for more aggressive compression.

A survey and empirical study of various trajectory simplification techniques is available in [44]. In another recent survey [25], several state-of-the-art simplification algorithms are empirically compared specifically against AIS data. Although such generic online or offline methods can provide reliable summaries over vessel trajectories, they entirely lack support for mobility-annotated features in the retained sampled points.

Maritime Trajectory Synopses Going beyond trajectory simplification, the maritime surveillance system presented in [31] tracks vessel trajectories and also recognizes 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. Although such synopses are lossy approximations of the original trajectories, they are generally of high quality, as they can reconstruct the actual course of vessels with tolerable error comparable to that of generic online simplification methods [30]. Empirical results show that less than 5% of the raw data suffice to offer reliable trajectory approximations. This framework has been further enhanced to run in cluster infrastructures against scalable streaming data [32]. 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 [42]. With additional reasoning, the reported mobility events may also act as triggers of more *complex events* [3, 7, 13, 14], analogous to that of CEP-traj [39], RTEC [5, 33], Wayeb [2, 42] and situational awareness systems based on Markov Logic [38]. 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 Approaches over AIS data Machine learning techniques have been applied against AIS data for various objectives. Indicatively, classifying vessels by type from raw AIS trajectories is suggested in [23]. Employing a recurrent neural network, the deep-learning scheme in [29] supports various tasks in maritime traffic surveillance, such as detection of abnormal behavior, trajectory reconstruction, vessel type identification, etc. A convolutional neural network model is employed in [4] to specifically detect fishing activity from large historical datasets of vessel positions. In another direction, the tool proposed in [17] 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). Furthermore, online tools based on inductive logic programming construct composite event patterns, that may subsequently used for recognition [16]. However, none of these techniques aims at online trajectory simplification as our proposed method.

3 Online summarization of vessel trajectories

In this Section, we briefly examine the functionality of the Synopses Generator module[2]. Details about the applied trajectory summarization can be found in [31, 32].

[2] https://github.com/DataStories-UniPi/Trajectory-Synopses-Generator

This module provides online, summarized representations of trajectories of vessels. Usually, large amounts of raw AIS positions can hardly contribute additional knowledge about the actual motion pattern of each vessel. Indeed, most vessels normally follow almost straight, predictable routes at open sea, with the exception of "emergency" circumstances like adverse weather conditions, accidents, etc. The Synopses Generator drops any easily predictable positions along trajectory segments of "normal" motion characteristics and judiciously retains only critical points that incrementally maintain lightweight synopses of coherent trajectory segments.

More specifically, from each fresh AIS position the Synopses Generator extracts four attributes: *MMSI* (Maritime Mobile Service Identity), which is used as vessel identifier, *Longitude* and *Latitude* coordinates of the reported position, as well as its *Timestamp*. Some AIS positions are discarded as noise, as raw AIS data often contain unwanted positions (such as duplicate or delayed messages, invalid coordinates, etc. [15]). For efficiency, this is performed with several single-pass heuristics in online fashion, which sometimes discard up to 20% of the original AIS positions without harming vessel monitoring and reconstruction of their trajectories [31].

At each given time point, the most recent noiseless positions per vessel are buffered in memory. These positions are used to calculate the *mean velocity* $\mathbf{v}_m$ over the *most recent portion* of each evolving trajectory, as well as several derived spatiotemporal features (distance, travel time, overall change in heading, etc.). To achieve an accurate estimation of these features, the number of positions buffered in memory never exceeds a given parameter m. Another parameter, the *historical timespan* ω is used to discard obsolete positions from the buffer (e.g. if $\omega = 300$ then no positions older than 300 seconds are stored in memory). Finally, a *distance threshold* D parameter tackles inherent agility of GPS positions by dropping those AIS messages less than D meters away when the vessel is stopped.

The Synopses Generator applies single-pass heuristics to detect and annotate mobility events with suitable parametrization (Table 1). One, two, or multiple *critical points* may be retained for each such event in the synopsis of a vessel trajectory. In particular:

- *Stop* indicates that a vessel remains stationary over a period of time. This is done by checking whether its instantaneous speed v_{now} is lower than a threshold v_{min} (e.g., 0.5 knots) for every incoming AIS position. Only the first and last locations that satisfy the aforementioned threshold are annotated as critical points and are kept in the synopsis (indicating the start and the end of the event). In case a fresh location is found more than D meters away from the previous one, the stop event ends even if $v_{now} < v_{min}$.
- *Slow motion* indicates that a vessel sails at low speed for some time. This happens when the current speed v_{now} is consistently below a given threshold v_θ (e.g., < 5 knots) over a time interval. As with the previous event, the first and last point in this sub-trajectory are both annotated as critical.
- *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 any AIS location recently, e.g., in the past $\Delta T = 10$ minutes. The locations marking loss of contact and its restoration are denoted as critical.
- *Change in Heading* is detected when the current heading deviates more than an angle $\Delta\theta$ (e.g., $> 4^o$) from mean velocity $\mathbf{v}_m$. The AIS positions that satisfy this condition are marked as critical turning points. Since vessels generally make smooth turns, multiple such points may be successively issued as critical ones.

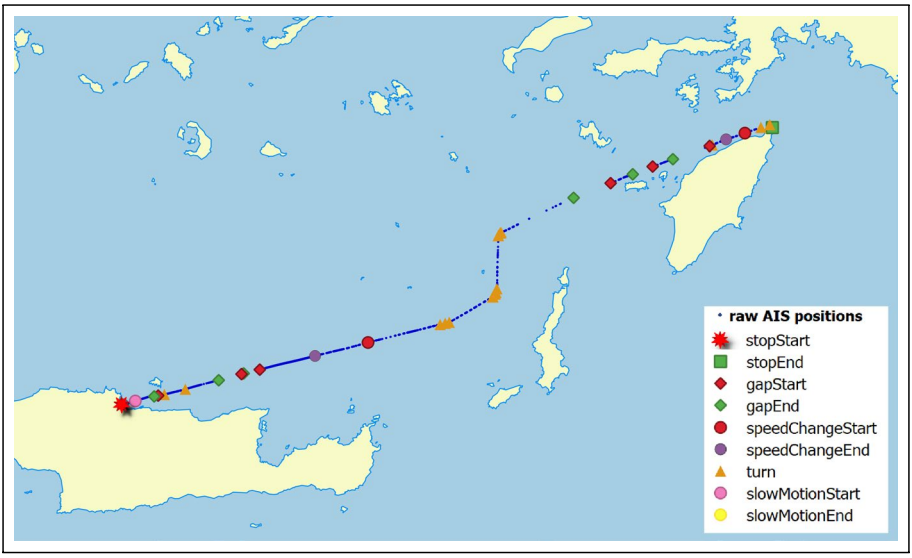


Fig. 1 Critical points detected along a vessel trajectory

We stress that this summarization process concerns only fresh locations actually relayed by the vessels; it does not make any short- or longer-term estimations about the future course of the vessel. All aforementioned mobility events are identified in online fashion based on the most recent historical portion of each trajectory, as continuously maintained and captured by its mean velocity $\mathbf{v}_m$. Hence, significant deviations in terms of heading or speed are not determined with respect to the entire trajectory, but only against its most recent motion segment spanning a few minutes and representing a handful of m recent locations per vessel. For example, a turning point is issued when the current heading differs more than $\Delta\theta$ from the known heading of velocity vector $\mathbf{v}_m$. Empirical evidence detailed in [30] indicates that the Synopses Generator yields high compression efficiency comparable with state-of-the-art trajectory simplification methods. Furthermore, unlike these general-purpose algorithms, this summarization framework purposely detects mobility patterns specific to vessels, like smooth turns or speed changes, by judiciously annotating selected locations as turn, stop, gap, etc., and discarding the rest.

