# Graph-Based Analysis of Maritime Patterns of Life

Wissame LADDADA*, Cyril RAY*

*Arts et Metiers Institute of Technology, Ecole Navale, IRENav, F-29240 Brest, France
{wissame.laddada, cyril.ray}@ecole-navale.fr,
https://www.ecole-navale.fr

**Abstract.** Moving object data analysis and more specifically Pattern of Life (PoL) analysis aims to understand behaviors, movement trends of entities evolving over space and time and their connectivity across spatio-temporal and semantic dimensions. This paper presents a contribution based on a graph modeling for PoL that handles these dimensions. We store our model, loaded with maritime traffic data, into Neo4j graph database. Together with Cypher query language, the model supports a bunch ways to analyze the different segments constituting trajectories. We exploit historical data of trajectories in order to demonstrate some typical analyses that leads to enhance the understanding of maritime routes and traffic. This work is part of a set of mining services designed for European Commission funded project CISE (Common Information Sharing Environment) DMS (Data Mining Services).

## 1 Introduction

Several researches from the literature have focused on *Pattern of Life (PoL)* topic. This notion has been addressed by several scientific communities; History, demography, biology, computer science. . . to understand the dynamic and statistical behaviors of the entities constituting the environment in study. Thereby, authors overcomes with definitions related to their research area. The definition proposed in [6] stipulate that "PoL describes a recurrent (eg. normalcy) way of acting by an individual or group towards a given object or in a given situation". Another definition from [8] characterize a PoL as "a logical representation of a spatio-temporal state of affairs". This last definition refers to ontological though derived from philosophical domain. All the definitions converge to the fact that we may interpret a state of an entity across space and time by analyzing pattern of relationships between all organisms based on their similarities and differences –which is a theory drawn from a Tree of Life claimed by *Darwin*–.

In this context of PoL, computer scientists proposed to understand trajectories followed by moving objects in different environments [4] and maritime environment is no exception. The Common Information Sharing Environment (CISE) project at European level aims to deploy an architecture structured with different services that will be integrated into maritime surveillance systems and among these services PoL is an interesting one. Among CISE projects, CISE DMS (Data Mining Services) aims to experiment and demonstrate the benefits of *mining on historical data* coming from legacy systems in order to provide enriched information to CISE queries [11].

The main idea presented in this paper is to define the life cycle of ships and how their mobility influences activities related to a certain points of interest (POI). Such analysis helps to detect anomalies from pattern extraction and allows predicting moves by processing the data using static analyzes. It is also a potential background for machine learning model. For these goals, we model a maritime dataset as a graph into a graph database (Neo4j [1]). This graph modeling helps us to handle semantic information across space and time. Nodes and edges are defined to structure the dataset in a relevant way to get the needed information with simple queries. First results show it is a pertinent approach to avoid a use of complex queries (like SQL JOIN queries) when handling an important volume of interconnections between entities.

The paper is organized as follows: Section 2 provides a brief overview on the existing researches related to PoL analysis and graph-based model. Section 3 describes the dataset used in this work. We develop then the different steps to extract a graph and model it, respectively in sections 4 and 5. The latter define also the storage of the data into Neo4j database in order to present an example use case using Cypher [2] query in section 6. We present in the last section some perspectives and area for future work.

# 2 Related work

The increasing availability of navigation data sparks interest for different research topics related to the domain of maritime navigation traffic analysis. Processing such maritime data for requires modeling, management and analyzing of moving objects. Considering an existing batch of data, two main processing approaches exist in the literature. Some works consider direct processing of data files, using general-purpose data science environments (e.g. *Python*, *R*, *Matlab*) or developing customized tools for spatio-temporal data processing. Other works rely database management systems (DBMS), which nowadays offer a native support, in different database model (i.e. not only in a relational model), for modeling spatial information, optimized functionalities for spatial indexing and knowledge discovery, and efficient integration with a query language. Regardless of the approach used, we categorize main works as follows [11]:

- Vessels trajectory analysis and prediction: This mainly includes research on analyzing AIS and other complementary sources to predict trajectories of vessels. Various predictive analysis techniques where deployed for that matter [2, 5, 27].
- Anomaly detection: This work endeavor encompasses mostly clustering of trajectories to distinguish any irregularity in behavior. This is often used to detect threats or entry to restricted areas [21, 15, 14, 24].
- Analyzing human activity at the sea: Such as fishing activity, illegal traffic (drugs, refugees, goods), piracy, etc. [23, 18].
- Port traffic forecasting: This includes port volume handling forecast, cargo throughput forecast [13].
- Collision prevention: analyzing risk of ships collisions through analyzing ships navigational behavior [12, 3].

1. https://neo4j.com/
2. https://neo4j.com/developer/cypher-query-language/

— Multi-source information fusion: it involves fusion of multiple data sources (AIS, radar, satellite, sensors) for solving contradiction, redundancy, and uncertainty in available data. This can be helpful in achieving better accuracy in analysis (for example surveillance) [9, 16, 22, 1].

