Online Distributed Maritime Event Detection & Forecasting over Big Vessel Tracking Data

Marios Vodas[1], Konstantina Bereta[1]
Dimitris Kladis[1], Dimitris Zissis[1,2]
[1]firstname.lastname@marinetraffic.com, [2]dzissis@aegean.gr
[1]*MarineTraffic*, [2]*University of the Aegean*

Elias Alevizos, Emmanouil Ntoulias, Alexander Artikis
manosntoulias@iit.demokritos.gr
alevizos.elias,a.artikis@iit.demokritos.gr
NCSR Demokritos

Antonios Deligiannakis, Antonios Kontaxakis
Nikos Giatrakos
adeli, akontaxakis, ngiatrakos@athenarc.gr
Athena Research Center

David Arnu, Edwin Yaqub
Fabian Temme
darnu,eyaqub,tftemme@rapidminer.com
RapidMiner GmbH

Mate Torok, Ralf Klinkenberg
mtorok@rapidminer.com
rklinkenberg@rapidminer.com
RapidMiner GmbH

***Abstract*—We present a Maritime Situational Awareness (MSA) framework for detecting and forecasting maritime events (e.g., illegal fishing) over streams of Big maritime Data. The architecture of the MSA framework relies on the following state-of-the-art components: (i) the Maritime Event Detector which uses data-driven distributed techniques deployed on a computer cluster to detect maritime events of interest in an online, real-time fashion, (ii) the Complex Event Forecasting module, which implements state-of-the-art distributed Complex Event Forecasting techniques for maritime data, (iii) the Synopses Data Engine component, that creates synopses of maritime data improving the scalability of the framework and (iv) the streaming extension of a popular data science platform, namely RapidMiner Studio, that integrates all the above, allowing users to graphically design and rapidly implement Big Data analytics pipelines which can be deployed transparently on top of distributed architectures.**

I. Introduction

The Maritime Situational Awareness (MSA) involves the efficient utilisation of maritime surveillance means in order to assist in the understanding of the global maritime activities. Since more than 90% of the global trade is carried by vessels [1], improving the global MSA is crucial. The development of the Automatic Identification System (AIS) and its standardisation by IMO in 2018 [2] was disruptive. AIS transponders transmit AIS messages that contain dynamic, navigational information about vessels (i.e., location, speed, heading, course, etc.) that hold for a given timestamp as well as static information (i.e., identifier, name, vessel type, dimensions, etc.). These messages are collected by AIS receivers that are installed aboard, ashore or on satellites (SAT-AIS). Although all passenger ships and ships with more than 300 gross tonnage bear an AIS transceiver, vessels may switch off their transponders when engaging in illegal activities (e.g., smuggling, illegal fishing, trafficking, illegal trans-shipments).

The rapidly increasing availability of multiple sources of data in the maritime industry, together with the recent advances in the areas of Big Data and AI have unlocked opportunities to derive new knowledge, otherwise hidden in the vast maritime data silos. This motivation has fueled the development of new data science techniques for maritime data [3].

However, significant challenges still need to be addressed in the industrial maritime setting: (i) The increasing availability of maritime data sources resulted in Big maritime Data that have surpassed the limits of centralised data processing architectures. Maritime data are huge in *volume* and come in a streaming fashion, such as high-speed streams of AIS messages (*velocity*). Moreover, they are available via a *variety* of sources (i.e., AIS, acoustic, satellite image data) that call for different data cleaning methods (to deal with the lack of *veracity*) in order to be usable. MarineTraffic[1] owns the largest AIS network worldwide and processes nearly 1 Billion AIS messages accumulating ∼100 GB of AIS data, every day. This data is complemented by other data sources such as satellite image data of tens of TBs. (ii) Maritime data science workflows can become very complex and not easy to maintain or update. (iii) Different technologies/platforms are suitable for different data science tasks. In turn, data scientists need to know the specifications of different Big Data platforms, languages and libraries. (iv) Most MSA applications are bound to technologies and tightly coupled/monolithic architectures that often become deprecated and are cumbersome in new implementations of evolving application information needs.

