Big Data Analytics for Time Critical Mobility Forecasting: Recent Progress and Research Challenges

G. A. Vouros, A. Vlachou, G. Santipantakis, C. Doulkeridis, N. Pelekis, H. Georgiou, Y. Theodoridis, K. Patroumpas,
University of Piraeus
Piraeus, Greece

Elias Alevizos, Alexander Artikis
IIT, NCSR 'D', Greece

Georg Fuchs, Michael Mock, Gennady Andrienko, Natalia Andrienko
Fraunhofer Institute IAIS
Sankt Augustin, Germany

Christophe Claramunt, Cyril Ray
Ecole Naval & ENSAM
Lanvéoc, France

Elena Camossi, Anne-Laure Jousselme
CMRE, La Spezia, Italy

David Scarlatti
Boeing Research & Technoogy - Europe, Spain

Jose Manuel Cordero
CRIDA, Spain

ABSTRACT
The correlated exploitation of heterogeneous data sources offering very large archival and streaming data is important to increasing the accuracy of computations when analysing and predicting future states of moving entities. Aiming to significantly advance the capacities of systems to promote safety and effectiveness of critical operations for large numbers of moving entities in large geographical areas, this paper describes progress achieved towards time critical big data analytics solutions in user-defined challenges concerning moving entities in the air-traffic management and maritime domains. Specifically, the objective of this paper is to report progress and present further research challenges concerning data integration and management, predictive analytics for trajectory and events forecasting, and visual analytics.

Keywords
Big Spatio-temporal Data, Moving Entities, Trajectory prediction, Event forecasting

1. INTRODUCTION

The current Air Traffic Management (ATM) is nowadays changing its point of view from a time-based operations concept to a trajectory-based operations (TBO) one, which means a better exchange, maintenance and use of the aircraft trajectories for a collaborative decision-making environment, involving all the stakeholders in the process. In addition to that, real-time tracking and forecasting of trajectories, and early recognition of events related to vessels are essential for operating at sea. To address these challenges the knowledge of more accurate and more predictable trajectories is needed. Thus, the more accurate and rich information on trajectories and related events we have, and as we increase our abilities to predict trajectories and forecast events regarding moving entities' behaviour, we advance situational awareness, and consequently the decision-making processes.

Once the decision-making process has been improved, there are direct consequences in safety, efficiency and economy in the ATM and maritime domains. For instance, by having a better understanding of the air navigation data (historical data of flight plans, sector configurations and weather), the number of published regulations could be more accurately forecasted to improve the adherence to scheduled trajectories, with less delays and operational costs. Also, surveillance systems of moving entities at sea are important for the safety and efficiency of maritime operations. For instance, preventing ship accidents by monitoring vessels' activity represents substantial savings in financial cost for shipping companies (e.g., oil spill cleanup) and averts irrevocable damages to maritime ecosystems (e.g., fishery closure).

Due to the complexity of the ATM system, as well as due to factors contributing to increased uncertainty in the maritime domain, the current techniques for predicting trajectories are limited to a short-term horizon, while the event detection and forecasting abilities are limited. This is also due to the lack of methodologies to exploit the big amount of data from heterogeneous data sources with lack of veracity for (actual, historical and planned) trajectories and other contextual aspects (e.g. airspace sector configurations, regulations and policies, sea protected areas, weather patterns, for instance).

The objective of this paper is to describe progress achieved towards big data analytics solutions in user-defined challenges concerning moving entities in the air-traffic management and maritime domains, and presents research challenges concerning data integration and management, predictive analytics for trajectory and events forecasting, and visual analytics.

Most of the ideas and proposals developed in this paper are generated by the datAcron European funded project (http://www.datacron-project.eu/) whose aim is to advance the management and integrated exploitation of voluminous and maritime data sources, so as to significantly advance the capacities of systems to promote safety and effectiveness of critical operations for large numbers of moving entities in large geographical areas.

The rest of the paper is organised as follows. Section 2 introduces the challenges from both domains, motivating our research efforts. Section 3 presents the overall datAcron architecture, presenting the interactions of components described in subsequent sections: Section 4 presents the data management components and the datAcron ontology, while section 5 presents the location and trajectory predictors. Section 6 presents the events detection and forcasting components, and section 7 the online and offline visual analytics components. All sections present experimental results, providing evidence of the progress achieved towards time critical (i.e. real time) data processing and mobility analytics tasks. Finally, Section 8 draws the conclusions.

2. USER-DEFINED CHALLENGES IN THE ATM AND MARITIME DOMAINS

Efficiency in the air-traffic management system requires minimizing costs for both the airspace users (mainly airlines) and the operators (ANSP's: Air Navigation Service Providers). In general, one key enabler for reducing costs is the predictability of the system. In particular, from the point of view of the ANSP, maintaining the balance between the demand (number of users trying to use limited resources like airports, airspace sectors…) and the capacity (number of users which can safely use the mentioned resources) is one of the main challenges. For the airline, flying according to the plan, avoiding delays or extra fuel consumption represents the ideal to achieve daily operations.

The role of the trajectory in this efficiency enhancement endeavour is obvious: it defines which resources of the air-traffic management system will be used by each flight (airports, airways, sectors…), and it defines the achievable schedules, as well as the implied costs.

Big data technology presents opportunities to increase predictability capacities which are based mainly on complex theoretical models of the different components of the air-traffic management system. Exploitation of very large historical and streaming data sources for positioning, contextual aspects and weather is now possible, thanks to recent developments in data management.

Surveillance is an ever-increasing data source since new technologies are deployed (like ADS-B) which allow to collect data more widely (space based ADSB-B promises global coverage) and more frequently. Weather data, identically, each time is offered with more resolution both geographical and temporal. Contextual data, like flight plans, waypoints, or airways is increasing, linked to the traffic growth, year after year. While each data set is big, correlating and jointly exploiting all of them together is what makes big data technology necessary.

The aircraft trajectory must be understood not only as the 4D collection of points but also, including events relevant for the traffic management and the airline operations. So, predicting the aircraft trajectory implies predicting these events too, and visa-versa. The amount of information involved in this trajectory prediction process requires advanced visual analytics aids in order to understand the patterns of the predicted trajectories and events, inspect the exact reasons for deviating from plans towards either making adjustements to the actual system, or tune trajectory and event detection and prediction methods for more accurate results.

Accurate predictions of trajectories will further advance adherence to flight plans (intended trajectories) reducing many factors of uncertainty, allowing stakeholders to do better planning of the operations, reducing risk of disruptions.

Considering maritime scenarios [11], we aim to address operational concerns regarding fishing activities, highlighting the need for continuous, timely (i.e. real time) tracking of fishing vessels and surrounding traffic, as well as the need for offline data analytics.

S*ecurity in fishing* addresses the need to detect and foresee collisions between ships, potentially optimizing rendez-vous between rescuing ships in proximity of a vessel in danger and emergency services. *Collision avoidance is a* typical situation to be addressed: To prevent collision of fishing vessels with other ships we need to predict which other vessels (such as cargos, tankers, ferries) will cross the areas where the fishing vessels are fishing, sending a warning to the vessels identified for possible collision, taking also appropriate action as specified by COLREGs[1]. To advance decision making in these cases the potential risk assessment should be as accurate as possible. Such a development could also be used on board to enhance situational awareness, specifically when it is anticipated that a vessel will be required to "give way" to a fishing vessel.

Additionally, we need to detect *vessels in distress*, and further detect vessels in their vicinity to optimise rescuing operations. Analytics for detecting fishing patterns that are robust to noise and lack of verarcity in data, and accurate trajectory prediction algorithms, are fundamental to support effectively those operational requirements.

In m*aritime sustainable development* scenarios, supporting the monitoring of fishing activities' impact, including the illegal ones, is of immense importance. In particular, towards the *protection of areas from fishing* we address the issue of Illegal Unreported Unregulate (IUU) fishing, which is a global threat to the preservation of maritime ecosystems and could potentially undermine the sustainable development in large areas of the world that depend on maritime resources. Beside the introduction of maritime protected areas where protected species live and where navigation is prohibited, fishing seasons are regulated and fishing activities are forbidden in certain periods of the year, depending on the area and on the type of catch. Towards these objectives we need to predict and detect vessels entering, exiting, sailing, spending time or fishing in monitored geographical zones.