Many analysis techniques entail the application of different approaches to fulfilled the goals listed using maritime navigational data. Storing and managing such complex and big data becomes onerous when using traditional RDBMS approaches [17]. Moreover as prior works mainly concentrate on semi instant positioning data of ships they mostly stay limited to short term predictions and can be vulnerable to noise in data.

In the maritime domain, PoL addresses the understanding of navigation data at an aggregated (meso, macro) level [11]. A more abstract representation of ship routes will be more relevant. As a technically optimized alternative, graph database becomes more attractive to store, represent and query this kind of interlinked data.

The graph-based model has a typical way to represent networks and connections between entities. In this context, the field of *semantic trajectory* has been addressed and discussed through several studies. For instance, the work proposed in [7] defined a graph to describe trajectories considering different elements. Hence, the contextual information about data is managed by the representation of an important element into the graph which is the dimension hierarchies of an online analytical processing (OLAP). This graph data description provides a possibility to analyze trajectories at different aggregation levels with different queries using Cypher (query language for Neo4j). Furthermore, the authors show how a graph database may help to simply analyze and extract information from dataset comparing to classical representations using relational databases.

Following the same concept of graph-based model using Neo4j, the study proposed in [25], adopts a graph database to model and analyze simplified trajectories (using Semantically Enriched Line Simplification –SELF– structure) which are extracted from a relational database and loaded into Neo4j. The Figure 1 shows how the model is structured by joining each trajectory to its origin and destination point. In addition, the authors include some semantics in other nodes labeled “INTERMEDIATE” and enriched with some properties. Even if some unnecessary points are deleted in the simplification process of the trajectory, this graph may help to reconstruct it. Some classified queries using Cypher are presented in order to demonstrate how trajectories may be analyzed.

A triplestore is a graph oriented database that follows triplet structure. In this field, the approach in [8] associates a spatio-temporal ontology with different concepts that handle the PoL of a ship and formalize it. Thereby, a normalcy dimension is defined which allows alerting some anomalies as shown in the results. However, big data manipulation requires the use of a most relevant database in terms of complexity. In ontology database, increasing data volume involves complexity and time consuming for the reasoning process. In fact, enhancing the semantic expressiveness and analyzing data stored into ontology oriented database requires probably the use of some built-ins that we have to develop within some SWRL (Semantic Web Rule Language) rules. Therefore, we have to take into account some technical issues, since the query language SPARQL does not handle all inferences and all serializations, specially when the ontology is non trivial.

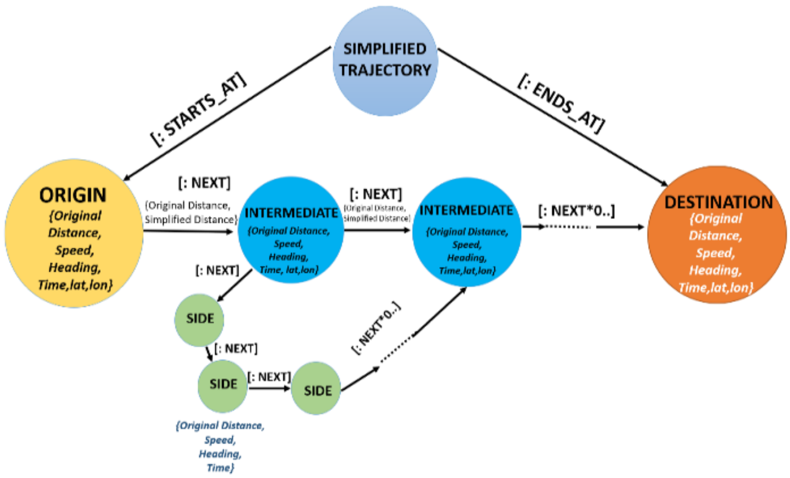


FIG. 1 – *Graph based model describing simplified trajectory in graph database using SELF [25]*

## 3 Dataset description

In the recent years, the literature addressing maritime analysis have been extensively exploiting Automatic Identification System (AIS) messages that ships broadcast via radio communications for collision avoidance. The system provides information on the ship kinematic (position, speed, destination) and additional nominative information on the ship, for instance its name, type or its international identifier (Maritime Mobile Service Identity: MMSI).

The design and the implementation of our pattern of life is based on these AIS data and is organized to be evaluated at different geographic scales. The first step presented in this paper considers the design and the development of data structure and analysis on a regional and well known dataset. The main reason for this arises from the necessity to train and validate the approach first. It is notable that the stationary and turning points of vessels and their navigational behavior in general vary significantly with the varied types of ships and areas of navigation understudy. Therefore, we postulate that evaluating the approach on a local well known maritime scale, at first, would allow to realistically and closely asses and justify the outcome of algorithms.The dataset used in this research relies and extends the representative heterogeneous maritime dataset produced under the umbrella of the *H2020 datAcron* project [20]. It includes historical traces of maritime vessels at regional scale with one major route crossing Ushant traffic separation scheme (cf. Figure 2).