We present a novel MSA framework for detecting and forecasting maritime events that addresses the aforementioned Big maritime Data challenges. Figure 1 illustrates the components of our MSA architecture (described in Section II). Some early efforts on developing individual components of this architecture have been presented independently, such as the Synopsis Data Engine in [4], a *non-distributed*[2] version of the Complex Event Processing and Forecasting module (CEP/CEF) in [5], [6] or part of the Maritime Event Detector in [7], [8]. In the current work we present for the first time a full-fledged, scalable, distributed architecture in which the var-

[1]https://www.marinetraffic.com
[2]The terms parallel and distributed are used interchangeably.

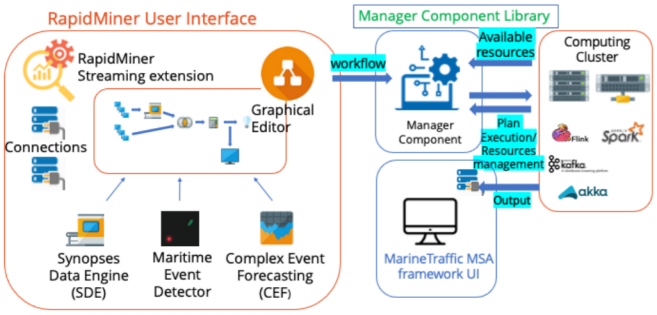

Fig. 1: Components of the MSA architecture.

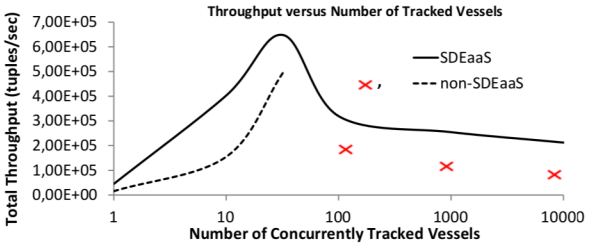

Fig. 2: SDE utility in the MSA Setting

ious components are integrated and interplay in order to tackle the real-world problem in an industrial MSA environment. Moreover, this paper contains not previously documented parallel implementations of all architectural components. With these components in place, our MSA framework goes beyond baseline MSA solutions [3] in the following dimensions: (i) it is the first end-to-end MSA architecture that fully scales out the computation, since all its modules distribute the processing load to the machines available in MarineTraffic's corporate data centers, (ii) it enables both the real-time detection and the forecasting of complex maritime events ahead of time, allowing for proactive decision-making, (iii) it includes provisions for operating on data synopses to boost scalability in extreme-scale scenarios that truly arise in practice, and (iv) it enables users to graphically and, thus, rapidly design MSA related workflows, drastically cutting down time-to-production for new MSA services. Our MSA framework is currently being validated operationally in order to significantly enhance the existing suite of services provided by MarineTraffic.

II. Architecture

The architecture realising the Maritime Situational Awareness framework include the following components that are displayed in Figure 1:

The Streaming extension of RapidMiner Studio. We extend a well-known data science platform, RapidMiner Studio,[3] to enable users graphically design and execute data science workflows over streaming Big Data platforms such as Apache Flink, Apache Spark and Big Data toolkits such as Akka, without the need to write custom code. In the case of the MSA framework, this component is necessary as it seamlessly incorporates complex stream processing operators, as the ones described below, while integrating different Big Data platforms, transparently. Our experience on real-world MSA scenarios shows that this component reduces the time-to-production for new MSA workflows, from weeks to minutes.

The Synopses Data Engine (SDE) is an open source component[4] used for maintaining summaries of data, instead of the original input streams, to allow rapid analytic results delivery. Data summaries are representative views (samples, expected values, counts, frequency moments) of important aspects of the incoming data, approximated with predefined accuracy guarantees. By operating on compact data summaries, the SDE can boost the horizontal scalability of MSA workflows, i.e., scaling the computation with the volume and velocity of incoming streams.

