Experimental Comparison of Complex Event Processing Systems in the Maritime Domain

Alexandros Troupiotis-Kapeliaris
Institute of Informatics & Telecommunications
National Centre for Scientific Research "Demokritos"
Athens, Greece
alextroupi@iit.demokritos.gr

Konstantinos Chatzikokolakis
MarineTraffic
Athens, Greece
konstantinos.chatzikokolakis@marinetraffic.com

Dimitris Zissis
Department of Product and Systems Design Engineering
University of the Aegean
Syros, Greece
dzissis@aegean.gr

Elias Alevizos
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
Athens, Greece
ilalev@di.uoa.gr

***Abstract*—Complex Event Processing (CEP) 's main purpose is recognizing interesting phenomena upon streams of data. So its only natural that it would find applications in the maritime domain, where detecting vessel activity plays an important role in monitoring movement at sea. In this study we briefly examine the field of Complex Event Processing; we present two CEP implementations, one based on machine learning techniques and a rule-based system modeled with Event Calculus. Finally, we evaluate their ability in modeling activities that involve multiple vessels, by comparing their results on real-life examples.**

***Index Terms*—event streaming, maritime analytics, complex event processing, event recognition, pattern matching**

I. Introduction

As more than 80 percent of the global trade is conveyed by sea [1], maritime transport plays a fundamental role in global merchandise and manufacturing supply chain, making thus the need for monitoring and analysing operations at sea (e.g., ship-to-ship transfers, bunkering, tugging, piloting, etc.) crucial so as to improve efficiency, reduce waiting times and minimize operational risks. Although in the past it was hard to perform such an analysis due to lack of the necessary equipment, today multiple maritime surveillance systems that broadcast vessel's information periodically and track other nearby vessels are available on board. Such is the Automatic Identification System (AIS), a self-reporting and cooperative system that all the vessels above 299 Gross Tones were enforced to carry on board for safety reasons after July 2002, following the International Maritime Organization's International Convention for the Safety of Life at Sea (SOLAS) [2]. AIS was initially designed to help vessels' officers monitor maritime traffic in their vicinity and to assist maritime authorities oversee a specific area of interest (e.g., a port, a canal, etc.). However, the fact that AIS relies on an open communication protocol and that broadcast data can be received through terrestrial and satellite communication, enable vessel tracking at a global scale. Several online tools such as MarineTraffic[1] are collecting the transmitted information to provide maritime transportation information services to the general public, making data driven knowledge discovery now possible.

Mining voluminous surveillance datasets to discover frequent patterns and provide accurate insights on vessels' whereabouts is a major step towards digitization of the shipping industry. Even though several data mining techniques that rely on frequent pattern discovery [3], trajectory clustering [4] and vessel classification [3], [5]–[7] have been proposed in the literature, the majority of those follow a batch processing methodology as their focus is on extracting knowledge from historical data, completing the processing workflow within several hours. On the other hand, less attention has been given in handling information coming from high-throughput (and often multiple) sources in true real time and streaming conditions. Data streams can be huge, rapidly changing, infinite and possibly temporally ordered series of real time data that are often collected from distributed networks. Querying data to extract spatial relations such as intersection and distance, or to calculate predicates (e.g., contain, within, etc.) among entities on data streams is a highly complex challenge, as such queries are executed continuously in real time over windows of recent data. Data stream processing algorithms are typically evaluated in terms of memory consumption, CPU time and number of times the algorithm needs to scan the data. Pattern discovery algorithms should rely on single-scan, online, multi-dimensional stream processing and analysis methods [8], while machine learning techniques (e.g., classification, regression, clustering, etc.) should be incremental and adaptive, including forgetting mechanisms such as sliding windows or fading factors to achieve fast and efficient change detection [9].