AIS data are aggregated with contextual data spatially and temporally aligned [3]. The information contained in this dataset can be grouped into 4 main categories: navigation data (AIS data for vessel positions), vessel registers, geographic data (cartographic, topographic or regulatory context of vessel navigation), and environmental data (climatic and sea state related information). It covers a time period of six months from October $1^{st}$ 2015 to March $31^{st}$ 2016 and provides around 18.6 million observations of ship positions collected from 4,842

3. Link to the dataset: https://zenodo.org/record/1167595

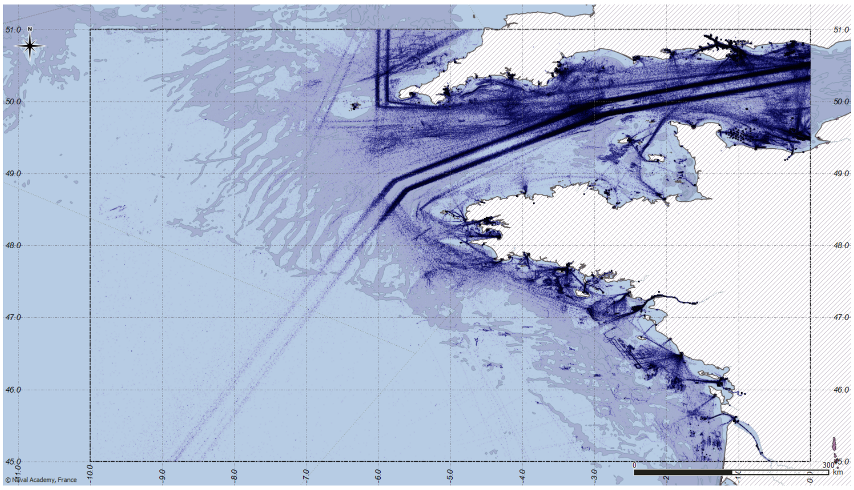


FIG. 2 – *A view of the location of the geo-localised points in AIS data-set*

ships over the Celtic sea, the North Atlantic ocean, the English channel, and the Bay of Biscay (France). The public version of the dataset has been extended additional geographic features extracted from nautical charts and with maritime events (search and rescue operations, ship assistance operation, etc.) reported by control centers along the same period.

# 4 Graph extraction

PoL graphs offer abstract semantic representations of ships' navigation routes that can ease an automated monitoring and understanding of maritime networks. The main components of such a network are nodes and edges. The nodes are prominent way points that form the route and that are repeatedly taken by ships (ex: stationary areas, ports, turning points, offshore stations, etc.). Whereas, the directional lines between nodes represents the transition from one stationary location to another while holding important semantics on this specific path. Since the simplification of trajectory data generates different points of interest interconnected by the trajectory segments, it is intuitive to think about trajectories as a network or a graph.

The approach considered for the graph building is similar to the work of [10], defining points of interest and inferring maritime routes from them. This leads to represent ships' navigation routes as node graphs instead of time series aligned with minutely geographic position coordinates. Figure 3 illustrates the 7 different steps considered; From raw position to routes, enriched network and finally life cycles.

The PostgreSQL/PostGIS database model and the associated pre-processing that extract a nodes-edges structure from AIS data has been detailed in a previous work [11]. This extraction of graph entities from raw AIS data (Figure 4: *dynamic_ships* and *static_ships*) is composed by two main phases processed in the geographic database:

— Ports, stationary, activity areas and turning points extraction: First we compute a set of areas of interest regarding ships navigation and activities at sea. The extraction of

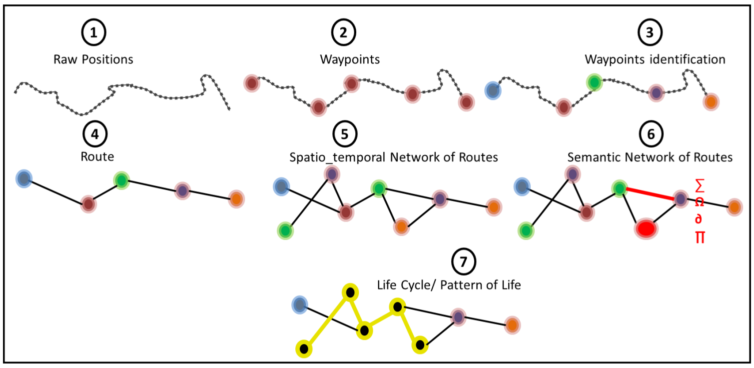


FIG. 3 – *Steps to constructing a semantic PoL graph of maritime traffic (colors of nodes illustrate that waypoints may be of different types; e.g. port, stationary area. Yellow path illustrate a life cycle as a subset of the overall network).*

stationary/activity areas and turning points is computed by the mean of Density-Based Clustering of Applications with Noise (DBSCAN).

— Routes extraction: From this list of areas we derive all the possible maritime split in separate maritime ways. The aim of the route extraction process is first to establish an origin/destination matrix describing connection between areas. The network basically represents the spatio-temporal information accompanied with usual routes.