Type	Source	Format	Volume	Velocity
Surveillance	Automatic Identification System	Flat files	19.680.743 messages (1.05 GB)	~ 76 messages per min (in average)
	Automatic Identification System	Flat files	81.722.110 messages (8.11 GB)	~ 1.830 messages per min (in average)
	Automatic Identification System	Stream of messages in JSON	~ 400 KB / min (in average)	~ 3.700 messages per min (in average)
	FlightAware	Stream of messages in JSON	13GB/day	1.2Mb/s
	IFS	CSV Files	12GB/day (Spanish Airspace)	1.1Mb/s
Weather	Sea state	Flat files	79.652.684 forecasts (3.02 GB)	1463 forecast files - 1 file / 3 hours
	Weather forecast	Flat files	71.516 observations (5 MB)	1 obs/hour, from 16 stations
Contextual	Geographical	ESRI shapefiles	22 different features (1.4 GB)	Static
	Port Registers	ESRI shapefiles	5754 different ports (70 MB)	Static
	Vessel Registers	Flat files	166.683 distinct ships	Static
	ECTL NM B2B	CSV Files	1.7 GB/day	Static
	ECTL NM B2B	Flat Files	30MB/cycle	Static
Other	ECTL	CSV Files	30MB/month	Static

Table 1: The datAcron surveillance, weather and contextual data sources: lightly-shadowed rows correspond to ATM data sources, while weather forecast sources is for both domains.

Given the above cases in both domains, real-time integration of disparate data sources enabling scalability for massive amounts of dynamic data is an existing challenge, which is very closely connected to the maritime domain, as well as to the ATM domain. Table 1 presents the main data sources exploited in datAcron. A series of specific challenges concerning processing and managing data from these sources are as follows:

- Scalable, automatic, real-time processing, semantic annotation

[1]www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx

and linking of data towards coherent views on integrated cross-streaming and archival data;
- incremental integration of data, allowing advanced management and query-answering of spatio-temporal data;
- efficient distributed management and querying of integrated spatio-temporal data.

Revolving around the notion of trajectories and further making advances towards trajectories and events' detection and prediction, both domains present the following challenges:
- real-time reconstruction of entities' trajectories, supported by real-time processing and analysis of streams of data;
- algorithms for the prediction of anticipated trajectories at different time scale;
- algorithms for complex event recognition and prediction in real-time.

Visual Analytics (VA) [32] creates opportunities for a synergy between human analyst and computer by providing appropriate visual interfaces to all facets of analytical reasoning, from data exploration, pattern discovery and outlier identification, to prediction validation. It therefore facilitates the inclusion of the human domain expert's tacit knowledge and his capabilities for reasoning and intuition into the decision process, which are of fundamental importance in surveillance activities. The most important VA research challenges for both domains are as follows:
- interactive pattern extraction considering archival (data-at-rest) and streaming (data-in-motion) data, supporting the validation of early alerts obtained by the analysis tools;
- building situation overview and situation monitoring, capable of providing the overall operational picture of mobility at desired scales and levels of detail, both in spatial and temporal dimensions.

3. datAcron ARCHITECTURE FOR TIME CRITICAL MOBILITY FORECASTING

Critical mobility operations require integrating data that stems from a wide variety of diverse data sources, both archival (data-at-rest) and streaming (data-in-motion), having all big data characteristics. During data acquisition, various tasks need to be performed, incuding data cleaning, compression, transformation to a common representation model, and data integration. Besides real-time operations that must be supported with minimum latency requirements, there exists a need for offline analysis of the integrated data in order to discover patterns and extract useful knowledge.

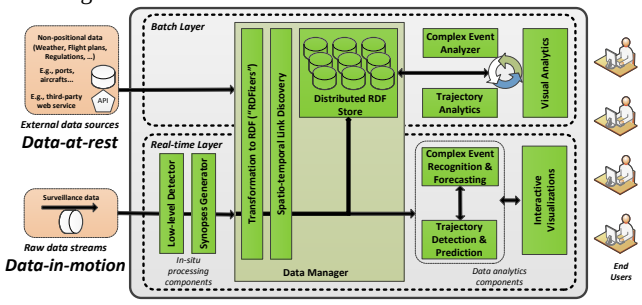


Figure 2: The datAcron system architecture

The datAcron system architecture, depicted in Figure 2, is a Big Data architecture for processing both real-time and archival data. While it bears similarities with the *Lambda architecture* [19], since it encompasses both a real-time and a batch processing layer, these layers in this architecture exist for different purposes (e.g. online trajectory/events forecasting vs offline trajectory clustering and visual analytics over archival data).

In the real-time layer, streaming surveillance data describing the positions of moving entities (e.g. vessels and aircrafts), collected from terrestrial and satellite receivers are fed into the system, while several operations are performed: Statistics (min/max/avg) are computed over properties, such as speed and acceleration, in an online fashion; online data cleaning of erroneous data, as well as trajectory reconstruction and compression are performed. Then, the generated trajectory synopses are transformed to RDF, according to the datAcron ontology, thereby facilitating the expression of links with relevant data originating from other sources. To this end, spatio-temporal link dicovery is performed that discovers relations between surveillance data and archival data (weather, contextual, etc.), resulting in semantically *enriched trajectories*. Further online analysis of enriched trajectories is performed, aiming at: (a) deriving predictions of the future location of a moving object, and (b) complex event recognition and forecasting. Finally, real-time visualizations support human interaction with the datAcron system.

In the batch layer, the enriched trajectories as well as data from other sources that have been transformed in RDF are collected for persistent storage, in order to support offline data analytics. Due to the immense data volume, parallel data processing is performed over RDF data stored in a distributed way. On top of the distributed RDF store, higher-level data analysis tasks run, in order to perform trajactory analysis (clustering, sequential pattern mining) and towards building models for complex event forecasting using machine learning techniques. Last, but not least, visual analytics provide the ability to discover hidden knowledge and patterns, by means of interaction with a domain expert or a data analyst, further improving situation awareness, as well.

Below, we describe the main components of the architecture.

In-situ processing components. In-situ processing allows computation as close to the sources as possible, thus reducing communication and latency. In datAcron, we apply in-situ processing on the streaming surveillance data, as it is ingested in the system. This supports computing statistical measures of moving entities' properties (such as speed and acceleration) and executing low-level event detection, annotating positions of moving entities with information regarding entry/exit to/from geographical areas of interest. In addition to that, trajectory compression aims to retain only a small set of positions of moving entities, also called *critical points*, without sacrificing the accuracy of the representation significantly.

Data manager. The data manager is responsible for providing a common representation of all sources by integrating and linking data in a knowledge graph, and for query processing over that graph. First, any incoming data (no matter whether streaming or archival) is lifted to RDF, by means of RDF genarators. The obtained representation is based on the datAcron ontology [27] (also in: http://www.datacron-project.eu/). Data interlinking is achieved via a spatio-temporal link discovery framework, which is designed to operate on streaming data sources, apart from archival. Finally, the integrated spatio-temporal RDF data is stored in a distributed way, supporting spatio-temporal RDF query answering by means of a parallel processing engine for RDF data, offering batch processing and analysis, with notable difference to existing solutions (see [1] for a recent survey).

Trajectory detection and prediction. This component predicts the future location of moving entities in real-time,

exploiting enriched trajectories offered by the data manager. The trajectory prediction component complements the future location predictor, while offline *trajectory analytics* (not in the scope of this article) over distributed RDF data are delivered by the corresponding component.

Complex event recognition and forecasting. This component targets the need to detect and forecast complex events related to the movement of moving entities. To detect and forecast events in a timely fachion, a novel technique using Patern Markov Chains is proposed for continuous narrative assimilation on data streams. In addition to that, machine learning methods are applied to build prediction models, while an offline *complex event analyser* operates on the historical data and discovers patterns of events to be predicted. The latter are not within the scope of this article.

Visual analytics. The aim is to support exploratory and interactive analysis of data, in order to enable the task of human interpretation, which is necessary in the case of Big Data. Visual analytics does not represent a single, specific analysis technique but rather a methodological approach to gain insight into large, complex, noisy and often conflicting data, to develop and test hypotheses, and to build and understand complex analytical models. The key aspect is the collaborative work between the computer and the human analyst, whereby the human expert imparts background knowledge about the current analysis task's context and reasoning in the overall analytical process.

For the implementation of the overal architecture, the big data technologies employed include a blend of state-of-the-art solutions that are used in production environments successfully. Stream procesing components have been developed in Apache Flink, harnessing the scalability and low latency offered. For batch processing and analysis, we have selected Apache Spark which is the most popular batch processing framework to-date, achieving scalability, high performance, and exploiting in-memory processing. The stream-based communication between components is achieved by means of Apache Kafka.