Data stream processing algorithms have been applied in

[1] www.marinetraffic.com

various fields including online transactions [10], social media analysis [11]–[13] and healthcare [14], but less attention has been given in shipping. In this domain pattern discovery algorithms focus on vessel spoofing incidents detection [15], simple events processing such as proximity and route deviations [16], or trajectory prediction [17]. The focus of our study is on detecting behaviors through event-patterns within the data streams; namely Complex Event Processing (CEP). More precisely, the goal of CEP algorithms is to recognize events of a higher nature, comprised of simple input events, that express the occurrence of interesting phenomena. These Complex Events are defined by input events along with additional conditions upon their attributes, their relations within the stream or both. CEP on the maritime domain may include the detection of illegal activity, such as fishing in prohibited areas, or reporting on ship-to-ship transfer of goods [18], [19]. Deciding on a CEP engine proves to be a non-trivial task, as different architectures and pattern modeling techniques have been proposed over the years. In our work we focus on two systems: a machine learning based vessel activity classification method developed in 2019 for maritime data and an open source agnostic system, Run-time Event Calculus (RTEC). In order to compare the capabilities, expressiveness and fitness for maritime applications we perform CEP for three types of activities in which two or more vessels are involved at a time. More specifically, the activities this paper focuses on are ship-to-ship transfer, tugging and piloting. We use a real-life dataset of AIS messages and evaluate each system in terms of accuracy based on ground truth given on the data.

The rest of the paper is structured as follows: we present related work regarding streaming and CEP applications (Section 2). Afterwards, we briefly show the capabilities of the CEP systems in question (Section 3) and present the characteristics of our input data as well as the complex events we use in order to compare our systems (Section 4). Finally, we display and discuss the empirical evaluation of the compared systems (Section 5) and conclude our work giving also the directions for future work (Section 6).

II. Related Work

A. Data Streaming

As numerous applications process data that are being produced in the form of a stream, several studies have attempted to determine and define the basic concepts of this domain [20]. In general, information in these streams is represented as an, often ordered, series of events; with each event being a tuple of data accompanied with the timestamp of its occurrence. Different techniques have been proposed for processing data streams in order to effectively extract useful information [21] such as sampling or similarity mining from these streams. Real-time applications in the big data era require low latency in responding time in order to handle the great amounts of data being produced every second. In many domains, such as security or finance, information becomes less valuable with time, resulting in additional requirements of responsiveness. For that purpose several, generic and easy-to-use languages have been implemented [22].

B. Complex Event Processing

The domain of CEP, or Complex Event Recognition in some works, focuses on detecting patterns of input events on streams of data. This task may be interpreted as a continuous query processing task with syntax similar to database queries [23], [24] or be implemented using automata theory, logic or other techniques [25], [26]. The need of having expressive languages to describe behaviors based on data streams is explained by Sharghi and Sartipi in [27]. Furthermore, the main issue that each system attempts to resolve is trying to balance between an expressive enough dialect for patterns, together with ensuring the use of high-performance methods for calculating such patterns [28].

While several studies have been conducted in order to present and compare the capabilities of CEP systems [26], [29], [30], most of them have the form of a survey and focus on the operators available by each system; thus providing an enumeration of the operator each one supports and their methods of computation. An extensive and more elaborate analysis of the patterns CEP systems can model was presented in 2009 [24]; with multiple systems included in order to provide a complete picture of all system options at the time. More recently, Zhang et al. [31] have explored an aspect of CEP languages rarely provided: the language's semantics and the relations to its complexity. Moreover, a performance comparison of its proposing language and some widely used CEP systems is presented, using character sequences as patterns. Furthermore, the RTEC system, which we also include in our study, was compared to the most used open-source CEP engine at the time, Esper, in the work by Alevizos and Artikis [32]. Besides an empirical evaluation of the two systems, focusing on the latency of queries, this work provides an analysis on the translation methods of RTEC's constructs into Esper's dialect. The experiments of the latter paper were performed with data and patterns used for city transport management purposes.