Considering areas and routes detected, the objective is understanding and monitoring the behavioral patterns (i.e. semantic trajectories) of a certain ship for potentially predicting its future location and events at an aggregated (meso, macro) level., in regards with itself and other vessels. This allows to apply analysis for anomaly detection and route prediction but at a higher (semantic) level of abstraction. A recent study had been published addressing a similar approach to vessel trajectory mining proposed with a difference in the structure of the constructed model and the main application objectives [26]. In the following we explain how this nodes-edges structure pre-processed in a relational database is modeled and transferred into a Graph Database Management System (Graph DBMS).

# 5 Graph database modeling

From this previous nodes-edges structure, we specially consider two entities (cf. Figure 4): *edges_list* and *port_locations_(nodes)*. Those entities are generated after the aforementioned different steps from original data rows allowing us to structure our graph-based model. The first entity (nodes) collects a set of ports and their different properties such as the label or geographical coordinates. The edge entity is a segment delimited by a starting and an ending point, described respectively as a *source_id* to identify a port and a *target_id* to identify also a port (Figure 4). We identify in each segment other important features to describe the temporal dimension like *source_time* (for a segment starting time) and *target_time* (for a segment ending duration). This allows to define the total time duration of a trip (*delta_time*).

Our graph representation is inspired from the study proposed by Tamilmani and Stefanakis [25] (Section 2). Hence, we defined a graph as a structure representing the trajectory network.

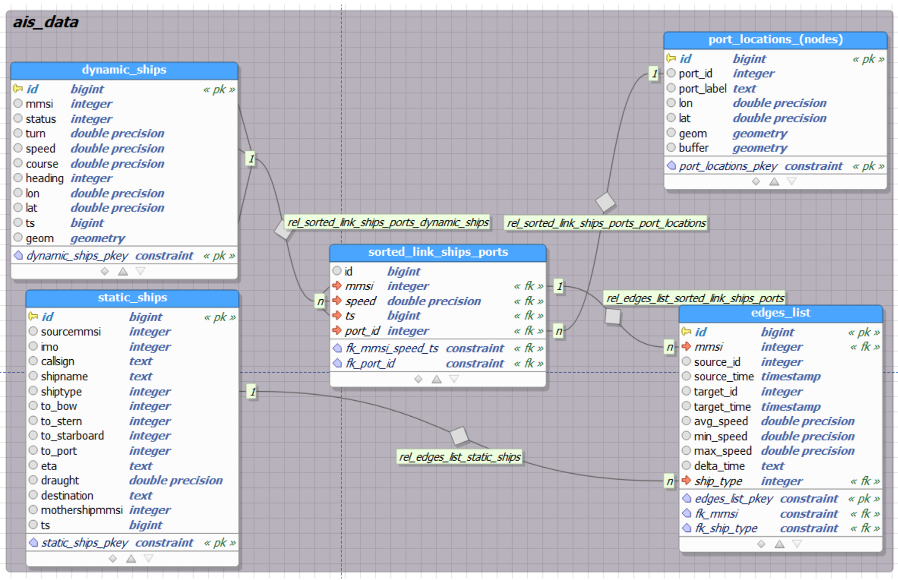


FIG. 4 – *PostgreSQL Data Base model (subset) illustrating the structure of the PoL graph [11]*

According to the representation in [25], each simplified trajectory node is related to a starting and ending points which are converted into origin and destination nodes (Figure 1). We apply the same concept except that the ending and starting points are described by *Port* nodes. The *simplified trajectory* node can refer to a *Segment* node in our approach. This node is linked to *Port* nodes with two different edges specified by *STARTS* and *ENDS* relationships. The first one links a *Segment* to its starting point and the second one to the ending point. Furthermore, these edges are *labeled* with temporal dimension. Since the PoL is partially associated to a vessel we add another node (*Ship*) to characterize it. Thus, the latter is related to one or more *Segment* nodes by an edge labeled *FOLLOWING*. The main idea of adding a Ship node is to allow the analysis of this entity in its own across the network for future perspectives.

Once the model defined, we analyze the structure in a graph database that implement the different entities and their connections. Thence, the process to manipulate the PoL analysis follows different steps : i) storing the data into a graph database –for this purpose we choose Neo4j as explained in next section– and ii) analyzing semantically the structure stored and nodes connectivity with Cypher (query language for Neo4j).

## 5.1 Technical motivations

Neo4j is the world's leading Graph Database Management Systems (GDBMS) [19] [4]. It is a Not only SQL (NoSQL) and an open source graph database written in Java and Scala. In database storage, ACID (atomicity, consistency, isolation, durability) is a an important set of properties that Neo4j provides which makes transactional operations possible [17]. The architecture of this GDBMS comes up with some core APIs (like JVM-based languages) for

4. https://db-engines.com/en/ranking, accessed November 2019

working with graphs. Since Neo4j is a graph database, it adopts a structure where the nodes are represented as vertices and the relationships as edges. The user can also add properties on edges and vertices which allows more semantic expressiveness across the database.