What is more, the SDE significantly enhances vertical scalability, i.e., scaling the computation to huge numbers of streams. This is particularly important in MSA scenarios where thousands of vessels, each producing a separate AIS data stream, are being concurrently monitored. The SDE [4] follows a Synopses-as-a-Service (SDEaaS) paradigm: it is deployed as a service – constantly running job – over one or more computer clusters. This unique SDEaaS design allows the SDE to serve on-the-fly requests for maintaining new synopses, plugging code for new synopses in the SDE Library, perform ad-hoc and continuous queries and a number of other important operations, with zero downtime for the MSA workflows that use summaries. The SDE implements a wide variety of data summarization techniques which can be applied in different MSA use cases. An example of MSA-specific data synopsis is the STSampler synopsis [9], which simplifies vessel trajectories by keeping only a carefully-crafted sample of important vessel positions, instead of the original streams. No or little change of position, heading or direction over time favors exclusion of a position from the trajectory sample.

To showcase the performance benefits of the SDEaaS, Figure 2 compares our SDEaaS design to non-SDEaaS approaches (i.e., simple synopses libraries), using the STSampler data synopsis. We design an experiment, over a cluster with 40 CPUs, where we start with maintaining 2 STSampler synopses. Then, we express demands for monitoring more vessels up to 10000. We do that without stopping the already running synopses. Figure 2 shows the sum of throughputs (number of processed tuples/sec) of all running jobs for non-SDEaaS and the throughput of SDEaaS. non-SDEaaS does not allow on-the-fly requests to running jobs for maintaining new synopses. Instead, SDEaaS starts new jobs for new synopses assigning at least 1/40 threads to each. Hence, the ∼40 threads in our cluster are depleted and × signs in Figure 2 denote that non-SDEaaS cannot maintain more than 40 synopses. SDEaaS has no

[3]https://rapidminer.com/products/studio/
[4]https://bitbucket.org/infore_research_project/6.1-sde-release/src/master/

such limit since it initiates new tasks to a single running service (job) at runtime, with zero downtime. SDEaaS also exhibits higher throughput than non-SDEaaS due to fine-grained resource utilization at the task, instead of thread, level.

The Maritime Event Detector. The main goal of a complex event processing system is to detect interesting maritime activity patterns occurring within a stream of events, coming from sensors or other devices. The input to a CEP system consists of two main components: a stream of events, also called simple derived events (SDEs), usually in the form of tuples with numerical and categorical attributes; and a set of patterns that define relations among the SDEs. Instances of pattern satisfaction are called Complex Events (CEs). The output of the system is another stream, composed of the detected CEs. Typically, CEs must be detected with very low latency, which, in certain cases, may be in the order of a few milliseconds.

The Maritime Event Detector [7],[8] is one of the few distributed CEP engines [10] and, to our knowledge, the only one of industrial scale in the maritime domain. This component processes large streams of real-time and historical maritime data (i.e., AIS data containing the locations of vessels together with other navigational information) in order to detect maritime events related to vessel behavior (e.g., vessel with AIS transponder switched off for a period of time, vessels in proximity, deviation from common routes, etc.) or activities (e.g., ship-to-ship transfer, bunkering, fishing, vessel entering shallow waters or an area of interest). The Maritime Event Detector is deployed on an Akka[5] cluster enabling distributed (parallel) processing of vessel tracking data describing the navigational status of the worldwide fleet. The actor model employed in Akka is utilized and a ship is mapped to an actor instance which stores and updates the latest state for the ship as the stream is consumed.

The Complex Event Processing and Forecasting Component is the first that exploits the virtues of distributed processing in the context of Complex Event Forecasting (CEF). Our CEF Component constitutes an integrated version of the open source tool Wayeb[6]. It employs symbolic automata as a computational model for pattern detection and Markov models for deriving a probabilistic description of a symbolic automaton. Wayeb accepts as inputs a set of patterns defined by analysts in the form of symbolic regular expressions and a stream of input events. It then attempts to detect instances of patterns' satisfaction upon the input stream with minimal latency. In its forecasting mode, it additionally produces predictions about when a certain pattern is expected to occur, thus widening the decision margins for analysts and users. The Wayeb tool is documented in more detail in [5], [6], [11], where a detailed empirical analysis on maritime data is presented.