CEP in the maritime domain has been the main focus of few studies over the years. Ivan Zappia et al. describes an attempt to monitor dangerous goods through maritime transportation, using an SQL-like CEP language [33]. Recognizing vessel behavior through AIS messages has been explored in works such as [34] and [35]. The former work examines detecting low and high speed movement of vessels and the avoidance of collisions between vessels, while the latter proposes an architecture for recognizing threats for a vessel though the messages it transmits. Furthermore, the use of the RTEC system for maritime purposes was discussed in [36], focusing on the runtime performance of the system, and more extensively in [19]. The paper by Pitsikalis et al. describes multiple complex events for modeling vessel activities and provides an evaluation of the system's results compared to a expert-given ground truth for data of a six month period that highlight the efficiency of its methods. Finally, approaching complex event detection as a trajectory classification problem was presented

in [37] and was later revisited in 2017 [38] and 2018 [39]; with these studies focusing mainly on the representation of trajectories and the appropriate machine learning techniques for such data.

III. Background

A. Anomaly Detection Service

Intended for maritime applications the solution developed by Chatzikokolakis et al. was presented in 2019 [16]. The platform incorporates two components: the Akka based system that determines complex events regarding the movement of vessels and a Kafka model that enables for the former to act as a distributed system. The Akka system uses a database of information on common trajectories to detect anomalies on the vessels' movements. This database is created using machine learning algorithms on historical data of vessel trajectories [4]. The results are sets of polygons that define a normal movement trajectory of a vessel, based on the vessel type along with the departure and arrival ports. These polygons are accompanied with additional information regarding the movement of vessels within, such as velocity thresholds and expected time spent inside.

Akka actors are the primary component of any Akka architecture and are able to work without withholding resources if not necessary. The operations of this Akka system are performed by four separate actor types, namely, vessel, cell, Database (DB) writer and proximity manager. The vessel actors are responsible for monitoring vessel movement and sending information to the cell actors that handle all traffic messages within an area. The cell actors calculate the proximity between vessels passing such information on to the according proximity manager actor. This actor manages all proximity messages and determines the according initiations and terminations of such events, forwarding their results to the DB writer. Finally, the DB writer is responsible for combining messages from other actors and updating the safe path database as well as returning all anomaly detection events. The system that was initially designed to capture simple such as *route deviation* from the expected trajectory and *proximity* movement between vessels has been further extended with machine learning scheme capable to capture Complex Events such as vessel activity patterns which are described in the following section. The ML algorithm is the Random Forest (RF) classification algorithm as it is an ensemble method that combines the predictions of multiple Decision Trees into one model, making it thus less prone to overfitting [40].

B. Run-Time Event Calculus

Run-time Event Calculus [41] is an efficient dialect of Event Calculus (EC), which in turn is a language for representing events and their effects over time. The main feature of EC are fluents, which are defined as variables that may hold a value at any point in time. The rules determining the value of a a fluent may include the occurrence of actions and the current values of other fluents. Furthermore, the main purpose of EC is representing the effects of these actions, appearing as input events, upon fluents. These fluents can be considered as complex events or patterns upon the input events, and can be defined by the user.

In RTEC fluents are of two types fluents: simple and statically determined. Simple fluents are defined by rules that determine the point in time where the value of the fluent changes, with each one triggered by the occurrence of at least one event. The second type of fluents, statically determined fluents (SDFs), determine the intervals where a fluent holds a value continuously. These rules use the values (in terms of intervals) of other fluents in order to determine their results through interval manipulation predicates. Predicates for union, intersection and relative complements between intervals are defined.

IV. Data

In this section we present the nature of our data, some information about the dataset we used and the activities used for evaluating the accuracy of the two systems.