Handling the information retrieval of the database requires using a query language that provides a high level of expressiveness and an efficient query execution. Therefore, Cypher was implemented for Neo4j to simply manage the graph database. It is a declarative language inspired from SQL but more simple to perform. In fact, the graph database are used for entities connectivity modeling, so using Join queries in SQL to extract information is more complex to compute and to express than using Cypher. This query language includes some spatial built-ins to enable geographic capabilities. Likewise for temporal dimension, Cypher provides several time functions.

## 5.2 Neo4j modeling and data enrichment

Our graph modeling is easily transferred into Neo4j graph database. For this aim we collect the data stored in csv. files – related to the data extracted from the relational database (Figure 4) – to enrich our graph database with Cypher queries through a core API for Java programming. Each record in the dataset from *edges_list* entity represents one segment. So from this entity, we enrich the segment node by properties which are attributes in the relational database. More specifically, *id_segment*, *max_speed*, *min_speed* and *avg_speed* are implemented as properties for *Segment* node. The *mmsi* attribute remains as property for the *Ship* node in addition to its type (other properties, from *static_ships* table, must be loaded in order to perform certain mining queries related to vessels). In fact, we do not need to include *mmsi* property for the *Segment* node because the relation between *static_ships* and *edges_list* table is expressed by the edge *FOLLOWING* between *Ship* and *Segment* nodes. For the *Port* node, the data enrichment is performed from *Port_Locations_(nodes)* table.

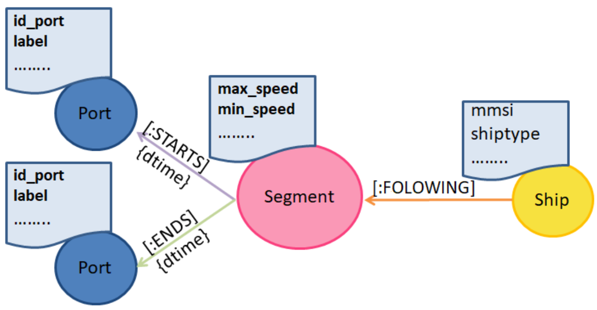


FIG. 5 – *Graph Model for PoL managed by Neo4j*

In the *edges_list* table, the attributes related to *Port* (*source_id* and *target_id*) are converted into *STARTS* and *ENDS* edges with respectively the properties *source_time* and *target_time*. These properties are described by a same label *dtime* for day time (which is a format supported by Neo4j). The Figure 5 details the graph modeling handled by Neo4j in our approach. The graph database in our case study is covered by more than 68,000 nodes and more than 90,000 relationships types. The Figure 6 illustrates an example of loaded data where we can see a ship with *mmsi:"211271630"* following several segments like the one with *avg_speed:4.503477465*. This segment has a starting point at port having *id_port:116* and an ending point with *id_port:112*. The starting time is defined with *dtime:"2015-10-03T13:43"* and the trip for this segment ended with *dtime:"2015-10-03T14:33"*.

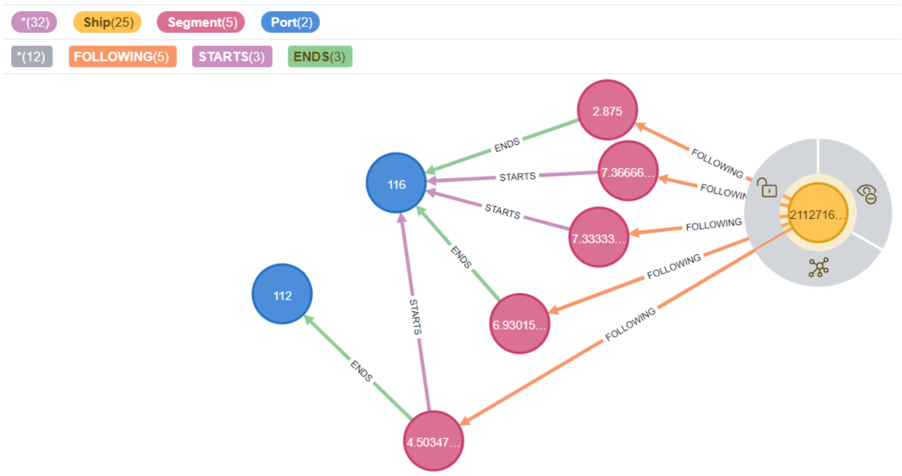


FIG. 6 – *Example (from Neo4j browser) of data enrichment for the graph database stored into Neo4j*

With the spatio-temporal dataset loaded in the graph database, queries using Cypher are performed for semantic analysis. We present in what follows some of those queries and the possible analysis that can be done through our graph modeling.

# 6 Query analysis using Cypher

Cypher provides a bunch of operations and functional graph algorithms that offer a semantic interpretation of a data set with a graph modeling in mind. Actually, by using queries like pattern matching or analytical queries, this semantic interpretation leads us to understand the PoL associated to different entities in our case study. Here we present and classify some primitive queries according to relevant features (such as flow, time constraint, point of interest...). The average execution time for *pattern matching* queries is around 200ms.