[5]https://akka.io
[6]https://github.com/ElAlev/Wayeb

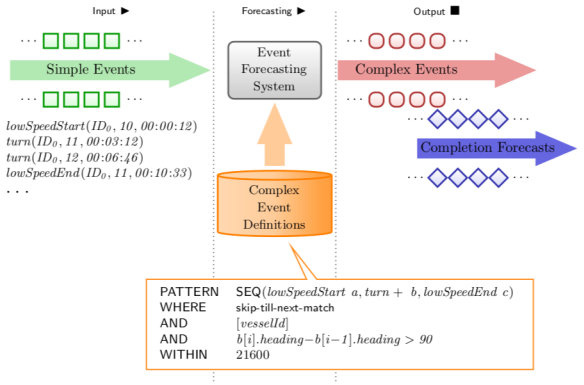

Fig. 3: Complex Event Processing/Forecasting system overview.

Note that CEP/F systems are generic, domain-independent solutions for processing real-time streams of events and detecting interesting patterns, defined in a declarative language, much like SQL in traditional database management systems. They are thus more expressive tools compared to those typically employed in fields such as time series processing/forecasting or event sequence detection/prediction.

In the MSA framework, the role of Wayeb is crucial, as it is able to forecast complex maritime events (e.g., illegal fishing), i.e., events which are composed of multiple simple events (e.g., vessel with its AIS transponder switched off, moving with decreased speed, etc.). As an example of a pattern which is of high value to maritime analysts, see the following definition:

$$\begin{aligned} illegalFishing := \; & (VesselType = Fishing \wedge \\ & speed > 1.0 \wedge speed < 9.0) \; ; \\ & (EventType = AISOff) \; ; \\ & (EventType = AISOn \wedge \\ & speed > 1.0 \wedge speed < 9.0) \end{aligned}$$

The definition described above is a possible definition for the activity of *illegal fishing*, where the symbol ; denotes sequence/concatenation, i.e., events separated by ; are assumed to follow one another in time. Typically, vessels engaged in this activity switch off their AIS transponders. Therefore, the above definition detects behavioral patterns of fishing vessels, where they have their AIS equipment switched off and their speed is within the typical speed range of vessels while they are actually fishing. A high-level overview of the CEF component is shown in Figure 3.

For the MSA framework, we use a distributed Flink-based version of Wayeb, as opposed to its previous, centralised version [5], [6], thus allowing for a distributed implementation of both its training (learning of the probabilistic model) and its testing (the online emission of forecasts as events are consumed) phase. Contrary to FlinkCEP though, the

built-in CEP module of Flink [7], Wayeb offers forecasting capabilities. It additionally offers an expressive framework for defining patterns, with formal and compositional semantics (a feature generally lacking in most CEP solutions [10]), as it is based on symbolic automata which have nice closure properties [12]. Wayeb's performance, in terms of throughput, is typically above 100K events/second for a single pattern or multiple patterns using parallelization [11].

The MSA front-end. The MSA user-interface displays the following information that is available through Kafka topics: (i) Real-time AIS data provided by MarineTraffic. (ii) Real-time and past detected events. (iii) Complex maritime events forecast by the Complex Event Forecasting component.

III. Real-world workflow

The MSA framework described in this paper targets to the following two broad categories of users: (i) the maritime data scientists that might be working at maritime intelligence companies (like MarineTraffic), who design and deploy complex maritime workflows involving operators executed on different Big Data platforms (i.e., Kafka, Akka, and Flink), and (ii) MSA end-users (e.g., coast-guard or vessel traffic officers, shipping companies, defence agencies etc.) that need to be notified about maritime events in time in order to take immediate actions and would like to explore the results of maritime analytics displayed on a map using a user-friendly UI. In the following, we describe a real-world scenario that corresponds to the two broad categories of end-users described above. A video demo is also available online[8].