A. AIS Messages

In most maritime monitoring applications the primary source of information regarding the vessels' movement is the AIS messages transmitted by the vessels. The initial purpose of AIS was to ensure safer traffic across the sea and in order to track mobility in their vicinity, all vessels carry an AIS transceiver that both transmits messages and enables for tracking nearby vessels. Moreover, simple receivers are available for port authorities and monitor stations, for the purpose of collecting AIS messages per area. Storing and analysing these data allows for understanding the vessel trajectories in a specific geographical area, as well as recognising vessel activities, such as fishing, tugging, ship-to-ship transfers, etc. There are several different types of messages that vessels can exchange (with an upper limit of 64 possible types). These message types are related to vessel's identification, its voyage information or to its position. In this paper we focus on types 1-3, 18 and 19 that are related to vessel tracking (i.e., position reports) and also message type 5 which includes vessel identification (e.g., vessel's name and its dimensions) and voyage information (e.g., reported destination).

B. Dataset

Collected by different stations around the globe, the data we used in our study include approximately 128k AIS messages. Within these messages, 1149 different vessels appear, with the vast majority ($\sim$90%) being either Cargo or Tanker vessels. Furthermore, the trajectories within our dataset represent the movement of vessels from separate areas around the globe, including but not limited to North and South America, Central and Northern Europe and Asia, as seen in Fig. 1. The time period these messages occurred are from the January 14 2019 at 8 PM until July 3 2019 at 6 AM.

For our experimental evaluation with the RTEC system we used a preprocessed version of the dataset; this process led to the extraction of 219k additional events that were added to the

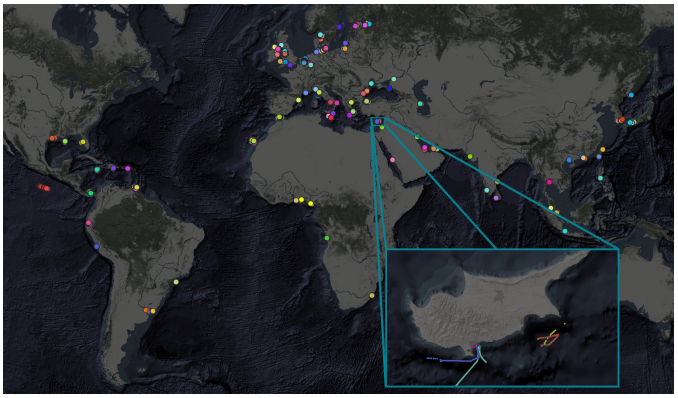

Fig. 1. Map of the trajectories used for our evaluation, with each colored point indicating a vessel trajectory. An expanded view of Cyprus is provided.

original AIS message stream (see Table I). More precisely, two stages of preprocessing resulted in our final stream version. First, a compression mechanism, that produces critical points within our stream. These critical points include information regarding the important or interesting changes in the vessels' movement. The second stage involves the creation of events regarding the spatial features of the AIS messages. Two types of events were produced: events that denote the position of a vessel within some predefined areas, referring to either ports or near the coastline, and the periods where two vessels appeared to be close to one another, with a threshold of 250 meters.

C. Complex Events

For the evaluation of the accuracy of the two systems we selected three complex events: *ship-to-ship transfer*, *tugging* and *piloting*. These events describe activity between at least two vessels in each instance, and originate from real-life vessel activities.

1) Ship-to-ship transfer.: Cargo and tanker vessels may carry tons of shipment each time. While the transferring of that shipment usually occurs with shore based terminals at ports, the loading can also be carried between vessels at sea. Vessels may be moving alongside or simply placed next to each other during the transfer. While this activity occurs between oil or gas tankers to facilitate the shipment to different port destinations more efficiently, this activity may occur between other types of vessels as well. Such activities may take from few minutes up to several hour at a time.

2) Tugging.: While large vessels -such as tankers- are capable of traveling long distances, their movement is limited when it comes to routes of a smaller width. In order to facilitate their movement in ports or harbors we enlist the aid of other vessels. Such vessels are equipped with remarkably powerful engines, able to haul others, much greater in size.