**A. Outgoing and incoming flow:**
For statistical analyses purpose, one may want to check the influence of ports (e.g.the more/less active port) in the network and their connectivity or adjacency. Hence, it's important to get information about the port nodes degrees. With this kind of query, we have an answer for how much a port is important in the network.

**Query**

```
MATCH (p:Port{id_port:"86"})
MATCH ()-[s:STARTS]->()
RETURN distinct (size((p)<-[:STARTS]-())*100)/count(s) AS PortSource
```

This query allows us to get the percentage of outgoing traffic flow from one specific port allover the network. In the same way, it is also possible to analyze the incoming flow. We first get the matching pattern of the port in question and then all patterns that match the outgoing flow. Finally, some aggregation functions are needed to get the result. In our case study, the port with *id_port:86* has a score of 14% for the outgoing flow.

**B. Historical life cycle:**
In PoL concept, it is relevant to get the frequency movement of an object between points of interest. This means that for our ambitions, it is important to understand the ship characteristics over the network. The frequency of vessel's trips should be analyzed. Moreover, with time constraint, getting this frequency may entails the identification of changing routes for some specific vessels for example.

**Query**

```
MATCH (s:Ship{mmsi:''244901000''})
RETURN size((s)-[:FOLLOWING]->()) AS SegmentCount
```

The query in this case, focuses on the ship with *mmsi:"244901000"* identification. More specifically, by using aggregation functions and having our graph modeling in mind, this query reveals how many trips one vessel carries out. The result for this query informs the ship in question traveled over 11 segments. Neo4j provides a web interface to manage and visualize the graph or derived sub-graphs. The Figure 7 shows the different segments (trips) produced by the ship (with *mmsi:"244901000"*).

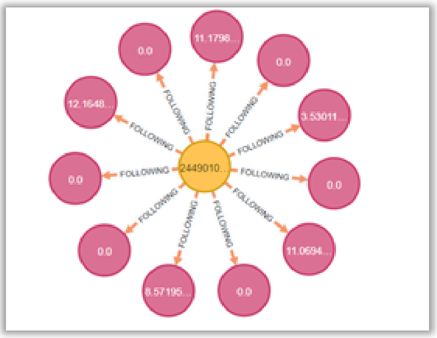


FIG. 7 – *Displaying in Neo4j browser example of Segments followed by one Ship*

**C. Outgoing and incoming analysis with time constraint:**
The pertinence of temporal dimension has no need to be proven in PoL notion. In fact, the PoL analysis requires historical data, in other words: time constraint. Here we present a way to obtain informative knowledge about ports' activity in a certain period of time. With this knowledge, it is possible to get not only the ports visited by vessels but also the remaining ports with no flow.

**Query**

```
MATCH (h:Ship)-[f:FOLLOWING]->(s:Segment)-[e]-(p:Port)
WHERE e.dtime >datetime(''2016-01-31'')
RETURN distinctp,h
```

In this Cypher query, we match patterns that follow outgoing and incoming flows from ports, produced by ships starting from a specific date. We explain the query as follows: Ships are following different *Segments* that have a starting and ending points which are *Ports*. The time constraint is applied to when vessels pass through ports.

**D. Meeting Points detection using time constraint:**
Sometimes in Maritime traffic surveillance a suspicious trajectory involves interactions between vessels. In this context, it is possible to identify meeting points (ports in our case) for a specific ship with others. To do so, we must use time constraint.

**Query**

```
MATCH (h:Ship)-[f:FOLLOWING]->(s:Segment)-[r]-(p:Port)
<-[g]-(d:Segment)<--(t:Ship{mmsi:''244901000''})
WHERE r.dtime=g.dtime
RETURN distinct h,p,r.dtime,g.dtime,d,s
```

We use time constraint in this query that matches the moment when the boat meets another one in a port. Through this interrogation, we get a knowledge about vessels and time/place of meeting. The returning results shows that the ship with *mmsi:"244901000"* identification has two meeting points (in *id_port: 4* and *id_port:222*) with 4 other vessels.

**E. New edges and trajectory prediction perspectives:**
In order to predict the next segment of a vessel, we should have the percentage of outgoing flow from one port to another overall outgoings. This entails the creation of other edges that define directly the connectivity between ports for each vessels. More specifically, for each segment, we link its starting point to its ending point by an edge labeled *LEADS*. This edge has as a property, the mmsi of the vessel that follows the segment formed by the two points. For instance in Figure 8 the ship with *mmsi:"211271630"* follows a segment that has a starting point at port having *id_Port:116* and an ending point with *id_Port:112*.

These two ports are linked with an oriented edge (LEADS) from *Port:116* to *Port:112*. Here is the query that managed this edges updates.