4. DATA PROCESSING, INTEGRATION AND MANAGEMENT

4.1 The datAcron data model and ontology

As it has been made apparent in the previous sections, ATM and maritime challenges concerning advanced predictive analytics methods, upon where we focus, mainly revolve around the notion of trajectory: Thus, our focus is to build solutions towards managing data that are connected via, and contribute to enriched views of trajectories. In datAcron we revisit the notion of semantic trajectory and built on it towards representing, storing and manipulating the wealth of information available in heterogeneous data sources in both domains, integrated in a representation where trajectories are the main entities. To support a coherent view on data towards addressing the user-defined challenges, we proposed a coherent and generic ontology for the representation of semantic trajectories, in association with related events and contextual information: The datAcron ontology (http://ai-group.ds.unipi.gr/datacron_ontology/) has been designed to provide a common model for all data sources in both domains towards supporting analysis tasks. Its development has been driven by ontologies related to our objectives (e.g. DUL, SimpleFeature, NASA Sweet and SSN) as well as schemas and specifications regarding data sources from the different domains.

To a greater extent than other models for representing trajectories, this ontology provides the means for specifying trajectories at varying levels of spatio temporal analysis: Trajectories can be seen as temporal sequences of moving entities' positions derived from raw data, as raw data aggregations signifying meaningful events providing a synoptic view of raw trajectories (generalizing on the stops and moves model [30], according to the types of critical points), as temporal sequences of meaningful trajectories segments (each revealing specific behaviour, event, goal, activity etc), or as mere geometries. Representations at any such level of analysis are linked to each other, as well as to related information and events.

Beyond answering spatio-temporal SPARQL queries concerning trajectories along with information regarding aspects that affect and are affected by the mobility of moving entities, this ontology supports generic data transformations for adapting available data to the analysis goals, or to specific requirements of the methods that the analyst wants to apply. This is done by extracting relevant parts of the data, reducing irrelevant details, converting movement data from one form to another, to support different task foci: movers, spatial, events, space, and time. Details are provided in [27].

According to the ontology specifications, as illustrated in Figure 3, a trajectory (`Trajectory`) can be segmented to trajectory parts (`TrajectoryParts`), each including other segments and/or more semantic nodes. Each semantic node may be associated with a specific raw position or a temporally ordered sequence of raw positions of a moving object. Trajectories and trajectory parts can be associated with any relevant information, as well as with events (`dul:Event`). Although events may happen independently from the trajectory but spatio-temporally co-occur with the trajectory, we focus on those happening on the trajectory itself (e.g. a "turn" or a "gap of communication") and on those concerning moving entity's state (e.g. vessel in a protected or in a bad-weather area). The detailed patterns for specifying structured trajectories and occurring events are presented in [27].

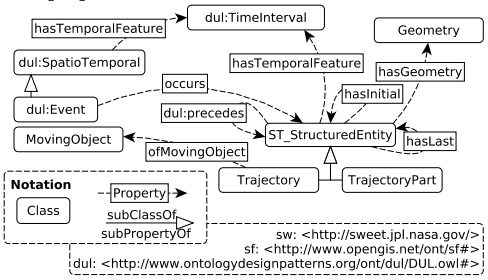


Figure 3. The main concepts and relations of the ontology.

4.2 Data processing and generation of synopses

4.2.1 Low level events detector

The low-level event detection component is aiming at enriching the raw-data generated by the moving entities with basic derived attributes that serve as input for higher-level procesing. A major consideration in this low level is to achieve an enrichment with low-latency, preferably by so-called "in-situ" processing. In-situ processing refers in general to the case of processing streaming data as "downwards" in-stream as possible. Processing streaming data close to data source provides a number of inherent advantages, such as decreased communication delays, savings in communication, and reduced overhead in sub-sequent evaluation steps.

The low-level events refer to two basic datAcron tasks to be performed in real-time on the trajectories: generating metadata on

incoming raw data for detection of erroneous data and ensuring data quality, and enriching the data stream with contextual information for further analysis. For supporting the data quality assessment, which is described in Section 7, attributes of min/max, median/average of properties (e.g. speed, acceleration etc.) are generated on a per trajectory basis.

In addition to that, raw position data are enriched with low-level events of entering or leaving of moving entities from one area to another one, by processing the real-time stream of moving entity positions.

4.2.2 Synopses Generation

Detecting important *mobility events* along trajectories has to be carried out in a timely fashion against the streaming positional updates received from a large number of moving entities, either vessels or aircrafts. Instead of retaining every incoming position for each object, we have implemented a *Synopses Generator* module that drops any predictable positions along trajectory segments of "normal"' motion characteristics, since most vessels and aircrafts usually follow almost straight, predictable routes at open sea and in the air, respectively. By doing so we may only retain positions that signify changes in actual motion patterns. We opt to avoid costly trajectory simplification algorithms like [16][17] operating in batch fashion, online techniques employing sliding windows [18] or safe area bounds for choosing samples [17], as well as more complex, error-bounded methods [10]. Instead, emanating from the novel trajectory summarization framework introduced in [24][26], specifically for online maritime surveillance, but significantly enhanced with additional noise filters and also extended for the needs of the aviation domain, the Synopses Generator applies single-pass heuristics for achieving succinct, lightweight representation of trajectories. We prescribe that each trajectory can be approximately reconstructed from judiciously chosen *critical points* of the following types:

- `Stop` indicates that an entity remains stationary (i.e., not moving) by checking whether its instantaneous speed is lower than a threshold over a period of time.
- `Slow motion` means that an entity consistently moves at low speed over a period of time.
- `Change in Heading`: Once there is an angle difference in heading greater than a given threshold with respect to the mean velocity vector (computed over the most recent course), the current location should be emitted as critical.
- `Speed change`: Such critical points are issued once the rate of change for speed exceeds a given threshold with respect to its mean speed over a recent time interval.
- `Communication gaps` occur when an entityhas not emitted a message over a time period, e.g., the past 10 minutes.
- `Change in Altitude` may be detected for aircrafts by checking their rate of climb (or descent), i.e., the vertical speed of the aircraft (in feet/sec) when ascending (respectively, descending). Once, this value exceeds a given threshold, a critical point should be issued in the synopsis.
- `Takeoff` is the latest location of an aircraft while still on the ground, as its next location reports an altitude above ground.
- `Landing` for flying aircrafts is the first reported location when they touch the ground.

This module can achieve dramatic compression over the raw streaming data with *tolerable error* in the resulting approximation. At lower or moderate input arrival rates, *data reduction* is quite large (around 80% with respect to the input data volume), but in case of very frequent position reports, compression ratio can even reach 99% without harming the quality of the derived trajectory synopses.

Empirical results [24] indicate that such critical points can be emitted in real-time keeping in pace with the incoming raw streaming data. As a next step, we plan to address the case of *cross-stream processing*, i.e., correlating surveillance data from multiple (and perhaps contradicting) sources in order to provide a coherent trajectory representation.

4.2.3. RDF generation and data integration

To convert the data from diferent sources into the common RDF model and integrate them in a knowledge graph, we designed and implemented a generic RDF generation framework, which can be instantiated and reused on any of the given (streaming or archival) data sources, to populate the datAcron ontology. Triples produced by the RDF generators are directed to a group of Link Discovery components, to further link entities in the generated knowledge graphs.

Due to the syntactic and semantic heterogeneity of data sources exploited in datAcron, and also given that sizes vary from a few thousands (e.g. aircraft or vessel registries), to practically infinite streams of data (e.g. reported positions of moving entities) we need an efficient method that can easily be integrated to widely used SPARQL workflows, to rule all the data sources, and that will also be easily adapted to changes on both the ontology and the sources, while the output will be easily verified.

The proposed method stands on two main components: a) the data connector, and b) the triple generator.

The data connector is responsible to connect to a data source and accept the data provided. It is capable of applying basic data cleaning operations, computing and converting values, applying simple filters, or generating values from the incoming entries, e.g. extracting the Well-Known-Text representation of a given geometry in a Shapefile. The output of these connectors is directed to instances of the triple generator component.