3) Piloting.: On the other hand, the movement of a vessel inside a port can be assisted by maritime professionals that are experts in navigation of the specific port. In this case a boat that carries the pilot leaves from the port, comes to proximity with the vessel and the pilot embarks the vessel, so as to guide its navigation in the port. Then the small boat returns to the port. The duration of the whole embarking operation may be just a few minutes at a time.

TABLE I
TYPES OF EVENTS PRODUCED FOR RTEC DURING THE PREPROCESSING STAGE.

Event Type	Indication
Critical Points	
gap (start / end)	Long period without receiving messages from a vessel.
slow_motion (start / end)	Vessel movement at a low speed.
stop (start / end)	Vessel halting its movement.
change_in_speed (start / end)	Notable vessel speed change.
change_in_heading	Notable vessel heading change.
Spatial Event Points	
entersArea / leavesArea	Entrance to or exit from a specified area (usually ports or areas near the coast).
proximity	Two vessels appear to be moving close to each other.

V. EXPERIMENTAL EVALUATION

In our comparison we focus on the detection accuracy of those activities that each system has achieved, in order to examine how maritime phenomena between vessels can be represented. For that purpose we compare the resulting matches of the complex events mentioned in the previous chapter (*piloting, tugging, ship-to-ship transfer*). We use a ground truth that includes a total of 4524 such complex events, covering up to aforementioned positions (tagged messages) and is provided by MarineTraffic. Moreover, we evaluate the results using the accuracy, recall and F_1-score metrics and we consider both the positions correctly tagged and the number of full complex events detected by the systems (Table II).

For the RTEC system all patterns rely on rules that have been designed by experts. The patterns require for the vessels to move close to another, as denoted by the proximity events, and are defined as follows:

- *Tugging* denotes movement of vessels with a speed of 1.2 to 15 nautical miles per hour, can be terminated by a communication gap of one of the vessels and lasts at least 5 minutes.
- *Ship-to-ship transfer* occurs when two vessels move closely to one another, with a low speed (below 5 miles), out of port areas, and away from the coast, and lasts more than 4 minutes.
- Finally, *piloting* happens when two close-by ships move away from the coast, with a low speed or are stopped but not docked at a port.

In order to examine the importance of the trajectory and vessel features used, multiple pattern variations were tested, besides the aforementioned version (*RTEC(1)*). More precisely, we also included information about the types of vessels involved (*RTEC(2)*), and removed all conditions concerning the distance of the vessels from the coastline (*RTEC(3)*).

The anomaly detection system uses a random forest classifier that models the kinematic behavior of the vessels and learns through a training dataset the boundaries that distinguish

each one of those activities. To train the model and ensure that it will not overfit to the training data a 5-fold cross validation has been applied and the average values for the precision, recall and f1 score have been extracted.

The results presented in Table II show that even though the proximity events produced by the preprocessing come close in encapsulating the complex events, the RTEC patterns miss about half of them. Moreover, we can conclude that conditions that include the distance of the vessels from the coastline or their types are responsible for omitting a significant amount of the desired events. On the other hand, the anomaly detection system performed satisfactorily by detecting approximately 83% of the positions where an event takes place, indicating thus, that the classification model used could interpret properly the vessels behaviours and distinguish the investigated complex events. An indicative example can be found in Fig. 2, where the trajectories of two vessels that take part in a *piloting* event are shown. In this case, while the anomaly detection system recognizes the event, RTEC is not able to do so because of the vessels' high speed during the event. In other words, although static conditions regarding the vessel's movement can be found to be very useful in many cases, they are not suitable for modeling complex events at sea in others, as their values can vary depending on several parameters (e.g. area type of movement, the distance to other vessels or the coast etc.).