Figure 1 depicts the trajectory of a vessel as reconstructed from the AIS positions it relayed during an itinerary in the Aegean Sea between the ports of Rhodes and Heraklion, Crete. The critical points detected along the route are shown with different symbols. Note that each such critical point is issued in real time, i.e., within milliseconds upon admission of a fresh AIS position. Thus, detection of mobility events keeps in pace with the streaming AIS data, and incrementally maintains the trajectory synopses. As a result, the original course of a vessel can be reliably approximated from its synopsis. Non-critical locations discarded from the synopsis can be estimated using a time-based interpolation. 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 [31].

Table 1 Parameters of vessel trajectory synopses

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

4 Adapting compression parameters

Trajectory compression is very sensitive to *parametrization*. Table 1 lists the parameters that control which positions are marked as critical by the Synopses Generator. The same table presents their default values, which have been picked with the valuable advice of domain experts, specifically for an AIS dataset concerning vessel activity in Brest, France [36]. Unfortunately, AIS datasets may differ in the sampling rate, the geographic area, the types of monitored vessels, etc., so the performance of the Synopses Generator with the default parameters would be far from acceptable. Additionally, using these parameters leads to treating all vessels uniformly, without taking into consideration their type, length, tonnage, etc. Because of the differences in the mobility patterns of the vessels this approach lacks flexibility. For example, because a larger ship takes turns more smoothly than a small ship (e.g. a Fishing or Tug Boat) allowing a more relaxed (i.e., greater) angle threshold for larger ships would not harm 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 in a dataset, the optimal compression parameter values from the ranges shown in the last column of Table 1. Our system keeps as few AIS messages 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 defined as the percentage of locations kept as critical points in the synopses over the noise-free raw locations for all vessels, i.e.:

$$Ratio = \frac{number\ of\ critical\ points\ for\ all\ vessels}{number\ of\ noiseless\ points\ for\ all\ vessels} \tag{1}$$

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):

$$RMSE = \sqrt{\frac{\sum_{p \in \text{noiseless points}} H^2(p, p')}{\text{number of noiseless points for all vessels}}} \tag{2}$$

where $H(\cdot)$ is the Haversine distance between each original location p from its time-synchronized point p' in the synopsis.

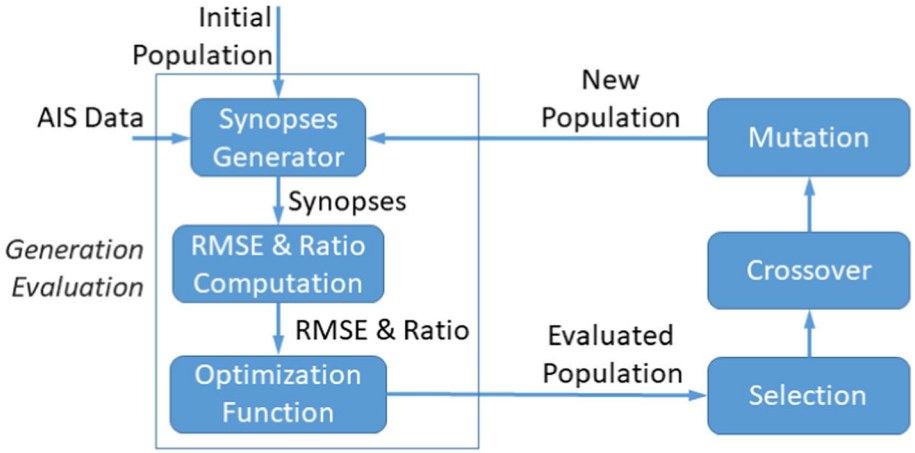


Fig. 2 Fine-tuning trajectory compression parameters

4.1 Genetic algorithm

Our system employs a Genetic Algorithm (GA) to optimize the trajectory compression parameter values *per vessel type*. GAs can solve maximization and minimization problems that are often intractable because of their large space of parameters and complicated nature of the optimization objective (non-monotone, non-convex, etc.). Through randomized decisions, GAs manage to both exploit and explore the parameter space by both focusing on parameter values that optimize the optimization function and doing more "random" decisions that lead to exploring the space away from values that seem to have good performance.

In each generation, GA keeps a population of *individuals*. In our case, each individual is a tuple of values that represents the trajectory compression parameters (see Table 1). The individuals of the initial population are usually picked at random, e.g. using a uniform distribution. After picking the individuals the fitness of each one is computed. To do this, the Synopses Generator (see Sect. 3) is instructed to compute the synopses of the vessels according to the parameter values of the individual. Then, the *Compression Ratio* and the *RMSE*, which are extracted from the synopses, are used for calculating the fitness of the individual; details about the fitness (optimization function) will be discussed in the following section. Figure 2 illustrates our optimization process.

After calculating the fitness of all the individuals, the operators of selection, crossover, and mutation are successively applied. First selection is applied, where we use *Tournament Selection*. This operator repeatedly and randomly picks three individuals and selects the fittest, until the desired number of total individuals has been chosen. Selection is followed by crossover; we adopt *single-point crossover*, with a probability of 0.4. Finally, for the mutation operator we employ *Gaussian Mutation*, which adds random Gaussian noise to each value of an individual with probability 0.5. The mutation probability is set to 0.8 since the Gaussian Mutation restricts mutation. The steps of generation evaluation, selection, crossover and mutation are repeated for a fixed number of times.

4.2 Optimization function

In our earlier work [11], we had instructed the GA to minimize the following optimization function, that simultaneously minimizes both *RMSE* and *Ratio*:

$$(RMSE + r)^n \times Ratio \tag{3}$$

where r and n are hyper-parameters. The goal of this function was to make *RMSE* and *Ratio* approximately inversely proportional to each other. This is exactly what happens when $r = 0$ and $n = 1$, in which case the optimization function considers equally good two summarizations with the same product of *RMSE* and *Ratio*.

In order to set values to these hyper-parameters, we had to train the GA for various combinations of their values and choose the combination that achieved *RMSE* and *Ratio* values below certain, user-specified thresholds (simply put, there is no combination of *RMSE* and *Ratio* that constitute an optimal synopsis but rather different applications require different compression levels). This hyper-parameter tuning could lead to 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 two thresholds for the *RMSE* and *Ratio* requires knowledge of the given dataset, since there is no guarantee that these thresholds can both be satisfied. This especially applies to datasets that have different sampling rates, where drastically different *RMSE* and *Ratio* combinations may be computed. Finally, to set the hyper-parameters we needed to run a laborious hyper-parameter setting phase which led to values that were hard to understand by the user.

To avoid such issues, as well as the computational overhead required for hyper-parameter tuning, we define the following optimization function:

$$Ratio + ReLU(RMSE - \theta) \tag{4}$$

where $ReLU = \max(x, 0)$ is the *Rectified Linear Unit* function and θ is a threshold for *RMSE*. Intuitively, many applications require the *RMSE* to be below a certain threshold so that the synopsis can be characterized as accurate and not deviating from the original trajectory. This is the role that *ReLU* serves: if *RMSE* exceeds the value of θ then the synopsis is considered less reliable and minimizing function (4) 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 the threshold θ, the minimization of function (4) leads to the minimization of *Ratio*, since the *RMSE* term "disappears".