The triple generator is responsible to convert all the data coming through the data connector, into meaningful triples w.r.t. the datAcron ontology exploiting graph templates and variable vectors. The variables vectors while enabling transparent reference to datasource fields as variables, it enables the RDF generation method to refer to data not explicitly available in the source, but generated during the generation process. The graph template on the other hand, uses these variables into triple patterns; i.e. in triples where any of the subject or object can be either a variable or a function with variable arguments. Such an example of data conversion into triples by exploiting a graph template made of triple patterns, is provided in Figure 3.

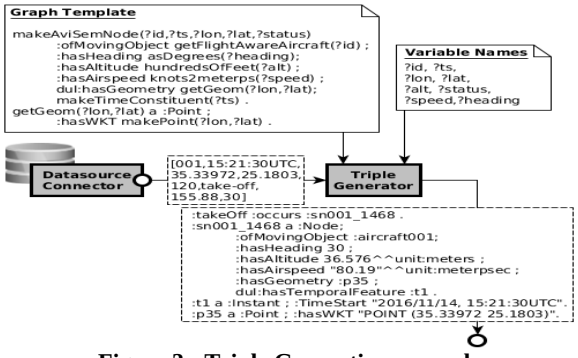


Figure 3 - Triple Generation example

By doing so, and in contrast to other RDF generators, the proposed method needs no further knowledge of a specific vocabulary (e.g. compared to RML [12]), and it can be used by anyone who can write simple SPARQL queries. Furthermore, it requires no underlying SPARQL engine, and it inherently

supports parallelisation and streaming data sources (e.g. compared to SPARQL-Generate [15] and GeoTriples [14]).

This RDF generation method manages to transform 10,500 input records to RDF per second. For some sources, this number may be smaller due to complicated geometries that need to be processed. Overall, the average time per triple generated is approximately 0.04 seconds, given that the frequency of position reporting per aircraft/vessel is at least 2 seconds.

4.2.4 Link Discovery

The output of the RDF generators is further exploited for the detection of associations between entities, or the encrichment of the generated RDF graph with additional information from any of the sources available.

The datAcron link discovery component mostly detects spatio-temporal and proximity relations such as "within" and "nearby" relations between stationary and/or moving entities. It is noteworthy that there is not much work on the challenging topic of spatio-temporal link discovery nor on link discovery over streaming datasets. State of the art approaches such as [23], [29], [28] focus on spatial relations in static archival datasets only. In particular RADON [28] employs optimizations that can be only applied if the datasets are a-priori accessible as a whole, which cannot be assumed for streaming datasets. Our work addresses explicitly proximity and spatio-temporal relations in both archival and streaming datasources.

The implemented component continuously applies SPARQL queries on each RDF graph fragment produced by an RDF generator, to filter only those triples relevant to the computation of a relation *r*. It applies a blocking method to organize entities (either being moving or stationary entities), and a refinement function to evaluate pairs of entities in any block.

Aiming to discover spatio-temporal relations among entities, methods use an equi-grid which organizes entities by space partitioning. The temporal dimension is not partitioned: given a temporal distance threshold, we can safely clean up data that are out of temporal scope, i.e. entities that will never satisfy the temporal constraints of the relations. To effectively prune candidate pairs of entities, the proposed method computes the complement of the union of those spatial areas that correspond to entities in a cell and intersect with the cell's area: This cell area is called the *mask* of cell. Examples of masks are depicted in Figure 4, where the green regions illustrate the mask of cells generated from 8,599 Natura2000 and fishing regions around Europe.

Thus, for each new entity we identify the enclosing cell, and then we evaluate that entity against the spatial mask of the cell. If it is found to be in the mask, we do not need to further evaluate any candidate pair with entities in that cell. In addition to masks, the link discovery component uses a book-keeping process for cleaning the grid, towards identifying proximity relations among entities when dealing with streamed data.

We have evaluated the performance of the Link Discovery method with and without cell masks on a dataset of 4,765,647 critical points, against a dataset of 8,599 regions generating 381,262 `dul:within` and 9,122 `geosparql:nearTo` relations. The method without masks achieves linking 23.09 entities per second, while activation of the mask boosts the throughput to 123.51 entities per second. Preliminary results concerning `geosparql:nearTo` relations among critical points, as well as critical points and 3,865 ports, have shown a throughput of 328.53 entities per second, producing 2,536,967 relations.

Challenges lying ahead for link discovery, include both, the identification of more complex spatio-temporal relations in real-time streaming data, and improving performance and scalability. The latter can be achieved by the refinement of blocking schemes for achieving better load-balancing for tasks' parallelization, as well as by the use of advanced optimization techniques that will further restrain the number of comparisons in the cells.

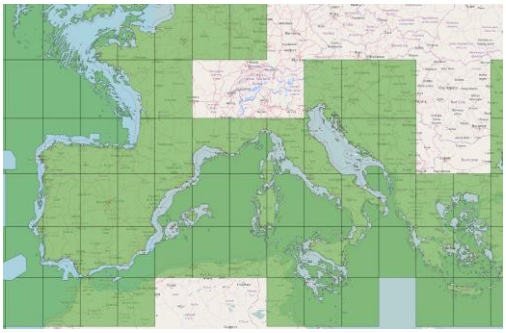


Figure 4 – Equi-grid with masks for stationary areas.

4.2.5 Knowledge Graph Store

Even though there exist several solutions for distributed RDF processing (see [1] for a survey), a notable difference is that we deal with mobility data that have a strong spatio-temporal flavour and typical queries also contain spatio-temporal constraints. A typical distributed RDF processing engine cannot process efficiently spatio-temporal constraints, as such constraints would have to be enforced in a post-processing step to obtain the final result, at the cost of having computed a much larger set of candidate results. Motivated by this limitation, we designed a solution for scalable processing of spatio-temporal RDF data. The system contains a distributed storage layer, and a batch processing layer developed in Apache Spark. In the storage layer, we employ a custom dictionary encoding technique for representing spatio-temporal entities. Our encoding technique allows representing an approximation of the position of any moving entity using a unique integer identifier, which corresponds to the spatio-temporal cell where the entity is located. We support different storage layouts, including "one-triples-table", vertical partitioning, and property tables. Also, for the file layout we exploit Parquet, which provides a columnar layout and achieves compression. The RDF triples are stored in HDFS, while the dictionary is stored in REDIS (a main-memory key-value store). In the processing layer, we have developed different implementations of basic operators (such as filtering and join) that can be used to generate different physical execution plans from a given logical plan. Moreover, the spatio-temporal encoding is used during query processing, by filtering triples that do not match with the spatio-temporal query constraints. This happens in parallel to filtering RDF triples, matching with the RDF graph patterns specified in the query.

Experimental results performed over more than 269M RDF triples from surveillance, weather, and contextual data sources show that we can improve query processing time for star join queries with spatio-temporal constraints by a factor of 5 when using our techniques.

5. TRAJECTORY PREDICTION

The prediction of a trajectory evolution can be seen either (a) as a Future Location Prediction (FLP), or (b) as a Trajectory Prediction (TP) problem. In FLP, the task is to exploit previous positions in order to predict the next *k* points in the trajectory, a process that is inherently dynamic and continuously adaptive, exploiting measured (reactive mode) or predicted (proactive mode) error as feedback. On the other hand, TP aims to exploit all the information regarding trajectories and produce a "best guess"

of the complete trajectory in the maximum likelihood sense. The two tasks are interconnected and applied in parallel, with FLP (TP) being the short-term online (full-length offline, respectively) predictor. Generally, there are two main approaches in addressing these prediction tasks:

(a) The *Kinetic approach;* which describes the forces and momentums that describe the motion of the moving entity in terms of physical laws. The main advantage of the kinetic approach is the accuracy of its predictions; however, the main drawbacks are the high-intensity processing required due to detailed simulation, as well as the quickly deviating predictions as the temporal window expands, being sensitive to changes in many of the (stochastic) parameters involved. In the aviation domain, the kinetic approach uses extremely accurate aircraft performance models, such as BADA (Base Of Aircraft Data), combined with localized weather forecasts. Similar kinetic approaches are used in various forms, e.g. for dead reckoning in navigation modules, in the maritime domain.

(b) The *Kinematic approach*; which considers only the temporal evolution of the model's parameters as time series and exploits the causalities discovered. In practice, this includes data-driven methods that exploit enriched trajectories as training sets for FLP and TP purposes. In other words, the model "learns" the kinetic behaviour of the moving entityby processing historical information of its own trajectory in case of FLP or of an entire group of "similar" trajectories in case of TP.