TABLE II
COMPLEX EVENTS ACCURACY EVALUATION

CER System	Precision	Recall	F_1-score	Positions
Anomaly Detection	0.817	0.832	0.821	83.33 %
RTEC (1)	0.556	0.361	0.438	32.6 %
RTEC (2)	0.556	0.448	0.496	44.8 %
RTEC (3)	0.439	0.529	0.48	52.9 %
proximity events	0.934	0.922	0.928	92.2 %

VI. CONCLUSIONS AND FUTURE WORK

In this study we focused on CEP for maritime data and particularly on assessing the accuracy of a machine learning approach and an EC system when detecting complex maritime events.The empirical evaluation conducted was based on real data and the performance of the systems was compared when detecting three different types of activities between vessels. Our evaluation indicates that additional requirements regarding the vessels' velocity or distance from the coastline prevent the RTEC system from detecting half of the desired events. On the other hand the anomaly detection approach manages to recognize up to 83 % of these events. In future work we intend to examine the runtime performance aspects of these systems and extend out assessment with other CEP engines, such as Apache Flink CEP[2] and Siddhi[3].

[2]https://ci.apache.org/projects/flink/flink-docs-stable/dev/libs/cep.html
[3]https://siddhi.io/

Fig. 2. A *piloting* event between two vessels near the South Hook LNG terminal in Milford Haven, Wales. Each color indicates the trajectory of a single vessel. This event is not detected by RTEC because of the high velocities recorded.

ACKNOWLEDGMENT

This work was supported by the INFORE project, which has received funding from the European Union's Horizon 2020 research and innovation programme, under grant agreement No 825070.

REFERENCES

[1] UNCTAD. *Review of Maritime Transport 2019*. UN, 2019.
[2] Solas chapter v - regulation 19 - carriage requirements for shipborne navigational systems and equipment. http://solasv.mcga.gov.uk/regulations/regulation19.htm. [Accessed: 24-Sep-2018].
[3] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of The Vldb Endowment - PVLDB*, 1, 08 2008.
[4] D. Zissis, K. Chatzikokolakis, G. Spiliopoulos, and M. Vodas. A distributed spatial method for modeling maritime routes. *IEEE Access*, 8:47556–47568, 2020.
[5] Gerben Klaas Dirk Vries and Maarten Someren. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Systems with Applications*, 39, 12 2012.
[6] Konstantinos Chatzikokolakis, Dimitrios Zissis, Giannis Spiliopoulos, and Konstantinos Tserpes. Mining vessel trajectory data for patterns of search and rescue. In *EDBT/ICDT Workshops*, 2018.
[7] Konstantinos Chatzikokolakis, Dimitrios Zissis, Spiliopoulos, and Tserpes. A comparison of supervised learning schemes for the detection of search and rescue (SAR) vessel patterns. *GeoInformatica*, May 2019.
[8] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
[9] João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. On evaluating stream learning algorithms. *Machine Learning*, 90(3):317–346, 2013.
[10] Jianzhong Li, Beng Chin Ooi, and Weiping Wang. Anonymizing streaming data for privacy protection. In Gustavo Alonso, José A. Blakeley, and Arbee L. P. Chen, editors, *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, pages 1367–1369. IEEE Computer Society, 2008.
[11] Andreas Weiler, Marc H. Scholl, Franz Wanner, and Christian Rohrdantz. Event identification for local areas using social media streaming data. In Kristen LeFevre, Ashwin Machanavajjhala, and Adam Silberstein, editors, *Proceedings of the 3rd ACM SIGMOD Workshop on Databases and Social Networks, DBSocial 2013, New York, NY, USA, June, 23, 2013*, pages 1–6. ACM, 2013.

[12] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 181–189. The Association for Computational Linguistics, 2010.

[13] Jun Zou, Faramarz Fekri, and Steven W. McLaughlin. Mining streaming tweets for real-time event credibility prediction in twitter. In Jian Pei, Fabrizio Silvestri, and Jie Tang, editors, *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015, Paris, France, August 25 - 28, 2015*, pages 1586–1589. ACM, 2015.