Thanks to function (4), the user can set the desired value for θ (whose value and purpose are easily interpretable), which results in computing synopses with a similar *RMSE* and the smallest possible *Ratio*. This process does not require any hyper-parameter training for the optimization function and allows the algorithm to work with different levels of compression, according to the user's goals. Something similar happens in many trajectory simplification methods, where the resulting synopses are controlled by user specified parameters (see Sect. 2). In Sect. 5, we will present an empirical evaluation of the use of this function and comparison with our previous function, (3).

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 data used for training the GA are the batches n to i, where $n < i$. The initial population of the i-th step consists of the best individuals of the previous

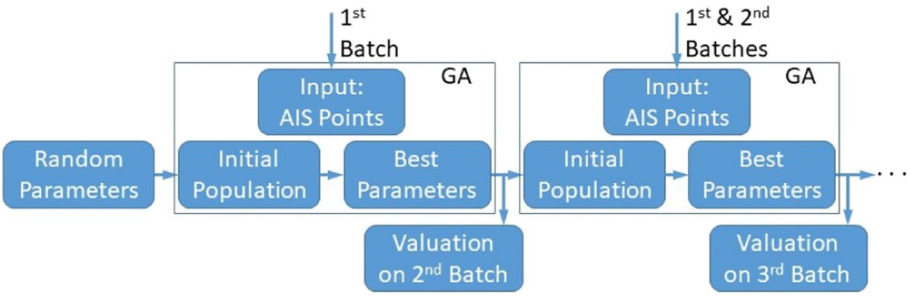


Fig. 3 Incremental Optimization

step (for the 1st step, the population may be randomly chosen). Thus after the training of the i-th step, we evaluate the best individual(s) on the (unseen) $(i + 1)$-th data batch. Figure 3 shows the processing flow. Incremental optimization aims to support "online" reasoning. In a real life system, where the Synopses Generator is used to compress the vessels' trajectories, one could re-train the GA periodically (e.g. every week), to update the compression parameters (e.g. various ships tend to have different behavior in different seasons).

4.4 Composite event recognition

Composite event recognition engines [3, 7, 13, 14] are an integral part of maritime monitoring systems aiming to support safe shipping via online detection of events, like ship-to-ship transfer, loitering and tugging. These composite events are defined using spatiotemporal patterns that are continuously matched on AIS data streams and static information, like protected zones and port areas. In order to achieve online performance, composite event recognition engines rely heavily on trajectory summarization. Our system uses the 'Run-Time Event Calculus' (RTEC), an open-source[3] composite event recognition engine with formal, declarative semantics, successfully deployed in the maritime domain [5, 33]. As will be shown in our empirical analysis, when RTEC operates on compressed trajectories generated by the GA optimization, it performs more efficient maritime event recognition, compared to consuming compressed trajectories generated under the default parametrization.

4.5 Relabeling of critical points

The GA summarizes the trajectories of the vessels using as less positions as possible, while at the same time keeping the reconstruction error bounded. To achieve this, the GA picks a set of parameter values that may be used to annotate the retained vessel positions (e.g., label them as stop, turn, gap). The positions that have one or more such labels are considered critical (as listed in Sect. 3) and are used in the synopsis maintained per vessel. Using such synopses and this labeling, RTEC detects composite maritime events.

Consider, for example, that the GA sets the value for *angle threshold* to $\Delta\theta = 20^o$; then, all positions where the vessel turns more than 20^o will be labeled as *turning points*. The

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

value of 20^o for *angle threshold* might be good to effectively summarize a vessel's trajectory, but it might not accurately represent all turning points. Since the range in parameter values is quite large (Table 1), the GA may pick values that compromise the predictive accuracy of composite event recognition. For instance, labeling as turning points only those with turning angle greater than 20^o may lead to false negatives in event recognition.

To address this issue, we designed a re-labeling scheme: Given that the points (and not the labels) picked by the GA summarize the original trajectories effectively, we change some of their labels. More specifically, we change the following annotations that are particularly important in recognizing certain complex events:

- *Stop* annotations, as these are important for recognizing *anchored or moored*, *piloting*, *rendezvous*, and *loitering* events. The relevant parameters that control the stopping labels are the *speed threshold* and *distance threshold*.
- *Change in heading* annotations, as these are important for recognizing *trawling* and *search and rescue* events. The relevant parameter that controls the turning labels is the *angle threshold*.
- *Change in speed* annotations, as these are important for recognizing *search and rescue* events. The relevant parameter that controls the acceleration labels is the *speed threshold*.

All other annotations assigned by the GA remain unchanged, as they have small impact on composite event recognition and their number is much smaller.

In order for the relabelling process to be successful, we need values for the four parameters specified above which accurately represent real world events. To do so, one needs domain knowledge and experts' advice. The other option is to have a ground truth that contains information about where turn, stop and acceleration events actually occur — using this we could automatically calibrate suitable parameter values for the relabeling process. Unfortunately, such ground truth is difficult to obtain even for a small number of trajectories, as it requires complete and accurate information in the raw AIS data and then meticulous inspection by experts. For this reason, we used parameter values that were chosen by domain experts (the corresponding experiments are presented in Sect. 5.6).

After specifying values for the parameters, the relabeling is done using the vessels' synopses. Note that since this process examines only critical points, it can be applied in online fashion by simply checking the mobility status (speed, turn, etc.) of each one and readily determining its new annotations. For example, if the new value for angle threshold is 4 degrees, then every critical point at which a turn of more that 4 degrees was recorded is labeled with a *change in heading* annotation, while critical points with a turn of less than 4 degrees have their *change in heading* annotation removed (if they had one). It should be noted that in the uncompressed trajectories there might have been points in which the vessel turned more than 4 degrees; if these points are not present in the GA's synopses, then they are also absent after the relabeling process. A similar process is followed for relabeling the other two annotations and is based on the respective rules outlined in Sect. 3 and discussed in detail in [32].

The relabeling process results in trajectories that contain the same positions as the synopses computed by the GA, but with different labels. Therefore, we retain the good summarization of the original trajectories, while at the same time aiding composite event recognition.