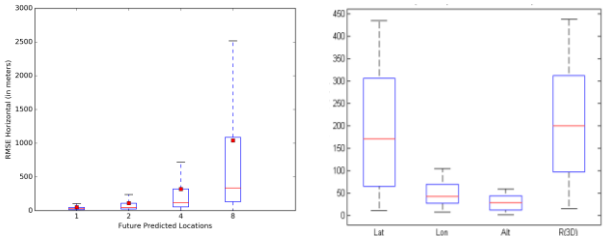
In contrast, the data-driven FLP and TP targeted in datAcron relies exclusively on reference points of actual trajectories, enriched with features (e.g. weather conditions, operational constraints, etc.) that affect trajectories.

The current state-of-the-art in data-driven approaches for FLP and TP range from standard signal processing to advanced regression learners. In FLP, standard regression methods, as well as motion-type modelling have been applied primarily in the short-term time frame [31]. Since TP includes complete trajectories, a number of supervised and unsupervised methods have been applied in the context of classification: Grouping together "similar" trajectories and predicting new ones based on these groupings for "similar" input conditions, e.g. for the same departure/destination, same weather conditions, etc. [33]. The current state-of-the-art approaches do not address the range of options from short to long term predictions in its entirety, nor exploit the full enrichment of the data points as constraints to optimize the training. Additionally, the volume and velocity of the data are considered of less or even no importance compared to the spatio-temporal prediction accuracy.

Recursive Motion Functions (RMF) for FLP: Mobility patterns over short-term time frames are often studied in the sense of online predictive analytics, i.e., involving small set of positions as "recent history" and strict constraints with regard to storage and processing resources. Tao et al. [31] propose an STP-tree (Spatio-Temporal Prediction tree), an indexing scheme that supports predictive queries and incorporates a general framework that computes different *non-linear motion patterns* to capture movements of arbitrary characteristics. In this context, the Recursive Motion Function (RMF) approach enables the computation of different types of movement (such as linear, polynomial, circular, etc) by exploiting the recent past of an object's position sequence and adapting the prediction model according to its specific characteristics.

According to our knowledge RMF is the most prominent candidate for addressing the online FLP task and under big data specifications. RMF captures the motion dynamics of an entityin a differential recursive formula by combining the most recent data points per f (system parameter) and is most effective when the acceleration components are zero, constant or at least exhibiting slow drifts: It results to very low prediction accuracy when it is applied in any of our domains.

The proposed RMF* method includes significant modifications and enhancements of the base RMF algorithm producing in real time the next *k* forward positions, using minimal storage and processing resources. It exploits dynamic motion pattern matching interchangeably with linear-only modes of operation. RMF* incorporates the advantages of linear extrapolation for the steady parts of the flights, while at the same time exploits additional information regarding any shift in the motion type provided by critical points, before activating the full pattern-matching mode. This means that the algorithm continuously checks for drifts to non-linear phases, i.e., the beginning of turn and/or altitude change, activating the proper differential approximator accordingly, including sections of circular, ellipsoid, parabolic, hyperbolic or general quadratic trajectory.



(a) (b)

Figure 5 (a) RMF* prediction accuracy over various look-ahead time frames. (b) Accuracy estimations for the per-waypoint deviation (m) from flight plan (clusterID=1, size=255)

RMF* can achieve very accurate predictions for the FLP task, as the data effectively capture the dynamics that the given trajectory is based upon. Since FLP is very difficult mostly for the take offs and landings, the experimental evaluation of the proposed RMF* is primarily focused on the aviation domain and specifically on these non-linear phases. Results provided in Figure 5(a) are based on complete flights between two airports (Barcelona-Madrid) and present average 2-D spatial error (longitude, latitude) of roughly 1-1.2 km for a look-ahead time frame of up to a minute, with a sampling rate of 8 secs, and 8 look ahead steps (mean≈1000m, stdev≈500m, skewed towards zero).

The proposed RMF* algorithm is under optimization and fine-tunning of the pattern-matching module, with special attention in identifying and modeling a set of motion patterns or "primitives", separately for the aviation and for the maritime domains, so that the module can promptly and correctly identify the best choice when in non-linear mode.

Hybrid Clustering/HMM method for TP purposes: In order to address the TP task, there is a trend of using stochastic models for discovering and retrieving patterns from past history. Additionally, there is a need to address the task in the scale of big data. To address this challenge towards long term trajectory predictions our proposal is to partition the incoming data into subsets, train separate predictive models for each one of them and then use these models for individual predictions, provided that the ability to select the correct model exists.

Clustering is the most popular approach for unsupervised learning, ranging from simple *k* nearest neighbor (*k*-NN) grouping, to multi-level hierarchical restructuring of the input data and using an arbitrary well-defined distance function as similarity

metric. In any case, the advantage of having "cohesive" clusters of trajectories is that the processing of each individual subset is of smaller scale. Further to that, a clustering approach can employ a distance function that incorporates any additional properties and data linked to enriched trajectories. The SemT-OPTICS algorithm [25] provides the means for creating robust and "dense" clusters of trajectories, while the similarity between two enriched points is decomposed at two parts: The one regarding their spatio-temporal similarity and another for the enriching information part, adopting an appropriate variant of Edit distance with Real Penalty (ERP) [10].

The Hidden Markov Model (HMM) approach is widely used in modeling and predicting time series, including spatio-temporal mobility patterns. The HMM approach models the evolution of an entity's motion by a set of *states* and *transitions* between them, each one accompanied by a probability that is typically extracted by analyzing historic data. Additionally, the deviations between "intented trajectories" (e.g. flight plans in the ATM domain) and actual routes are modeled as HMM observations or *emissions*, in order to construct a probabilistic model for trajectories. We designed HMMs in a way that exploits reference points in conjunction to the enriching information. This is in contrast to "blind" approaches exploiting raw trajectory data [8][9].

Based on these, we devised a novel Hybrid Clustering/HMM approach [13] to address the TP task, following the two-stage rationale described above: Clustering at the first stage of processing, using a distance function that exploits enriched reference points, and training HMMs for each cluster, using only the reference points of the *medoid* of each cluster.

This Hybrid Clustering/HMM method exhibits at least an order of magnitude better accuracy in terms of absolute cross-track error compared to the current state-of-the-art "blind" HMM for TP, while at the same time it exhibits two to three orders of magnitude less processing and storage resources, due to the combined scaling-down of data, to clustering and to the use of reference points. More specifically, in the aviation domain the valid assumption is that every pilot tries to follow the submitted flight plan as closely as possible, due to regulations, safety rules and cost effectiveness. Of course, external factors such as significant changes in local weather conditions result in deviations from this ideal path. The purpose of this modelling approach is to be able to predict these deviations optimally, based on all the information available, including local weather (per waypoint), aircraft size, seasonal factors (time, weekday), etc. The current experiments on real aviation data (Spain, April 2016) show that deviations from flight plans can be predicted with a combined 3-D spatial accuracy of 183–736m (RMSE), averaged over the entire sequence of reference points for all clusters and statistically significant at a=0.05. Results are shown in Figure 5(b).

The proposed Hybrid Clustering/HMM approach is still under optimization. For the clustering stage, the challenge is to customize the similarity metrics properly and separately for the aviation and the maritime domains, since the formulation of the TP task is significantly different w.r.t the enrichment of the critical points. For the HMM stage, the main challenge is to capture the statistics of the per-waypoint deviations for entire clusters of trajectories. Especially for the maritime domain, the reference points must be defined more dynamically (e.g. via detected critical points) since there are no equivalents to flight plans available. Hence, more specialized probabilistic distributions are tested for modelling the combination of distance-related Gaussian error distributions per-dimension. This typically involves exhaustive cross-validation experiments for prediction accuracy, rather than estimation of confidence intervals; e.g. via t-Student significance tests. Finally, segmented-trajectory models are also investigated, for very large training data sets.

6. EVENTS DETECTION & FORECASTING

Given a set of patterns that define the constraints that need to be satisfied in order to detect high-level events in the maritime and air traffic management domains, we aim to the real-time detection and forecasting of high-level events.

The problem is as follows: given a stream of low-level events (i.e. events detected using only the stream of surveillance data) and a set of patterns defining relations between low-level events, operational constraints and contextual information, we need to detect, in a timely manner, when these relations are satisfied. These relations may involve temporal and spatial aspects. Whenever these relations are satisfied, we say that a high-level (or complex) event has been detected. Besides event detection, which contributes to increasing situation awareness, given the importance of predictability in both domains, we additionally address the problem of forecasting the occurrence of complex events.