First, we explain a procedure followed by a data scientist working at a maritime intelligence company to design and deploy a workflow that forecasts complex maritime events using AIS data (e.g., illegal fishing), as input. The workflow is shown in Figure 4. The first step is to define the input data in the RapidMiner Studio. In the example of the video demo provided above, we use a real-world AIS dataset from MarineTraffic of 18GB size, that contains approximately 220 million positions in the Mediterranean during the time period 1/03/16-31/8/16. The AIS dataset contains navigational information of vessels (e.g., speed, location, navigational status, etc). Next, we use the Synopsis Data Engine to create simplified trajectories thus reducing the size of the data to be processed, which, after some pre-processing, are forwarded to the Maritime Event Detector that runs in distributed mode and is deployed on an Akka cluster owned by MarineTraffic. The Maritime Event Detector detects simple events such as `ais-off` events, that signify that a vessel has potentially switched off its transponder or a period of time. Other events that can be detected using this component are `proximity` events (i.e., when two vessels are too close given a distance threshold), `route deviation` events (i.e., when a vessel deviates from a common route given its port of origin, its destination, and the vessel category), events indicating that a vessel navigates into `shallow waters`, that could prevent potential groundings, and others. These events can be considered as indicators of potential illegal activities or accidents. The dataset used in the demonstration (in the online demo and video) contains over 18K proximity and 3M ais-off events.

These events, together with the simplified trajectories, are then consumed by the Complex Event Forecasting component. The CEF component uses this data to identify/forecast patterns of complex events. For example, once a vessel decreases its speed, switches off its transponder and then switches it back on, while being anchored in the meantime (i.e., moving in low speed), an illegal fishing event is triggered, as explained in the example of Section II). However, different patterns may exist for different areas (e.g., fishing vessels may have different speed in different parts of the sea while fishing). Using RapidMiner Studio, the user can easily change the parameters of the CEF component that correspond to different patterns (e.g., change the certainty threshold, the speed, or change the whole pattern using for example different illegal fishing patterns for different areas) and then re-execute the workflow. The dataset contains over 18K proximity and approximately 3 million ais-off events.

Eventually, the forecast and detected simple and complex events, together with AIS data are pushed to three separate Kafka topics. These Kafka topics can then be consumed by a number of applications. For example, this data could be consumed by an end-user application for a Defence Agency, another one for Vessel Traffic officers that monitor traffic in and around a port, a mobile application designed exclusively for coast-guard authorities, and many more. We have implemented our own end-user general-purpose application that incorporates functionalities that meet many of the common needs of the aforementioned user segments. In the following, we will describe typical use-case scenarios followed by end-users that need to monitor the maritime domain and identify uncommon events on a daily basis.

First of all, the MSA UI supports two views: The historical data view and the real-time view. The users select (according to the permissions they have been granted) which mode to log into. The first mode contains a historical dataset (like the one described in the demonstration video), allowing users to perform historical analysis about the simple and complex events that occurred in an area over a period of time. The real-time mode performs online maritime event detection and forecasting following the process described above.

Once a user has logged into the real-time mode of a system, they are able to zoom into an area of interest and see the events, simple or complex, occurring in this area. For example, they can spot the ais-off events that are triggered when a vessel intentionally or not switches-off its transponder, as shown in Figure 5. In the example shown in Figure 5, an ais-off event was detected after its occurrence. For some end-users, however, especially the ones that work in the maritime security domain, it is crucial to be able to forecast events, i.e., detect them before they actually happen. Figure 6 shows a forecast

[7]https://ci.apache.org/projects/flink/flink-docs-release-1.13/docs/libs/cep/
[8]https://www.youtube.com/watch?v=q2wxlgLjjiQ