Table 2 AIS Datasets

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

5 Empirical analysis

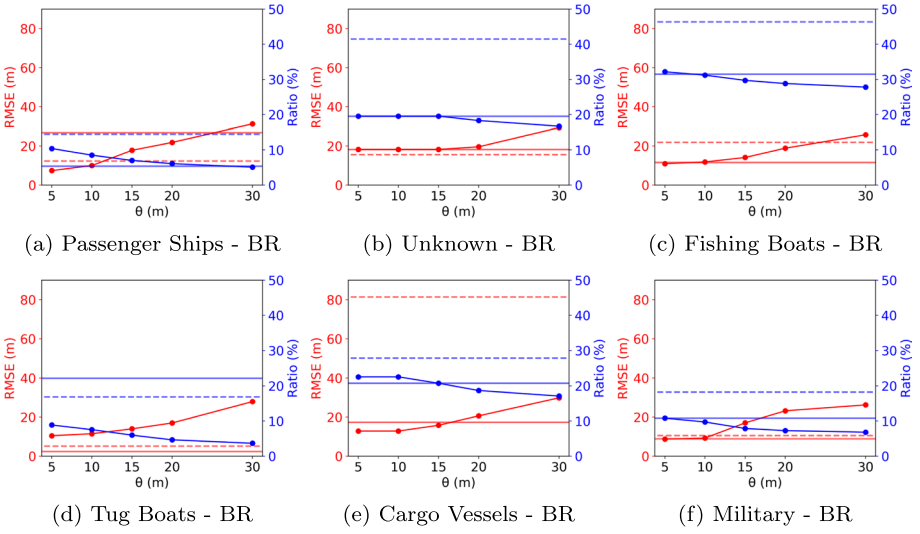
5.1 Experimental setup

We used two real-world AIS datasets to evaluate our system. The first one is a public dataset [36] covering the area of Brest, France (hereafter called the 'BR dataset'), which spans a time period of 6 months (October 2015 to March 2016). The second dataset also spans 6 months (March to August 2016), with vessel positions from the Mediterranean Sea (hereafter 'MS dataset'), given to us by MarineTraffic, who is our partner in the INFORE project, and also owner of the largest network of AIS base stations globally. Table 2 displays the number of AIS messages and vessels for the six vessel types with the most messages. In our analysis, we will mostly focus on these six vessel types.

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

Our system is open-source software[4] built using Python, Scala and Prolog. The 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[7] and SWI Prolog[8]. Our experiments were conducted on a Linux server with Intel® Xeon® CPU E5-2630 v2 @ 2.60GHz and 256GB RAM.

[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)
[8] https://www.swi-prolog.org/



(a) Passenger Ships - BR (b) Unknown - BR (c) Fishing Boats - BR

(d) Tug Boats - BR (e) Cargo Vessels - BR (f) Military - BR

Fig. 4 BR dataset: *RMSE* (red lines) and *Ratio* (blue lines) for different values of θ. Horizontal dashed (continuous) lines concern the use of the default parameter values (earlier optimization function Eq. (3))

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. (4)). 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. (3).

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. Figures 4 and 5 display the results. To facilitate understanding, we omitted the results achieved using the default parameter values on the MS dataset. As default parameter values increase *RMSE* significantly, we report these results only in Table 3. Concerning the use of the earlier optimization function, i.e. Eq. (3), 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, with the sparser update frequency of vessels, hyper-parameter tuning was performed by constraining *RMSE* to 80m and *Ratio* to 50%.

For the BR dataset, Fig. 4 shows that the results of the default parameters are satisfactory. However, as expected, using different parameter values for each ship type yields better results. In all plots, 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

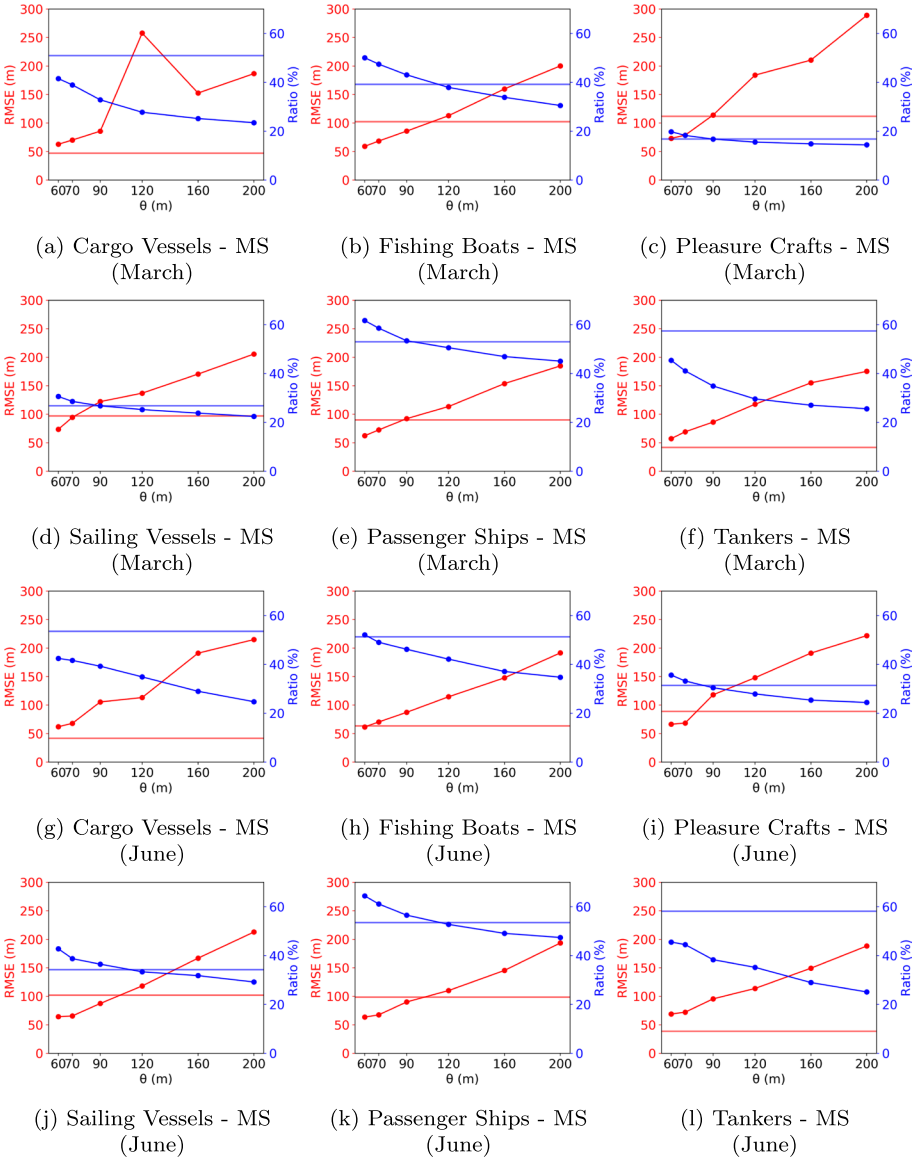


Fig. 5 MS dataset: *RMSE* (red lines) and *Ratio* (blue lines) for different values of θ. Top six figures: March; bottom six figures: June. Horizontal continuous lines concern the use of the earlier optimization function Eq. (3)

Boats which obtain tolerable *RMSE* values. However, with a proper choice for θ, the GA can give a similar *RMSE* and a lower *Ratio*.

Additionally, in Figs. 4 and 5, when the GA minimizes the earlier optimization function (Eq. (3)), 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

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%

dataset (Fig. 4), where although the result is more than acceptable, its *RMSE* is a lot less than the threshold value. Instead, the synopses generated by function (4) manage to keep an acceptable error and a lower *Ratio*. In the MS dataset (Fig. 5), where the same thresholds were used for all vessel types to tune the hyper-parameters of our earlier methodology, we also see unpredictability: although the threshold values are the same, the results for each ship type are different, because the hyper-parameters were computed on a different set of data. In contrast, when the GA minimizes function (4), the *RMSE* is almost always close to the value of θ, and given that, the *Ratio* takes the best possible value.