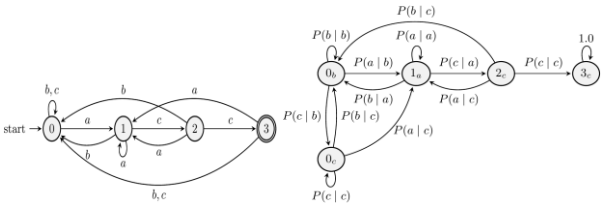
Considering the case where the low-level events are generated by the synopses generator described in Section 4.2.2. In this case, the input stream to the event detection module would consist of a sequence of events provided by the critical points detected, each one carrying extra information regarding the vessel identifier, the vessel's speed and heading etc. A maritime analyst might be interested in isolating parts of a vessel's trajectory during which a vessel changes its direction by 180 degrees (fishing vessels frequently execute such maneuvers). We could formally define this complex event as a temporal sequence of `Change In Heading` events where the first and last events in the sequence have opposite headings (their headings have a difference close to 180 degrees). This `HeadingReversal` pattern could be given to the event detection and forecasting module.

Whereas there are multiple event detection systems, at different levels of maturity, very few of them address the issue of forecasting ([22] is one of the few cases). Moreover, being able to predict complex events which are defined by patterns that are not simple sequences of input events, poses significant challenges. Our event detection and forecasting module advances the state-of-the-art by moving beyond sequential patterns. It has the ability to predict complex events that are defined in the form of regular expressions, where the low-level events may be related through *sequence*, *disjunction* or *iteration*. In addition, by employing a rigorous probabilistic framework, it can handle input streams that are generated by higher-order Markov processes (see [2] for a detailed description).

As a first step, event patterns in the form of regular expressions are converted to deterministic finite automata (DFA). A detection occurs every time the DFA reaches one of its final states. As an example, see Figure 6(a) which depicts the DFA constructed for the simple sequential expression R=`acc` (one event of type `a` followed by two events of type `c`) where the set of events that may be encountered are Σ={`a`,`b`,`c`}.

For the task of forecasting, we need to build a probabilistic model for (the behaviour of) the DFA. We achieve this by converting the DFA to a Markov chain. If we assume that the input events are independent and identically distributed (i.i.d.), then it can be shown that we can directly map the states of the DFA to states of a Markov chain and the transitions of the DFA to transitions of the Markov chain. The probability of each transition would then be equal to the occurrence probability of the event that triggers the corresponding transition of the DFA. However, if we

relax the assumption of i.i.d. events, then a more complex transformation is required, in which case the transition probabilities equal the conditional probabilities of the events. An an example, see Figure 6(b) which shows the Markov chain derived from the DFA of Figure 6(a) if we assume that the input events are generated by a 1st-order Markov process (see [3] for details). We call such a derived Markov chain a *Pattern Markov Chain* (PMC).



(a) (b)

Figure 6 (a) DFA and (b) corresponding Markov Chain

Once we have obtained the PMC corresponding to an initial pattern, we can compute certain distributions that are useful for forecasting. At each timepoint the DFA and the PMC will be in a certain state and the question we need to answer is the following: how probable is it that the DFA will reach its final state (and therefore a complex event will be detected) in *k* timepoints from now? The answer to this question depends on the state of the PMC. Hence, for each such state we need to calculate a separate distribution. These distributions are called *waiting-time distributions*. As an example, see Figures 7(a) and 7(b) which show a DFA and the waiting-time distributions for its states, respectively.

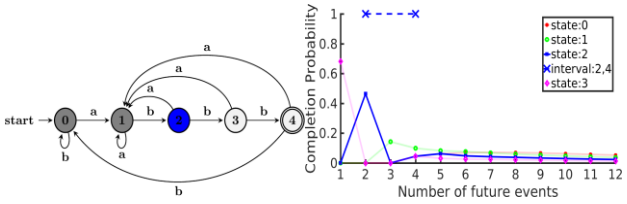
In order to estimate the final forecasts, another last step is required. Forecasts are provided in the form of time intervals, like *I=(start,end)*. When such a forecast is produced, its meaning is that the DFA is expected to reach a final state sometime in the future between *start* and *end* with probability at least some constant threshold θ (provided by the user). These intervals are produced by a sinlge-pass algorithm that scans a waiting-time distribution and finds the smallest (in terms of length) interval that exceeds this threshold. As an example, Figure 7 show a DFA being in state 2, the (highlighted) waiting-time distribution for this state in blue and the forecast interval extracted above the distributions (*I=(2,4)*).

The above described method has been implemented in the Scala programming language in a system called Wayeb. Wayeb was tested with real-world maritime trajectory data annotated and enriched. We show results from one pattern applied to a single vessel. The pattern is

```
R=ChangeInHeadingNorth
(ChangeInHeadingNorth+ChangeInHeading East)*
ChangeInHeadingSouth
```

where + stands for disjunction and * for iteration and each *turn* event has additionally been annotated with the vessel's heading. This pattern attempts to detect a `NorthToSouthReversal` event where a vessel executes a series of turns, initially heading towards the northern direction and eventually ends heading towards the southern direction. Figure 8 shows the precision of the proposed forecasting method for this pattern using different prediction thesholds. The precision is defined as the percentage of forecasts which were accurate (i.e. the event was indeed detected within the forecast interval). It shows results both for the assumption of a 1st-order and for a 2nd-order Markov process. We can see how increasing the assumed order does indeed positively affect precision.

Promising such results as they may be, there still remain significant challenges ahead. The most fundamental concerns *relationality*, i.e., the ability to naturally (without a pre-processing step) handle events with attributes and relations between the attributes of different events in a pattern. For example, in the `NorthToSouthReversal` pattern, the information about the vessel's heading would simply be an attribute which could be checked with predicates like `IsHeading(North)`. Moreover, the method that we have proposed assumes *stationarity* which implies that the transition matrix of the PMC does not change. However, the statistical properties of a stream may indeed change over time in which case we would need an efficient method for updating online the probabilistic model.



(a) (b)

Figure 7 (a) DFA and (b) waiting-time distributions

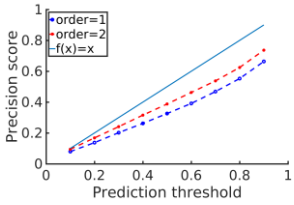


Figure 8 Precision achieved for events' forecasting using different Markov process' orders.

7. VISUAL ANALYTICS

The purpose of the Visual Analysis approach is to combine algorithmic analysis with the human analyst's insight and tacit knowledge in the face of incomplete or informal problem specifications and noisy, incomplete, or conflicting data [32]. Visual Analysis therefore is an iterative process where intermediate results are visually evaluated to ascertain and inform subsequent analysis steps based on prior knowledge and gathered insights. From the perspective of Visual Analytics, analysis methods fall into two categories within the overall architecture shown in Figure 2.

On the batch layer, Visual Analytics augments a wide range of tasks from initial data exploration and curation, complex analysis workflows, to refining and evaluating the different models. Synoptic analysis tasks that are the subject of such exploratory visual analyses presume availability of global measures like spatial extents, value ranges, (as yet undiscovered) patterns defined over large time spans/time cycles, and thus must be supported over sufficiently large data sets. Specifically, it is worth noting that due to the exploratory focus, VA does not prescribe a rigid pipeline of algorithmic processing steps, nor does it prescribe a fixed composition of specific visualizations, as opposed to typical dashboards [4].

To cope with these requirements in an efficient and scalable way, the VA component within the integrated architecture is itself of a modular, extensible design, as shown in Figure 9. It comprises four principal component groups – data storage, analysis methods, data filtering and selection tools, and of course, visualization techniques. Different components are typically composed in an ad-hoc fashion, through visual-interactive

controls, to facilitate the workflow required by the human analyst's task at hand. In particular, this allows creating linked multiple views to simultaneously visualize complementary aspects of complex data or analytical models.

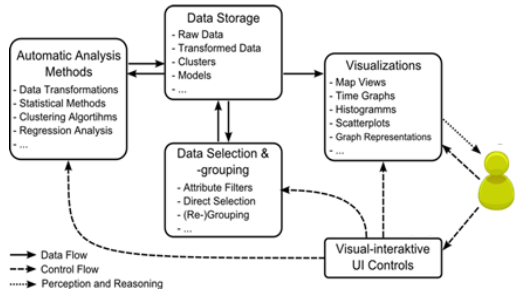


Figure 9. Principal components of the VA toolset.