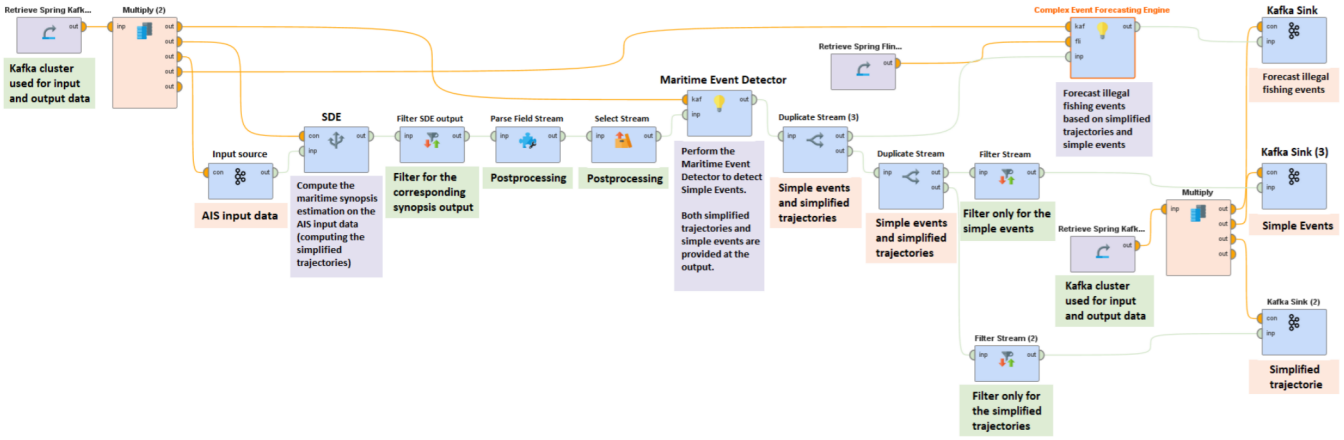

Fig. 4: MSA workflow in RapidMiner Studio

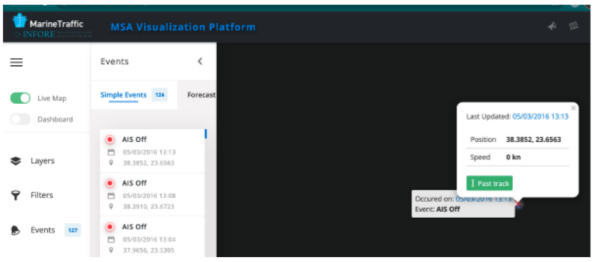

Fig. 5: MSA framework UI: Spotting AIS-OFF (simple) detected events

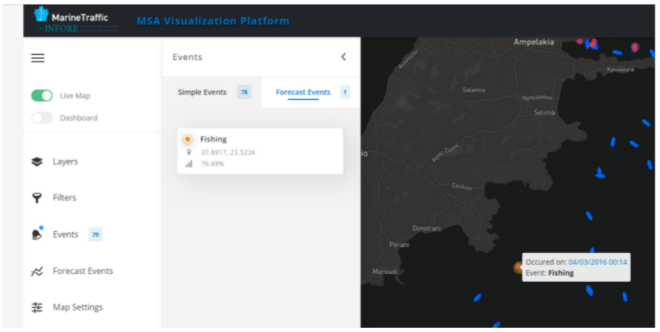

Fig. 6: MSA framework UI: Spotting fishing forecast (complex) events

fishing event. Then, the end-users are able to inspect further the events (detected or forecast, simple or complex) by clicking on the event on the map. In this way, they will be able to see more information apart from the location of the event, such as the time, and more importantly the past track of the vessel, which is the trajectory formed by the locations of the vessel before the occurrence of the event.

Traditionally, end-users try to process AIS data, using different tools for processing and visualisation (e.g., GIS, the MarineTraffic website), while trying to identify common illegal patterns combining a lot of manual work, possibly semi-automatic workflows, and expert knowledge. However, this approach lacks scalability, it highly depends on the skills and experience of the maritime officer, and it is also limited by it, as only well-known patterns of events can be identified. In our approach, we offer data-driven insights to the end user, being able to (i) detect complex maritime events with accuracy ranging from 83% to 97% (depending on the event), (ii) forecast complex maritime events over MarineTraffic AIS data of high velocity (nearly 12K messages/second) and (iii) display them in an integrated way via a web-based UI. This enables end-users to take informed decisions timely and take actions, such as sending patrolling vessels for inspection, within a few minutes, instead of hours/days (depending on the event).