Figures 4 and 5 illustrate for which vessel type the trajectory compression is more or less efficient. For example, in the MS dataset, while the *RMSE* curve is (with some exceptions) an almost straight line that satisfies the thresholds set by θ, the *Ratio* curve is similar in all diagrams, but shifted upwards or downwards — vessel types where the *Ratio* curve is higher correspond to less efficient trajectory compression. Interestingly, the *Ratio* curve is in similar position for both March and June for each vessel type.

The vessel types in the MS dataset where the *RMSE* curve is not a straight line that satisfies the thresholds set by θ, are Cargo Vessels and Pleasure Crafts. For Cargo Vessels, the only irregularity is in Fig. 5a, where for $\theta = 120m$ the *RMSE* is very high — this is most likely caused because the parameters picked by the GA during the training phase were not appropriate to summarize the dataset in one of the test datasets. Pleasure Crafts exhibit a different behavior. In Fig. 5c 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. Instead, Pleasure Crafts tend to exhibit higher *RMSE* values in the test sets, most likely due to variance in mobility patterns. Something similar happens in Fig. 5i (different season), although at a much smaller scale.

5.3 Comparison with online trajectory simplification methods

Next, we compare GA with several state-of-the-art trajectory simplification methods. In particular, we conducted a qualitative analysis regarding the effects of simplification over vessel trajectories, primarily in terms of compression efficiency (average *Ratio* over all vessels) and approximation quality (the average *RMSE* over all vessel trajectories). We consider several trajectory simplification algorithms: SQUISH-E [27], OPERB [20], FBQS [21], and *STTrace* [34], and we simulate their execution in *online* mode over the BR dataset using the implementations offered in [44]. Over the same data, we also apply GA, which apart from identifying critical points online, it also annotates them (as turns, stops, etc.), something that no simplification method inherently supports. Each method's settings

Table 4 Comparison of simplification algorithms over the BR dataset. Values in bold indicate the algorithms that achieve the best performance in average Ratio or RMSE

Algorithm	Parametrization	Avg. Ratio	Avg. RMSE (m)
SQUISH-E [27]	Target Ratio = 1, $\mu = 0.0002$	0.347	**21.29**
OPERB [20]	Error bound $\zeta = 0.0002$	**0.104**	206.39
FBQS [21]	Error tolerance $\epsilon = 0.0002$	**0.0996**	187.54
STTrace [34]	Target Ratio = 0.2	0.1996	31.64
GA	Target RMSE $\theta = 15^o$	0.245	22.04

and the corresponding results are listed in Table 4. Since a detailed empirical comparison is beyond the scope of this paper, we stress that parametrization of each method is not fine-tuned, hence these results can only be seen as indicative.

From the results in Table 4, it is clear that all methods tend to drop many original locations in order to yield a simplified trajectory, although at a different compression ratio depending on the specifications of each algorithm and its objectives. Therefore, they also differ in approximation quality. Of all tested methods, *SQUISH-E* seems to offer supreme quality with minimal RMSE in the resulting synopses. Yet, in order to achieve such accuracy it has to retain many more locations compared to *OPERB*, *FBQS*, and *SSTrace*, hence its compression efficiency (*Ratio*) is the worst.

If substantial data reduction is the principal objective, *OPERB* and *FBQS* are more suitable and competitive to each other both in terms of quality and compression. *OPERB* seems to preserve more locations along turns as those points exceed the specified error bound ζ with respect to the directed line constructed from the recent motion history. Still, both algorithms seem to yield increased approximation error as only a 10% of the original locations is retained, hence the compressed trajectories may deviate significantly from the original ones. Although this may be a side effect of the ad-hoc parameter settings applied in this test, it clearly shows the trade-off between data reduction and approximation quality.

STTrace accurately matches the target compression ratio (20% in this example), as it parsimoniously preserves "good" locations that incur minimal error in the resulting trajectory approximation. Unlike uniform sampling, it tends to keep more points along turns, so as to closely maintain the shape of each path. Since the buffer capacity is set to 20% of the original locations per trajectory, longer trajectories are not oversimplified and their approximations retain more samples compared to shorter ones.

GA generally yields acceptable compression efficiency as the amount of retained critical points is comparable with the aforementioned state-of-the-art methods. Its parameter settings are according to the values that GA found as optimal in the experiments in Sect. 5.2. Specifically tailored for maritime trajectories, it purposely detects mobility patterns of vessels, like smooth turns or speed changes, so it may preserve the involved locations in the resulting synopses. Most importantly, its added value is that each judiciously retained critical point also carries *annotations*, e.g., turn, stop, gap. etc., not available from any of the general-purpose simplification methods. Obviously, such filtering greatly depends on proper choice of parameter values, which is a trade-off between reduction efficiency and approximation accuracy. Overall, GA seems to have a very balanced behavior, as it accomplishes a very low approximation error (only *SQUISH-E* fares marginally better in terms of quality) at the expense of a fair compression ratio. With a more relaxed setting (e.g., $\theta = 5^o$

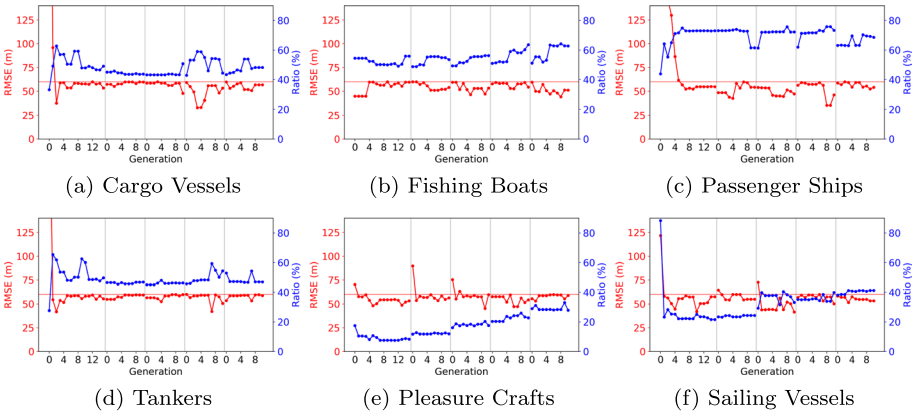


Fig. 6 Performance of Incremental Optimization on the *training sets* over generations for the MS dataset. *RMSE* (red) and *Ratio* (blue) are shown for the best individuals (parameter values). The vertical lines separate the training steps and the horizontal line indicates the *RMSE* threshold θ. For tankers, we omitted the first *RMSE* value to avoid re-scaling the plot

instead of $\theta = 15^o$) more locations could qualify as critical points (e.g., turns), capturing slighter changes along each trajectory.

5.4 Incremental optimization

We evaluated the process of incremental optimization, as described in Sect. 4.3. We used only the MS dataset, since this has enough data to create substantially large data batches. We set the threshold θ, which controls the desired *RMSE* (Eq. (4)), to 60m. We split the MS dataset into six disjoint data batches, where each batch had a month of AIS messages. Consequently, the system performed five training steps: in the first step, the GA was trained on the AIS messages of March and the resulting parameter values where evaluated on the data of April. In the second step, the initial population of the GA was set to the best individuals of the previous training step; the training data consisted of the AIS messages of March and April. Then, the evaluation was carried out on the data of May. The following three steps were performed in a similar way. We should note that the first training step includes 15 generations of individuals, while the remaining four training steps include only 10 generations. We made this choice because in all training steps, except the first one, the starting population is set to the best individuals of the previous step. In contrast, in the first training step the initial population is randomly chosen, requiring a larger number of generations to reach acceptable results.