The following paragraphs review several novel workflow compositions addressing different analytical challenges in both the ATM and maritime domains.

The ability to understand data properties and to assess their quality is a crucial first step in any data analysis setting. Dealing with massive movement data analyzed in context (e.g., as weather data) amplifies both the importance of that first step as well as the technical challenge involved in dealing with such large data.

Investigation of quality of movement data, due to their spatio-temporal nature, requires consideration from multiple perspectives at different scales. In paper [5], we review the key properties of movement data and, on their basis, create a typology of possible data quality problems and suggest approaches to identifying these types of problems. In particular, we systematically consider different approaches to position recording and related properties of movement data, taking into account properties of the mover set, spatial properties, temporal properties and data collection properties. Based on this, we define a typology of movement data quality problems and discuss visually supported means to detect them [5].

However, while [5] lays the foundation for a structured approach to detect and rectify data quality issues, cleaning and repairing data for curation purposes are still largely manual tasks that rely on a combination of tools and technologies such as database SQL, scripts, and functionality available in the VA toolkit. Especially when handling large data sets (many moving entities, long time periods) these tasks can become tedious and time-consuming. Therefore, as one facet of datAcron's objective to create advanced and scalable spatio-temporal data integration and management solutions, a modular framework is being developed that combines Big Data processing technologies with interactive visual reporting to automatically evaluate the quality of large movement data sets.

To support preparatory data analysis for building appropriate detection and prediction models, specifically patterns targeting at trajectories, events, spatial time series and spatial situations, novel methods are required that combine interactive visualizations with appropriate computational methods such as clustering, event detection, summarization and abstraction, as well as providing possibilities for manipulating parameters of computational methods and evaluating sensitivity to parameters.

In [7] we introduced the concept of time mask, which is a type of temporal filter suitable for selection of multiple disjoint time intervals in which some query conditions on arbitrary attributes hold. Such a filter can be applied to time-referenced objects, such as events and trajectories, for selecting those objects or segments of trajectories that fit in one of the selected time intervals. The selected subsets of objects or segments are dynamically summarized in various ways, and the summaries are represented visually on maps and/or other displays to enable exploration. The time mask filtering can be especially helpful in analysis of disparate data (e.g., event records, positions of moving entities, and time series of measurements), which in the considered scenarios even come from different sources.

To detect relationships between such data, the analyst may set query conditions based on one dataset and investigate the subsets of objects and values in the other datasets that co-occurred in time with these conditions (e.g., see Figure 10).

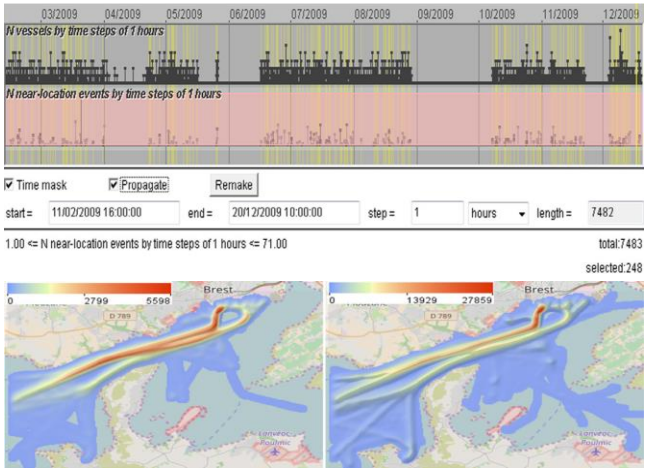


Figure 10. Top: A time series display shows the counts of the vessels (upper row) and the near-location events (lower row) by 1-hour time steps. A query selects the intervals containing at least one event (yellow markers). Bottom: The density of the trajectories in the times of occurrence of near-location events (left) and in the remaining times (right). Refer to [7] for details.

Clustering of trajectories of moving entities by similarity is an important technique in movement analysis. Existing distance functions assess the similarity between trajectories based on properties of the trajectory points or segments [2]. The properties may include the spatial positions, times, and thematic attributes. There may be a need to focus the analysis on certain parts of trajectories, i.e., points and segments that have particular properties. According to the analysis focus, the analyst may need to cluster trajectories by similarity of their relevant parts only. For example, when analysing routing decisions taken by airlines in the ATM context, only the cruise phase of a flight is relevant for comparison, but not holding patterns nor takeoff and landing runway directions [6]. Throughout the analysis process, the focus may change, and different parts of trajectories may become relevant, e.g., due to weather conditions.

In paper [6], we propose an analytical workflow that uses interactive filtering tools to attach relevance flags to elements of trajectories; subsequent clustering uses a distance function that ignores irrelevant elements. The resulting clusters are summarized for further analysis. The paper demonstrates how this workflow can be useful for different analysis tasks in three case studies related to ATM flow management (Figure 9). The paper [6] further proposes a suite of generic techniques and visualization guidelines to support movement data analysis by means of relevance-aware trajectory clustering.

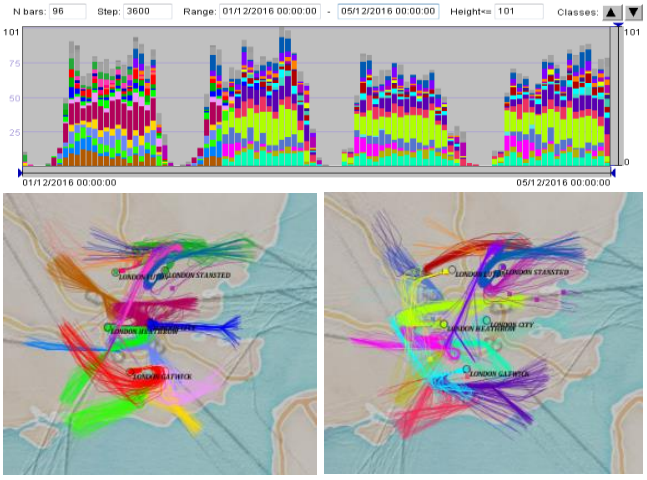


Figure 11. Top: Bars in a time histogram show the counts of the flight arrivals in hourly intervals. Bar segments are painted in the colors of the route-based clusters the flights belong to. A difference between day 1 and days 2-4 is notable. Middle: The final parts of the flight trajectories in days 1 and 3 are colored according to the cluster membership. Refer to [6] for details.

For developing and evaluating trajectory prediction algorithms it is important to have the possibility of detailed comparison of predicted trajectories to actual ones, to see how accurate the prediction is. It is also necessary to compare predictions obtained with different parameter settings, to understand the impact of the parameters and to choose the most suitable settings. A novel technique is the point matching method that is supplemented by interactive visual interfaces enabling the analyst to view and explore the results of point matching (Figure 12).

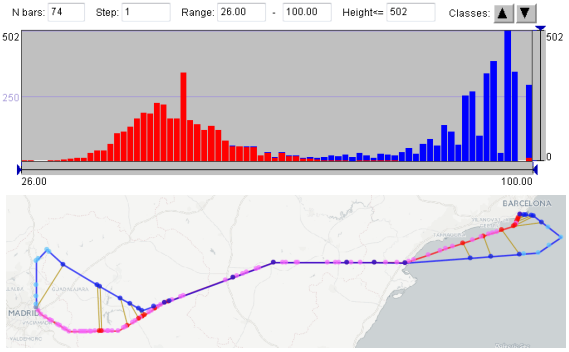


Figure 12. Detail view of a significantly mismatched pair of actual (blue) vs. predicted (red) trajectories. This outlier in a set of trajectory predictions was due to a short-term change of active runways for both takeoff and landing. The histogram shows the statistical distribution of the proportions of the matched points; the map shows the spatial footprints of both trajectories.

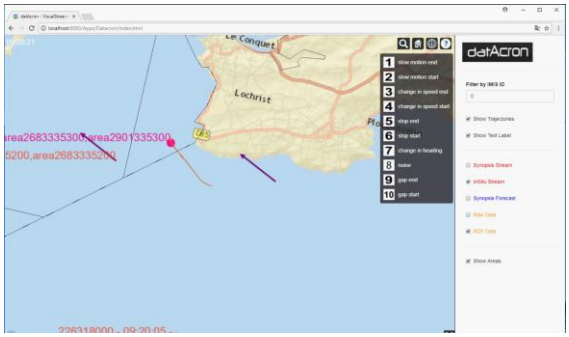


Figure 13. Real-time visualization dashboard for maritime situation monitoring.