IV. Conclusions and Future Work

In this paper we presented a Maritime Situational Awareness (MSA) framework for detecting and forecasting maritime events (e.g., illegal fishing) over streams of big vessel tracking data. The framework relies on maritime data synopses, that improve the scalability of maritime data science workflows by reducing the size of maritime data to be processed, as well as on techniques for detecting and forecasting maritime events, such as ais-off, fishing, etc. These components are nicely integrated via a user-friendly interface by using the RapidMiner Studio with its streaming extension, reducing the time to design and execute complex maritime workflows that can be deployed transparently on top of distributed architectures from days/weeks to minutes. The output of these workflows can then be consumed by a maritime application. Using the MSA application described in this paper, the end-user can inspect the detected/forecast simple and complex events as well as the involved vessel(s) and the past track of the event, providing valuable information about a vessel's movement around the time of the event. Using this tool, a coast-guard officer, for example, could easily monitor events and investigate arbitrary information (e.g., past track) of involved vessels, taking informed decisions, such as sending patrolling vessels

for inspection.

In future work, we plan to extend the presented workflow with more data sources and operators. More specifically, we plan to fuse AIS vessel tracking data from AIS with data coming from non-collaborative maritime reporting systems, such as satellite imagery, acoustic data, data coming from surveillance cameras, etc. We also plan to extend the MSA application to a robotics system that automatically sends autonomous unmanned vehicles to an area of interest for inspection (e.g., an area where an event such as illegal fishing occurred or it is about to occur).

ACKNOWLEDGMENTS

This work was funded by the EU Horizon 2020 Research and Innovation (RIA) program INFORE (GA No 825070).

REFERENCES

[1] Z. Wan, J. Chen, A. E. Makhloufi, D. Sperling, and Y. Chen, "Four routes to better maritime governance," *Nature*, vol. 540, no. 7631, pp. 27–29, 11 2016.

[2] IMO, "Technical characteristics for an automatic identification system using time division multiple access in the vhf maritime mobile frequency band," ITU, Tech. Rep., 2017. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-5-201402-I!!PDF-E.pdf

[3] M. Riveiro, G. Pallotta, and M. Vespe, "Maritime anomaly detection: A review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, p. e1266, 05 2018.

[4] A. Kontaxakis, N. Giatrakos, and A. Deligiannakis, "A synopses data engine for interactive extreme-scale analytics," in *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, M. d'Aquin, S. Dietze, C. Hauff, E. Curry, and P. Cudré-Mauroux, Eds. ACM, 2020, pp. 2085–2088.

[5] E. Alevizos, A. Artikis, and G. Paliouras, "Event forecasting with pattern markov chains," in *DEBS*. ACM, 2017, pp. 146–157.

[6] ——, "Wayeb: a tool for complex event forecasting," in *LPAR*, ser. EPiC Series in Computing, vol. 57. EasyChair, 2018, pp. 26–35.

[7] D. Zissis, K. Chatzikokolakis, G. Spiliopoulos, and M. Vodas, "A distributed spatial method for modeling maritime routes," *IEEE Access*, vol. 8, pp. 47 556–47 568, 2020.

[8] I. Kontopoulos, K. Chatzikokolakis, K. Tserpes, and D. Zissis, "Classification of vessel activity in streaming data," in *DEBS '20: The 14th ACM International Conference on Distributed and Event-based Systems, Montreal, Quebec, Canada, July 13-17, 2020*, J. Gascon-Samson, K. Zhang, K. Daudjee, and B. Kemme, Eds. ACM, 2020, pp. 153–164.

[9] M. Potamias, K. Patroumpas, and T. Sellis, "Sampling trajectory streams with spatiotemporal criteria," in *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, 2006, pp. 275–284.

[10] N. Giatrakos, E. Alevizos, A. Artikis, A. Deligiannakis, and M. N. Garofalakis, "Complex event recognition in the big data era: a survey," *VLDB J.*, vol. 29, no. 1, pp. 313–352, 2020.

[11] E. Ntoulias, E. Alevizos, A. Artikis, and A. Koumparos, "Online trajectory analysis with scalable event recognition," in *EDBT/ICDT Workshops*, ser. CEUR Workshop Proceedings, vol. 2841. CEUR-WS.org, 2021.

[12] L. D'Antoni and M. Veanes, "The power of symbolic automata and transducers," in *CAV (1)*, 2017.