In Fig. 6 we present the performance of incremental optimization during training: we report the *RMSE* and *Ratio* values of the synopses produced using the best individual, i.e. the compression parameter values that minimized Eq. (4), of each generation. Overall, there is continuity across the training phases; both 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 exhibit slightly different behavior: their *RMSE* value is higher than the threshold of 60m in the first generation of some training phases. This is caused by the fact that

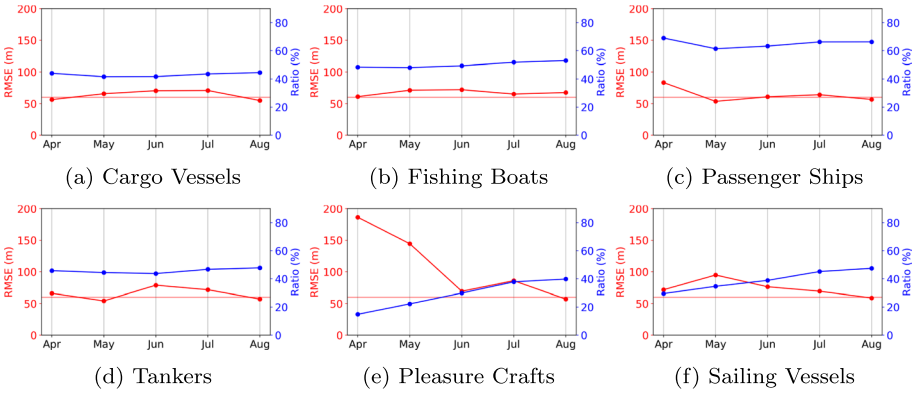


Fig. 7 Performance of Incremental Optimization on the *test sets* for the MS dataset

Pleasure Crafts have different behaviour in different months (recall that the training data spans from March to July). Additionally, Fig. 6 shows that finding parameter values that satisfy the *RMSE* threshold of $\theta = 60$m is easy, as these values are computed quite early during the training phase. Finally, the fluctuations that appear in some of the graphs in Fig. 6 are not troubling since the initial population of the i-th step is chosen as the best individuals from the entire training phase of the $(i-1)$-th step.

Figure 7 presents the performance of incremental optimization on the test sets. For example, the displayed *RMSE* and *Ratio* values of July concern the trajectory compression of the AIS messages of July, using the parameter values that minimized the optimization function on all the previous months (March to June). We can immediately notice that the good results observed during training carry over to the testing. For all but one vessel types, the attained *RMSE* is close to the threshold of 60m, whereas the *Ratio* values do not deviate from the values achieved during training. The exception is Sailing Vessels (Figs. 6f and 7f), where the compression ratio is higher for the summer months. This makes sense, since mobility during this period is increased and in order to maintain an acceptable trajectory reconstruction error we need to keep more of the original data points.

Concerning Pleasure Crafts, we notice a behavior similar to that of Sailing Vessels, where the values of *Ratio* increase as more months are examined (Figs. 6e and 7e). The performance during the training phase is more than satisfactory, where we observe the lowest *Ratio* values across all ship types (Fig. 6e). However, in the early testing phases we notice an unusually high value of *RMSE* (Fig. 7e). This behaviour is consistent with the findings reported in Sect. 5.2. The more training data consumed by the GA, the closer the *RMSE* becomes to the desired value of θ (see, again, Fig. 7e).

5.5 Composite event recognition

Next, we assess the effects of trajectory compression under GA optimization, on composite maritime event recognition as performed by RTEC. The complex events that we examine require spatial information that cannot be extracted directly from the vessels' trajectories.

Table 5 Dataset sources for composite maritime event recognition

Attribute	Brest	Source(s)
Position signals	19M	[36]
Spatio-temporal events	374K	[33]
Fishing areas	263	[28, 41]
Natura 2000 areas	1.2K	[40]
Anchorage areas	9	[8]
Near coast areas	197	[10]
Ports	222	[36]

Table 6 Accuracy of Composite Event Recognition

Composite Event	Precision	Recall	F_1-score
Anchored or Moored Vessel	1.0	1.0	1.0
Drifting Vessel	0.88	0.99	0.93
High Speed Near Coast	0.97	0.96	0.96
Search & rescue operations	0.97	1.0	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
Trawling	1.0	1.0	1.0
Tugging	0.99	1.0	1.0
Vessel under way	0.99	1.0	0.99

For this reason, the data used as input for RTEC is enhanced with information about the location of the vessels, such as proximity to a port or other vessels. This is done by means of spatio-temporal link discovery [33, 37]. Furthermore, additional data are required, such as protected areas (Natura 2000 areas[9]) and anchorage areas. Table 5 summarises the data sources used for composite maritime event recognition.

In the first column of Table 6 we present the composite events under investigation, all of which are durative. 'Ship-to-ship transfer', for example, is said to take place when two ships are stopped in the open sea, closely to each other for a duration longer than a certain threshold value. In order to recognize such events, we must fuse the trajectories of different vessels, as well as compute some spatiotemporal relationships between them (for more information see [33]).

Unlike our previous experiments, the event patterns are not restricted to the 6 vessel types with the most data points. In the composite event recognition tests, we used all vessel types available in the dataset. We restricted attention to the BR dataset, since in this dataset it is possible to achieve an accurate summarization of the entire dataset by using the optimized parameter values of the top-6 vessel types. Some of the vessel types that are not in the top-6 have very similar moving behaviour to a type that is in the top-6; in these cases, we used the GA's optimal parameter values of the latter for compressing the trajectories of

[9] https://ec.europa.eu/environment/nature/natura2000/index_en.htm

Table 7 Efficiency of Composite Event Recognition

RTEC Input	Average Recognition Time per Window (sec)	Standard Deviation (sec)	Worst Time (sec)
BR dataset	3.54	6.28	28.09
BR synopses [Def]	1.62	2.06	14.46
BR synopses [GA]	1.05	1.02	7.85

Table 8 Number of Annotations before and after relabeling

Label	Before Relabeling	After Relabeling
Stop	281,312	219,884
Change in heading	1,548,168	1,966,807
Change in speed	889,219	323,113

the former. Less than 8% of the dataset includes vessel types with movement patterns that differ from those in the top-6. For this small part of the dataset, we used the default compression parameter values. Finally, in the optimization function of the GA (Eq. (4)), the *RMSE* threshold θ was set to 15m.

Table 6 presents the predictive accuracy of RTEC when consuming the compressed dataset which was generated as described above. The ground truth, according to which we calculated predictive accuracy, consists of the composite event intervals computed when RTEC consumes the original uncompressed dataset. For example, the number 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 6 shows that our system achieves perfect scores for most composite events. For the remaining ones, the recognition scores are very close to optimal.

To support online recognition, RTEC operates using a sliding window [5]. Table 7 shows the time required to recognize all events displayed in Table 6, when RTEC operates using a 24-hour sliding window, over different inputs: the original (uncompressed) BR dataset, its synopses under default parameterization ('BR synopses [Def]'), and those optimized by the GA ('BR synopses [GA]'). As expected, operating on compressed trajectories offers very significant performance gains. Additionally, the synopses derived using the GA are more succinct and thus lead to more efficient composite event recognition, as opposed to those produced under the default parameter values.