On the real-time layer, in-stream processing algorithms operate directly on data streams under predefined parameter settings for monitoring purposes, i.e., trajectory & location prediction (Section 5) as well as event forecasting (Section 6). The main goal here is to provide a visual interface to the detection and prediction model output, presented in the context of real-time spatio-temproal data comprising the current situational picture (vessel trajectories, specific areas, wheather information etc.). These visualizations provide a limited set of interaction for confirmatory analysis of detected outliers and patterns, as well as in-context validation of model predictions, and typically are offered as dashboard components.

For the purposes of situation monitoring, a real-time visualization approach has been developed as an endpoint in the Kafka-based communication infrastructure (Figure 13). The visualization visually encodes a selectable subset of information layers from the enriched stream provided by the data manager. This stream, as described in previous sections, includes pre-processed position data (i.e., trajectory synopses), dynamic and static context information (e.g., weather conditions, maritime areas), trajectory and location predictions, as well as detected and forecasted events (e.g., initiation of a turn maneuver, danger of collision).

Further work will evaluate which visual encodings and interaction capabilities (e.g., to interactively adjust event detection parameters) best match different use case requirements in both domains.

8. CONCLUSIONS

Given the fact that more and more data of different nature and purposes is generated in both domains, and the user-defined challenges as defined in datAcron, this article reported on significant progress towards the real-time processing and analysis of big data for improving the predictability of trajectories and events regarding moving entities in the air traffic management and the maritime domains.

Albeit the progress, there are also significant challenges ahead: Discovery of a range of spatiotemporal relations among entities in both domains in a timely manner, efficient query answering of very large knowledge graphs for online and offline analytics tasks, cross-streaming synopses generation, long-term online full trajectory predictions and improvements in forecasting complex events together with learning/refining their patterns by exploiting examples; as well as the provision of online visual analytics workflows, are the major challenges ahead.

9. ACKNOWLEDGMENTS

This work is supported by project datAcron, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687591.

10. REFERENCES

[1] I. Abdelaziz, R. Harbi, Z. Khayyat, P. Kalnis: A Survey and Experimental Comparison of Distributed SPARQL Engines for Very Large RDF Data. *PVLDB* 10(13): 2049-2060 (2017).

[2] E. Alevizos, A. Artikis, and G. Paliouras. "Event Forecasting with Pattern Markov Chains." Proceedings of the 11th ACM Intl. Conf. on Distributed and Event-based Systems. ACM, 2017.

[3] E. Alevizos, A. Artikis, and G. Paliouras. "Event Forecasting with Pattern Markov Chains." Proceedings 11th ACM Intl. Conf. on Distributed and Event-based Systems. ACM, 2017.

[4] G. Andrienko, N. Andrienko, P. Bak, D.A. Keim, S. Wrobel. Visual Analytics of Movement. *Springer*, 2013.

[5] G. Andrienko, N. Andrienko and G. Fuchs, Understanding movement data quality. *Journal of Location Based Services*, 10(1), 31-46, 2016.

[6] G. Andrienko, N. Andrienko, G. Fuchs, J.M.C. Garcia. Clustering Trajectories by Relevant Parts for Air Traffic Analysis. IEEE Transactions on Visualization and Computer Graphics (proceedings IEEE VAST 2017), 2018, vol. 24(1), pp.??? (accepted)

[7] N. Andrienko, G. Andrienko, E. Camossi, Ch. Claramunt, J.M. Garcia, G. Fuchs, M. Hadzagic, A-L. Jousselme, C. Ray, D. Scarlatti, G. Vouros. Visual exploration of movement and event data with interactive time masks. *Visual Informatics*, 1(1):25-39, 2017

[8] S. Ayhan and H. Samet, "Aircraft Trajectory Prediction Made Easy with Predictive Analytics," in Proceedings of ACM SIGKDD 2016, 2016.

[9] S. Ayhan and H. Samet, "Time Series Clustering of Weather Observations in Predicting Climb Phase of Aircraft Trajectories," in IWCTS 2016, Burlingame (CA), USA, 2016.

[10] L. Chen and R. Ng, "On the marriage of edit distance and Lp norms," in Proceedings of VLDB 2004, 2004.

[11] datAcron Deliverable D5.1 Maritime Use Case and Scenarios, 2017 (http://www.datacron-project.eu/).

[12] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. de Walle, "RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data", Proc. *7th Work. Linked Data Web*, vol. 1184, 2014.

[13] X. Georgiou et al. "Semantic-aware Aircraft Trajectory Prediction" submitted in ICDE 2018.

[14] K. Kyzirakos, I. Vlachopoulos, D. Savva, S. Manegold, M. Koubarakis. T GeoTriples: a tool for publishing geospatial data as RDF graphs using R2RML mappings. Proceedings of the *ICSW 2014, Posters & Demonstrations Track* - Volume 1272, Riva del Garda, Italy, 393-396, 2014

[15] M. Lefrançois, A. Zimmermann, and N. Bakerally, "A SPARQL Extension for Generating RDF from Heterogeneous Formats", pages 35–50. *Springer International Publishing*, Cham, 2017.

[16] X. Lin, S. Ma, H. Zhang, T. Wo, and J. Huai. One-pass error bounded trajectory simplification. *PVLDB*, 10(7): 841-852, 2017.

[17] J. Liu, K. Zhao, P. Sommer, S. Shang, B. Kusy, and R. Jurdak. Bounded quadrant system: Error-bounded trajectory compression on the go. In *ICDE*, pp. 987-998, 2015.

[18] C. Long, R. Chi-Wing Wong, and H. V. Jagadish. Trajectory simplification: On minimizing the direction-based error. *PVLDB*, 8(1): 49-60, 2014.

[19] N. Marz and J. Warren, Big Data - Principles and best practices of scalable realtime data systems. *Manning Publications*. April 2015.

[20] N. Meratnia and R.A. de By. Spatiotemporal compression techniques for moving point objects. In *EDBT*, pp. 765-782. 2004.

[21] J. Muckell, P. W. Olsen Jr., J.-H. Hwang, C.T. Lawson, and S. S. Ravi. Compression of trajectory data: A comprehensive evaluation and new approach. *GeoInformatica*, 18(3): 435-460, 2014.

[22] V. Muthusamy, L. Haifeng, and H-A Jacobsen. "Predictive publish/subscribe matching." Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems. ACM, 2010.

[23] A. N. Ngomo. ORCHID - reduction-ratio-optimal computation of geo-spatial distances for link discovery. In Proc. of *ISWC 2013*, pages 395–410, 2013.

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

[25] N. Pelekis, S. Sideridis, P. Tampakis and Y. Theodoridis, Simulating our LifeSteps by Example", ACM Transactions on Spatial Algorithms and Systems, Vol. 2, Issue 3, October 2016.

[26] M. Potamias, K. Patroumpas, and T. Sellis. Sampling trajectory streams with spatiotemporal criteria. In *SSDBM*, pp. 275-284, 2006.

[27] G. Santipantakis, G. Vouros, C. Doulkeridis, A. Vlachou, G. Andrienko, N. Andrienko, G. Fuchs, J. M. C. Garcia, and M. G. Martinez. Specification of semantic trajectories supporting data transformations for analytics: The datAcron ontology. *Semantics 2017*.

[28] M. A. Sherif, K. Dreßler, P. Smeros, and A. N. Ngomo. Radon - rapid discovery of topological relations. In Proc. of *AAAI* 2017, pages 175–181, 2017.

[29] P. Smeros and M. Koubarakis. Discovering spatial and temporal links among RDF data. In Proc. of *LDOW 2016*.

[30] S. Spaccapietra, C. Parent, M.L. Damiani, J.A.Fernandes de Macedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Data Knowl. Eng.* 65, 1 (2008), 126–146.

[31] Y. Tao, Faloutsos, C., Papadias, D., & Liu, B. (2004). Prediction and indexing of moving objects with unknown motion patterns. In SIGMOD, pp. 611-622.

[32] J.J. Thomas, K.A. Cook. Illuminating the path: the research and development agenda for visual analytics. *IEEE Computer Society*, 2005.

[33] Yang, Yang, Jun Zhang, and Kai-quan Cai. "Terminal-area aircraft intent inference approach based on online trajectory clustering." The Scientific World Journal 2015 (2015).