[14] Tawseef Naqishbandi, C Imthyaz Sheriff, and S Qazi. Big data, cep and iot: redefining holistic healthcare information systems and analytics. *Int J Eng Res and Technol*, 4(1):1–6, 2015.

[15] Ioannis Kontopoulos, Giannis Spiliopoulos, Dimitrios Zissis, Konstantinos Chatzikokolakis, and Alexander Artikis. Countering real-time stream poisoning: An architecture for detecting vessel spoofing in streams of AIS data. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, DASC/PiCom/DataCom/CyberSciTech 2018, Athens, Greece, August 12-15, 2018*, pages 981–986. IEEE Computer Society, 2018.

[16] Konstantinos Chatzikokolakis, Dimitris Zissis, Marios Vodas, Giannis Spiliopoulos, and Ioannis Kontopoulos. A distributed lightning fast maritime anomaly detection service. In *OCEANS 2019-Marseille*, pages 1–8. IEEE, 2019.

[17] Angelos Valsamis, Konstantinos Tserpes, Dimitrios Zissis, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction. *Journal of Systems and Software*, 127:249 – 257, 2017.

[18] Kostas Patroumpas, Alexander Artikis, Nikos Katzouris, Marios Vodas, Yannis Theodoridis, and Nikos Pelekis. Event recognition for maritime surveillance. In Gustavo Alonso, Floris Geerts, Lucian Popa, Pablo Barceló, Jens Teubner, Martín Ugarte, Jan Van den Bussche, and Jan Paredaens, editors, *Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, March 23-27, 2015*, pages 629–640. OpenProceedings.org, 2015.

[19] Manolis Pitsikalis, Alexander Artikis, Richard Dreo, Cyril Ray, Elena Camossi, and Anne-Laure Jousselme. Composite event recognition for maritime monitoring. In *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems, DEBS 2019, Darmstadt, Germany, June 24-28, 2019*, pages 163–174. ACM, 2019.

[20] Eugene Wu, Yanlei Diao, and Shariq Rizvi. High-performance complex event processing over streams. In Surajit Chaudhuri, Vagelis Hristidis, and Neoklis Polyzotis, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, USA, June 27-29, 2006*, pages 407–418. ACM, 2006.

[21] Elena Ikonomovska and Mariano Zelke. Algorithmic techniques for processing data streams. In Phokion G. Kolaitis, Maurizio Lenzerini, and Nicole Schweikardt, editors, *Data Exchange, Integration, and Streams*, volume 5 of *Dagstuhl Follow-Ups*, pages 237–274. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013.

[22] Martin Hirzel, Guillaume Baudart, Angela Bonifati, Emanuele Della Valle, Sherif Sakr, and Akrivi Vlachou. Stream processing languages in the big data era. *SIGMOD Rec.*, 47(2):29–40, 2018.

[23] Lajos Jenő Fülöp, Gabriella Tóth, Róbert Rácz, János Pánczél, Tamás Gergely, Arpád Beszédes, and Lóránt Farkas. Survey on complex event processing and predictive analytics. In *Proceedings of the Fifth Balkan Conference in Informatics*, pages 26–31. Citeseer, 2010.

[24] Hai-Lam Bui. Survey and comparison of event query languages using practical examples. *Ludwig Maximilian University of Munich*, 2009.

[25] Alexander Artikis, Alessandro Margara, Martín Ugarte, Stijn Vansummeren, and Matthias Weidlich. Complex event recognition languages: Tutorial. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, DEBS 2017, Barcelona, Spain, June 19-23, 2017*, pages 7–10. ACM, 2017.

[26] Nikos Giatrakos, Elias Alevizos, Alexander Artikis, Antonios Deligiannakis, and Minos N. Garofalakis. Complex event recognition in the big data era: a survey. *VLDB J.*, 29(1):313–352, 2020.