5.6 Critical point relabeling

Finally, we evaluated the process of critical point relabeling, as described in Sect. 4.5. We used the BR dataset and relabeled the synopses generated by the GA. The parameter values used for the relabeling were the default parameters (see Table 1) since these values were manually chosen using experts' advice and thus represent real-world events more accurately. Table 8 presents the number of critical points before and after

Table 9 Effects of Critical Point Relabeling

Composite Event	Performance change with respect to GA
Anchored or Moored Vessel	-16%
Drifting Vessel	0%
High Speed Near Coast	0%
Search & rescue operations	0%
Loitering	-1%
Piloting	6%
Ship-to-ship transfer	-4%
Trawling	425%
Tugging	0%
Vessel under way	-1%

this relabeling. We observe an increase of 27% in turning points as we intended (the value picked by GA for angle threshold was too high), as well as a 22% drop in the stop events. There is also a sharp reduction by 64% in the 'change in speed' events. Since the default parameters were picked by experts, and thus the relabeled dataset represents simple events more accurately, the significant differences between the two summarizations indicate that before relabeling, the representation of simple events was far from accurate. This is to be expected, since GA is geared towards optimizing summarization performance, and thus picks parameter values that may not be the most suitable in recognizing real world events.

To measure the effect of relabeling on composite event recognition, we considered as ground truth the composite event intervals recognized by RTEC when consuming the uncompressed dataset, where the critical points were labeled according to the default parameters. Then we compared the ground truth with the following two datasets:

1. The composite event intervals produced by RTEC when consuming the compressed dataset, both summarized and labeled with critical points according to the GA's parameters.
2. The composite event intervals produced when consuming the compressed dataset, summarized according to the GA's parameters, but relabeled according to the default parameters.

Table 9 presents the F_1-score percentage difference of each composite event for both datasets. The second column presents how better percentage-wise the relabeled dataset is. The two approaches have similar performance. The notable exception is the *Trawling* event, in which using relabeling led to a vastly better performance. This is due to the fact that *Trawling* heavily depends on 'change in heading' critical points, controlled by the *angle threshold* compression parameter, that takes extremely high values in the parameters generated by the GA. This issue is addressed by the relabeling process that manages to capture many more turning points along trajectories (i.e., the mobility events concerning changes in heading in Table 8), leading to much better predictive accuracy.

6 Summary and future work

Maritime monitoring systems rely heavily on vessel trajectory summarization techniques, in order to effectively support online analytics. We presented an open-source system optimizing vessel trajectory compression, producing trajectory synopses minimizing the approximation error and the compression ratio. Furthermore, it supports incremental optimization and does not rely on expert knowledge; in contrast, the only user input required is the desired approximation error in the resulting synopses. The synopses produced can be used for composite maritime event recognition, allowing for effective maritime situational awareness; this can be improved by the process of relabelling, which requires some expert knowledge to describe simple events such as turning events. A comprehensive empirical evaluation on two real-world AIS 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 predictive accuracy. In future work, we plan evaluate our system on composite maritime event *forecasting*, i.e. the detection of future events before they occur, in order to support proactive decision-making.

Funding This work has received funding from the EU Horizon 2020 RIA program INFORE under grant agreement No 825070 and from the NSF grant number CCF-1563714. We would also like to thank MarineTraffic for providing the MS dataset..

Data availability The BR dataset analyzed during the current study is available in [36], https://doi.org/10.5281/zenodo.1167595. The MS dataset was given to us by MarineTraffic, our partner in the INFORE project, and restrictions apply to the availability of these data, and so are not publicly available. Finally, the sources for data used for composite maritime event recognition are summarized in Table 5.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest

References

1. Agarwal PK, Har-Peled S, Mustafa NH, Wang Y (2002) Near-linear time approximation algorithms for curve simplification. In: ESA. pp 29–41
2. Alevizos E, Artikis A, Paliouras G (2017) Event forecasting with pattern Markov chains. In: DEBS. pp 146–157
3. Alevizos E, Skarlatidis A, Artikis A, Paliouras G (2017) Probabilistic complex event recognition: a survey. ACM Comput Surv 50(5):71:1–71:31
4. Arasteh S, Tayebi MA, Zohrevand Z, Glässer U, Shahir AY, Saeedi P, Wehn H (2020) Fishing vessels activity detection from longitudinal AIS data. In: SIGSPATIAL. pp 347–356
5. Artikis A, Sergot MJ, Paliouras G (2015) An event calculus for event recognition. IEEE Trans Knowl Data Eng 27(4):895–908
6. Cao H, Wolfson O, Trajcevski G (2006) Spatio-temporal data reduction with deterministic error bounds. VLDB J 15(3):211–228
7. Cugola, G, Margara A (2012) Processing flows of information: from data stream to complex event processing. ACM Comput Surv 44(3):15:1–15:62
8. datAcron H2020 ICT-16 Project. https://www.iit.demokritos.gr/projects/datacron/. Accessed 26 Aug 2022