***i) Query to generate the LEADS relationship type***

**Query**

```
MATCH (p:Port)<-[t:STARTS]-(s:Segment)-[e:ENDS]->(p1:Port)
MATCH (h:Ship)-[f:FOLLOWING]->(s)
CREATE (p)-[:LEADS{mmsi:h.mmsi}]->(p1)
RETURN distinct h.mmsi
```

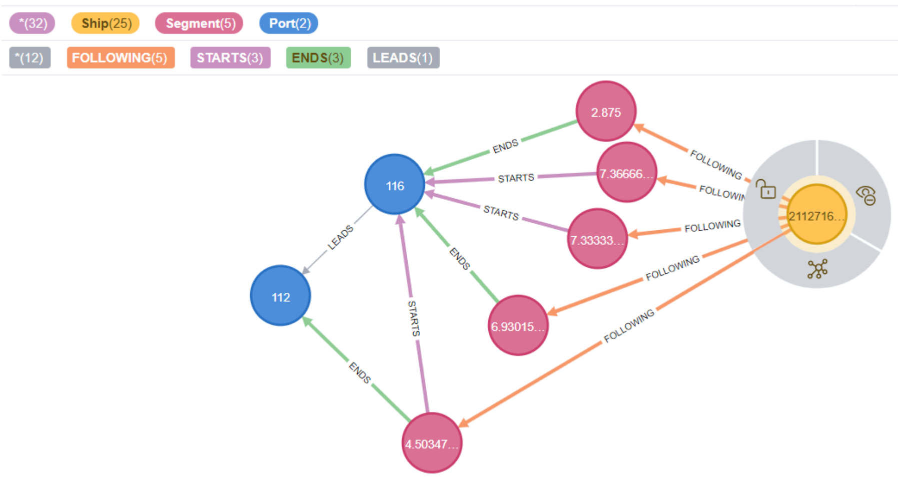


FIG. 8 – *Example of LEADS edge creation*

***ii) Query for percentage connectivity between ports***

**Query**

```
MATCH (p:Port)-[l:LEADS{mmsi:''227569380''}]->(p1:Port)
WITH p,p1, count(l) as sizeL
MATCH (p)-[l1:LEADS{mmsi:''227569380''}]->()
WITH p,sizeL,collect(p1) as ports, count(l1) as sizeL1
RETURN distinct p, head(ports).id_port, sizeL, sizeL1, (sizeL*100.0/sizeL1)
as percentage
```

The result of this query shows for instance that for the vessel with *mmsi:"227569380"* identification, when passing by the *Port:90*, about 35% of the outgoings flow move to *Port:86* and the remaining flow, about 64%, moves to *Port:91*. These results may help to run a prediction process.

# 7 Perspectives and discussion

We develop in this paper the notion of pattern of life (PoL) analysis applied to maritime traffic. To do so, a PoL graph modeling is presented and implemented in Neo4j graph database. The study of some recurrent ship's trips and activities through points of interest nodes is queried using Cypher query language. This PoL modeling is a important step to extend our results to more interesting result. Cypher provides several functions supporting graph algorithms. Applying those built-ins on our graph database enhances the analysis of the patterns produced by the entities coexistence in the graph. We can quote centrality algorithms (pagerank algorithm, betweenness centrality algorithm ...) or community detection algorithms (louvain algorithm, label propagation algorithm...). To conclude, we believe that graph model database is a promising approach to enhance PoL services for maritime traffic while providing semantic interpretation.

## Acknowledgment

This work belongs to Common Information Sharing Environment (CISE) Data Mining Services (DMS), which has received funding from the COSME program of the European Union under grant agreement No. EASME/EMFF/2016/1.2.1.9/03/SI2.763845 (CISE DMS).

## References

[1] Giulia Battistello and Wolfgang Koch. Knowledge-aided multi-sensor data processing for maritime surveillance. In *GI Jahrestagung (2)*, pages 796–799, 2010.

[2] Piotr Borkowski. The ship movement trajectory prediction algorithm using navigational data fusion. *Sensors*, 17(6):1432, 2017.

[3] Ahmad C Bukhari, Inara Tusseyeva, Yong-Gi Kim, et al. An intelligent real-time multi-vessel collision risk assessment system from vts view point based on fuzzy inference system. *Expert systems with applications*, 40(4):1220–1230, 2013.

[4] Trinh Minh Tri Do and Daniel Gatica-Perez. The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data. *IEEE Transactions on Mobile Computing*, 13(3):638–648, 2014.

[5] Shaojun Gan, Shan Liang, Kang Li, Jing Deng, and Tingli Cheng. Ship trajectory prediction for intelligent traffic management using clustering and ann. In *Control (CONTROL), 2016 UKACC 11th International Conference on*, pages 1–6. IEEE, 2016.

[6] Jianjun Gao, Haibin Ling, Erik Blasch, Khanh Pham, Zhonghai Wang, and Genshe Chen. Pattern of life from WAMI objects tracking based on visual context-aware tracking and infusion network models. In Ivan Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition XXII*, volume 8745, pages 528 – 536. International Society for Optics and Photonics, SPIE, 2013.

[7] Leticia Gomez, Bart Kuijpers, and Alejandro Vaisman. Analytical queries on semantic trajectories using graph databases. *Transactions in GIS*, 23(5):1078–1101, 2019.

[8] G. A. Gross, E. Little, B. Park, J. Llinas, and R. Nagi. Application of multi-level fusion for pattern of life analysis. In *18th International Conference on Information Fusion (Fusion)*, pages 2009–2016, July 2015.