[27] Hassan Sharghi and Kamran Sartipi. An expressive event-based language for representing user behavior patterns. *J. Intell. Inf. Syst.*, 49(3):435–459, December 2017.

[28] Jagrati Agrawal, Yanlei Diao, Daniel Gyllstrom, and Neil Immerman. Efficient pattern matching over event streams. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 147–160, 2008.

[29] V. Akila, V. Govindasamy, and S. Sandosh. Complex event processing over uncertain events: Techniques, challenges, and future directions. In *2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, pages 204–221, 2016.

[30] Elias Alevizos, Anastasios Skarlatidis, Alexander Artikis, and Georgios Paliouras. Probabilistic complex event recognition: A survey. *ACM Comput. Surv.*, 50(5):71:1–71:31, 2017.

[31] Haopeng Zhang, Yanlei Diao, and Neil Immerman. On complexity and optimization of expensive queries in complex event processing. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 217–228, 2014.

[32] Elias Alevizos and Alexander Artikis. Being logical or going with the flow? A comparison of complex event processing systems. In Aristidis Likas, Konstantinos Blekas, and Dimitris Kalles, editors, *Artificial Intelligence: Methods and Applications - 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings*, volume 8445 of *Lecture Notes in Computer Science*, pages 460–474. Springer, 2014.

[33] Ivan Zappia, Federica Paganelli, and David Parlanti. A lightweight and extensible complex event processing system for sense and respond applications. *Expert Systems with Applications*, 39(12):10408 – 10419, 2012.

[34] Fernando Terroso-Saenz, Mercedes Valdes-Vela, and Antonio F Skarmeta-Gomez. A complex event processing approach to detect abnormal behaviours in the marine environment. *Information Systems Frontiers*, 18(4):765–780, 2016.

[35] Juan Boubeta-Puig, Inmaculada Medina-Bulo, Guadalupe Ortiz, and Germán Fuentes-Landi. Complex event processing applied to early maritime threat detection. In *Proceedings of the 2nd International Workshop on Adaptive Services for the Future Internet and 6th International Workshop on Web APIs and Service Mashups*, WAS4FI-Mashups '12, page 1–4, New York, NY, USA, 2012. Association for Computing Machinery.

[36] Georgios M. Santipantakis, Akrivi Vlachou, Christos Doulkeridis, Alexander Artikis, Ioannis Kontopoulos, and George A. Vouros. A stream reasoning system for maritime monitoring. In Natasha Alechina, Kjetil Nørvåg, and Wojciech Penczek, editors, *25th International Symposium on Temporal Representation and Reasoning, TIME 2018, Warsaw, Poland, October 15-17, 2018*, volume 120 of *LIPIcs*, pages 20:1–20:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[37] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering. *Proc. VLDB Endow.*, 1(1):1081–1094, August 2008.

[38] A. Soares Júnior, C. Renso, and S. Matwin. Analytic: An active learning system for trajectory classification. *IEEE Computer Graphics and Applications*, 37(5):28–39, 2017.

[39] Zhijun Chen, Jie Xue, Chaozhong Wu, LingQiao Qin, Liqun Liu, and Xiaozhao Cheng. Classification of vessel motion pattern in inland waterways based on automatic identification system. *Ocean Engineering*, 161:69 – 76, 2018.

[40] Bart Lariviere and Dirk Van den Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29:472–484, 08 2005.

[41] Alexander Artikis and Georgios Paliouras. Behaviour recognition using the event calculus. In Lazaros S. Iliadis, Ilias Maglogiannis, Grigorios Tsoumakas, Ioannis P. Vlahavas, and Max Bramer, editors, *Artificial Intelligence Applications and Innovations III, Proceedings of the 5TH IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI'2009), April 23-25, 2009, Thessaloniki, Greece*, volume 296 of *IFIP Advances in Information and Communication Technology*, pages 469–478. Springer, 2009.