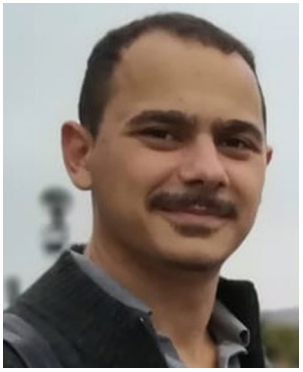
9. Douglas D, Peucker T (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Can Cartogr 10(2):112–122
10. European Environment Agency: Europe coastline shapefile (2013). https://www.eea.europa.eu/data-and-maps/data/eea-coastline-for-analysis-1/gis-data/europe-coastline-shapefile. Accessed 26 Aug 2022
11. Fikioris G, Patroumpas K, Artikis A (2020) Optimizing vessel trajectory compression. In: MDM. pp 281–286
12. Fikioris G, Patroumpas K, Artikis A, Paliouras G, Pitsikalis M (2020) Fine-tuned compressed representations of vessel trajectories. In: CIKM. pp 2429–2436
13. Giatrakos N, Alevizos E, Artikis A, Deligiannakis A, Garofalakis MN (2020) Complex event recognition in the big data era: a survey. VLDB J 29(1):313–352
14. Grez A, Riveros C, Ugarte M (2019) A formal framework for complex event processing. In: ICDT, vol. 127. LIPIcs, pp 5:1–5:18
15. Iphar C, Napoli A, Ray C (2015) Detection of false AIS messages for the improvement of maritime situational awareness. In: OCEANS. pp 1–7
16. Katzouris N, Artikis A (2020) WOLED: a tool for online learning weighted answer set rules for temporal reasoning under uncertainty. In: KR. pp 790–799
17. Kontopoulos I, Chatzikokolakis K, Tserpes K, Zissis D (2020) Classification of vessel activity in streaming data. In: DEBS. pp 153–164
18. Lange R, Dürr F, Rothermel K (2011) Efficient real-time trajectory tracking. VLDB J 20(5):671–694
19. Lin X, Jiang J, Ma S, Zuo Y, Hu C (2019) One-pass trajectory simplification using the synchronous Euclidean distance. VLDB J 28(6):897–921
20. Lin X, Ma S, Zhang H, Wo T, Huai J (2017) One-pass error bounded trajectory simplification. PVLDB 10(7):841–852
21. Liu J, Zhao K, Sommer P, Shang S, Kusy B, Jurdak R (2015) Bounded quadrant system: Error-bounded trajectory compression on the go. In: ICDE. pp 987–998
22. Liu J, Zhao K, Sommer P, Shang S, Kusy B, Lee J, Jurdak R (2016) A novel framework for online amnesic trajectory compression in resource-constrained environments. IEEE Trans Knowl Data Eng 28(11):2827–2841
23. Ljunggren H (2018) Using deep learning for classifying ship trajectories. In: FUSION. pp 2158–2164
24. Long C, Wong RCW, Jagadish H (2014) Trajectory simplification: on minimizing the direction-based error. PVLDB 8(1):49–60
25. Makris A, Kontopoulos I, Alimisis P, Tserpes K (2021) A comparison of trajectory compression algorithms over AIS data. IEEE Access 9:92516–92530
26. Meratnia N, deBy R (2004) Spatiotemporal compression techniques for moving point objects. In: EDBT. pp 765–782
27. Muckell J, Olsen P, Hwang JH, Lawson C, Ravi S (2014) Compression of trajectory data: a comprehensive evaluation and new approach. GeoInformatica 18(3):435–460
28. Natale F, Gibin M, Alessandrini A, Vespe M, Paulrud A (2015) Mapping fishing effort through AIS data. PLoS ONE 10(6):1–16
29. Nguyen D, Vadaine R, Hajduch G, Garello R, Fablet R (2018) A multi-task deep learning architecture for maritime surveillance using AIS data streams. In: DSAA. pp 331–340
30. Patroumpas K (2021) Online mobility tracking against evolving maritime trajectories. In: Artikis A, Zissis D (eds) Guide to Maritime Informatics. Springer
31. Patroumpas K, Alevizos E, Artikis A, Vodas M, Pelekis N, Theodoridis Y (2017) Online event recognition from moving vessel trajectories. GeoInformatica 21(2):389–427
32. Patroumpas K, Pelekis N, Theodoridis Y (2018) On-the-fly mobility event detection over aircraft trajectories. In: ACM SIGSPATIAL. pp 259–268
33. Pitsikalis M, Artikis A, Dreo R, Ray C, Camossi E, Jousselme A (2019) Composite event recognition for maritime monitoring. In: DEBS. pp 163–174
34. Potamias M, Patroumpas K, Sellis T (2006) Sampling trajectory streams with spatiotemporal criteria. In: SSDBM. pp 275–284
35. Potamias M, Patroumpas K, Sellis T (2007) Online amnesic summarization of streaming locations. In: International Symposium on Spatial and Temporal Databases. Springer, pp 148–166
36. Ray C, Dréo R, Camossi E, Jousselme AL, Iphar C (2019) Heterogeneous integrated dataset for maritime intelligence, surveillance, and reconnaissance. Data in Brief 21. https://doi.org/10.5281/zenodo.1167595
37. Santipantakis GM, Vlachou A, Doulkeridis C, Artikis A, Kontopoulos I, Vouros GA (2018) A stream reasoning system for maritime monitoring. In: TIME. pp 20:1–20:17
38. Snidaro L, Visentini I, Bryan K, Foresti GL (2012) Markov Logic Networks for context integration and situation assessment in maritime domain. In: FUSION. IEEE, pp 1534–1539

39. Terroso-Saenz F, Valdés-Vela M, den Breejen E, Hanckmann P, Dekker R, Skarmeta-Gómez AF (2015) CEP-traj: an event-based solution to process trajectory data. Inf Syst 52:34–54
40. Unit Nature & Biodiversity, DG Environment, European Commission: Natura 2000 data - the European network of protected sites (2016). https://www.eea.europa.eu/data-and-maps/data/natura-13. Accessed 26 Aug 2022
41. Vespe M, Gibin M, Alessandrini A, Natale F, Mazzarella F, Osio GC (2016) Mapping EU fishing activities using ship tracking data. J Maps 12(sup1):520–525. https://doi.org/10.1080/17445647.2016.1195299
42. Vouros GA, Doulkeridis C, Santipantakis G, Vlachou A, Pelekis N, Georgiou H, Theodoridis Y, Patroumpas K, Alevizos E, Artikis A, Fuchs G, Mock M, Andrienko G, Andrienko N, Ray C, Claramunt C, Camossi E, Jousselme AL, Scarlatti D, Cordero JM (2018) Big data analytics for time critical maritime and aerial mobility forecasting. In: EDBT. pp 612–623
43. Wolfson O, Sistla A, Chamberlain S, Yesha Y (1999) Updating and querying databases that track mobile units. Distrib Parallel Dat 7(3):257–287
44. Zhang D, Ding M, Yang D, Liu Y, Fan J, Shen HT (2018) Trajectory simplification: an experimental study and quality analysis. PVLDB 11(9):934–946

Giannis Fikioris is a third year PhD student in the theory group at Cornell University. He received his BSE from the department of Electrical and Computer Engineering, National Technical University of Athens. He completed his undergraduate thesis with Dimitris Fotakis, titled Perturbation Stability for Mechanism Design. From July 2019 to August 2020, he was working in NCSR "Demokritos", in the Complex Event Recognition Group, advised by Alexander Artikis.

Kostas Patroumpas joined the Information Management Systems Institute at Athena Research Center in 2012 and has collaborated in many research projects. Previously, he worked in the industry as developer, IT manager, and GIS consultant. He has served on the program committees of several conferences and has more than 60 publications in the areas of data stream processing, trajectory data management, spatial analytics, and geospatial data integration.

Alexander Artikis is an Associate Professor in the University of Piraeus and a Research Associate in NCSR "Demokritos". He has published over 100 papers in the field of Artificial Intelligence and, according to google scholar, his h-index is 35. Alexander has been serving as a member of the programme committees of several international conferences, such as IJCAI and AAAI; he was the PC co-chair of DEBS 2021.



Manolis Pitsikalis is currently a PhD student at the University of Liverpool. He received his BSc from the department of Informatics and Telecommunications at the National Kapodistrian University of Athens. In 2017 he joined the Complex Event Recognition group at NCSR Demokritos as a research associate. His general research interests evolve around temporal logics, event recognition, temporal phenomena description languages and maritime monitoring.

Georgios Paliouras is a research director and head of the Artificial Intelligence Lab SKEL of NCSR "Demokritos" in Greece. He is performing basic and applied research in Artificial Intelligence with an emphasis on Machine Learning and innovative applications. Among others, he has contributed to research in the field of Complex Event Recognition and has co-founded the corresponding SKEL group.

Authors and Affiliations

Giannis Fikioris[1] · Kostas Patroumpas[2] · Alexander Artikis[3,4] · Manolis Pitsikalis[5] · Georgios Paliouras[4]

Kostas Patroumpas
kpatro@athenarc.gr

Alexander Artikis
a.artikis@unipi.gr

Manolis Pitsikalis
E.Pitsikalis@liverpool.ac.uk

Georgios Paliouras
paliourg@iit.demokritos.gr

[1] Department of Computer Science, Cornell University, Ithaca, USA

[2] Information Management Systems Institute, Athena Research Center, Marousi, Greece

[3] Department of Maritime Studies, University of Piraeus, Piraeus, Greece

[4] Inst. of Informatics & Telecommunications, NCSR Demokritos, Athens, Greece

[5] Department of Computer Science, University of Liverpool, Liverpool, UK