[9] Marco Guerriero, Peter Willett, Stefano Coraluppi, and Craig Carthel. Radar/ais data fusion and sar tasking for maritime surveillance. In *Information Fusion, 2008 11th International Conference on*, pages 1–5. IEEE, 2008.

[10] Simen Hexeberg, Andreas L Flåten, Edmund F Brekke, et al. Ais-based vessel trajectory prediction. In *Information Fusion (Fusion), 2017 20th International Conference on*, pages 1–8. IEEE, 2017.

[11] A. ITANI, C. RAY, A. El Falou, and J. Issa. Mining ship motions and patterns of life for the eu common information sharing environment (cise). In *OCEANS 2019 - Marseille*, pages 1–6, 2019.

[12] Tor A Johansen, Andrea Cristofaro, and Tristan Perez. Ship collision avoidance using scenario-based model predictive control. *IFAC-PapersOnLine*, 49(23):14–21, 2016.

[13] Alen Jugović, Svjetlana Hess, and Tanja Poletan Jugović. Traffic demand forecasting for port services. *Promet-Traffic&Transportation*, 23(1):59–69, 2011.

[14] Richard O Lane, David A Nevell, Steven D Hayward, and Thomas W Beaney. Maritime anomaly detection and threat assessment. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8. IEEE, 2010.

[15] Bo Liu, Erico N de Souza, Cassey Hilliard, and Stan Matwin. Ship movement anomaly detection using specialized distance measures. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 1113–1120. IEEE, 2015.

[16] Fabio Mazzarella, Alfredo Alessandrini, Harm Greidanus, Marlene Alvarez, Pietro Argentieri, Domenico Nappo, and Lukasz Ziemba. Data fusion for wide-area maritime surveillance. In *Proceedings of the COST MOVE Workshop on Moving Objects at Sea, Brest, France*, pages 27–28, 2013.

[17] J.J. Miller. Graph database applications and concepts with neo4j. In *SAIS 2013*, volume 24, 2013.

[18] Gohar A Petrossian. Preventing illegal, unreported and unregulated (iuu) fishing: A situational approach. *Biological Conservation*, 189:39–48, 2015.

[19] Jaroslav Pokorný. Graph databases: Their power and limitations. In *14th Computer Information Systems and Industrial Management (CISIM)*, volume LNCS-9339, pages 58–69. Springer, 2015.

[20] Cyril Ray, Richard Dréo, Elena Camossi, Anne-Laure Jousselme, and Clément Iphar. Heterogeneous integrated dataset for maritime intelligence, surveillance, and reconnaissance. *Data In Brief*, 25:1–17, 2019.

[21] Stephen Roberts. *Anomaly detection in vessel track data*. PhD thesis, Oxford University, UK, 2014.

[22] Jean Roy and Eloi Bosse. Sensor integration, management and data fusion concepts in a naval command and control perspective. Technical report, Defence Research Establishment Valcartier (Québec), 1998.

[23] PAM Silveira, AP Teixeira, and C Guedes Soares. Use of ais data to characterise marine traffic patterns and ship collision risk off the coast of portugal. *The Journal of Navigation*, 66(6):879–898, 2013.

[24] Behrouz Haji Soleimani, Erico N De Souza, Casey Hilliard, and Stan Matwin. Anomaly detection in maritime data based on geometrical analysis of trajectories. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 1100–1105. IEEE, 2015.

[25] Rajesh Tamilmani and Emmanuel Stefanakis. Modelling and Analysis of Semantically Enriched Simplified Trajectories Using Graph Databases. In *Abstracts of the ICA*, volume 1, page 20, 2019.

[26] Iraklis Varlamis, Konstantinos Tserpes, Mohammad Etemad, Amílcar Soares Júnior, and

Stan Matwin. A network abstraction of multi-vessel trajectory data for detecting anomalies. *BMDA Workshop*, 2019.

[27] Tingting Xu, Xiaoming Liu, and Xin Yang. Ship trajectory online prediction based on bp neural network algorithm. In *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on*, volume 1, pages 103–106. IEEE, 2011.

## Résumé

L'analyse des données issus d'objets mobiles et plus particulièrement l'analyse des habitudes de vie (Pattern of Life) vise à comprendre les comportements, les tendances de mouvements des entités évoluant dans l'espace et le temps, ainsi que leur connectivité au travers des dimensions spatio-temporelles et sémantiques. Cet article présente une contribution basée sur une modélisation à base de graphe et qui gère ces dimensions. Notre modèle est implémenté dans la base de données graphes Neo4j et peuplé de données de trafic maritime. Avec le langage de requête Cypher, le modèle prend en charge de nombreuses méthodes pour analyser les trajectoires de navires. Nous exploitons les données historiques afin de démontrer certaines analyses typiques conduisant à une meilleure compréhension des routes et du trafic maritime. Ce travail fait partie d'un ensemble de services de fouille de données conçus pour le projet CISE (Common Information Sharing Environment) DMS (Data Mining Services) financé par la Commission européenne.