Online Co-movement Pattern Prediction in Mobility Data

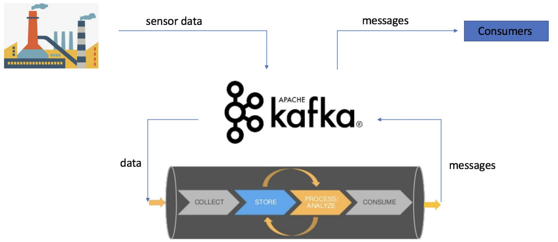
Andreas Tritsarolis, Eva Chondrodima, Panagiotis Tampakis, and Aggelos Pikrakis
Data Science Lab., University of Piraeus
{andrewt, evachon, ptampak, pikrakis}@unipi.gr
datastories.org

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Outline

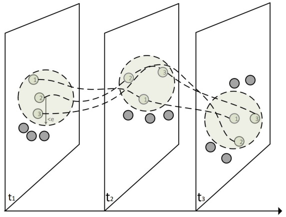
1. Introduction
2. Related Work
3. Problem Definition
4. Methodology
5. Evaluation Measures
6. Experimental Study
7. Conclusions and Future Work

Introduction

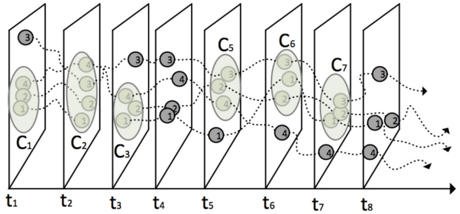


[source]

- Vast spread of GPS-enabled devices → Voluminous (mobility-related) data.
  - Common data sources: smartphones, tablets, GPS-enabled vehicles
- Streaming data pose new challenges
  - Efficient Storage
  - Knowledge Discovery
  - Online Analytics (e.g. Co-movement Patterns)

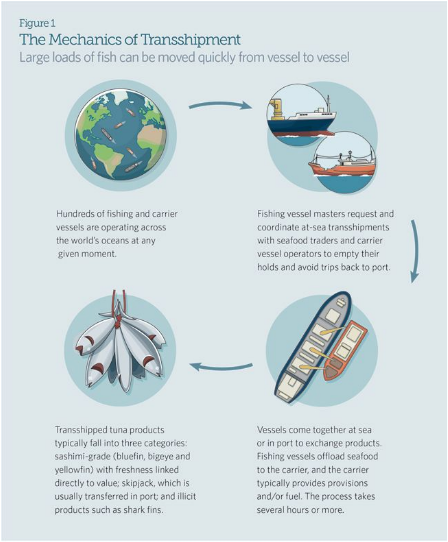


- Even more challenging is the task of *Online Prediction of Co-Movement Patterns*
  - A task that (to the best of our knowledge) has not been addressed in the relevant literature yet.



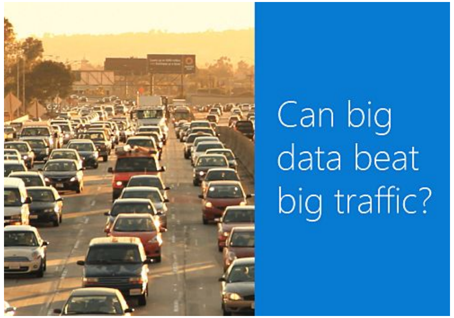
Introduction (cont.)

- Several mobility-related applications could benefit from such an operation.
- Urban Domain
  - Traffic Jams Detection → Help domain experts to take measures
  - Assist the Authorities in effectively adjusting traffic flow
- Maritime Domain
  - Detection of Illegal transshipment
  - Control fishing effort → Reduce fishing pressure in (protected) fishing areas
- Contact Tracing
  - Identify individuals that have been close to infected persons for some time duration → Help avoid future contacts with possibly infected individuals.

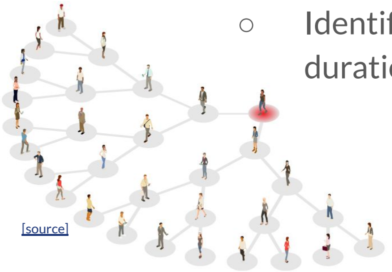


Source: The Pew Charitable Trusts

[source]



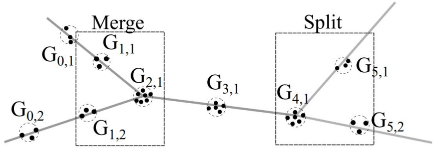
[source]



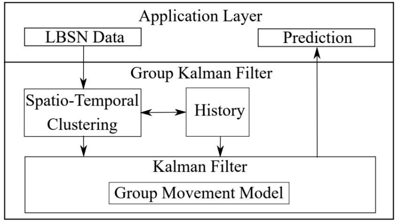
[source]

Related Work

- The closest work to ours has only been recently authored in [Kannangara et al. 2020]



- More specifically, the authors:
  - Divide time into time slices of fixed step size
  - Adopt a spherical definition of groups
    - Each group consists of moving objects that are confined within a radius *d*.
  - Use Kalman Filters (Group Kalman Filter - GKF) to track the movement of groups using Location Based Social Network (LBSN) data

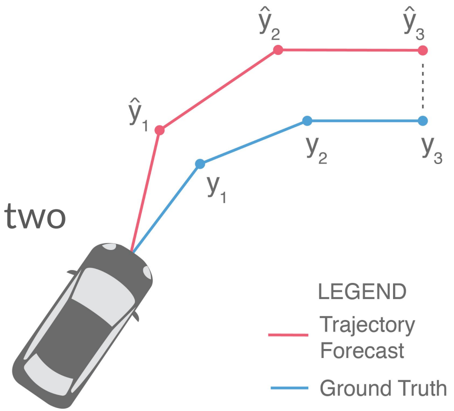


- However, we deviate from the above works in the following aspects:
  - Our work is Online → Can be used in a timely manner within streaming datasets (Kafka Topics)
  - We cluster our data-points using both Spherical- and Density-based methods
  - Predict not only the trajectory of a single cluster, but also its shape and members

Our Contribution

- Our contribution consists of:
  - An accurate solution to the problem of *Online Prediction of Co-movement Patterns*.
  - A similarity metric, which helps us “match” the predicted with the actual clusters.
- Our study, using a large-volume real-world marine traffic dataset verifies:
  - The efficiency of the aforementioned algorithm towards
    - Voluminous streaming mobility data
    - Prediction Accuracy
  - Its value towards a generic tool able to monitor and uphold traffic safety
- To the best of our knowledge this problem has not been addressed in the relevant literature yet

Problem Definition



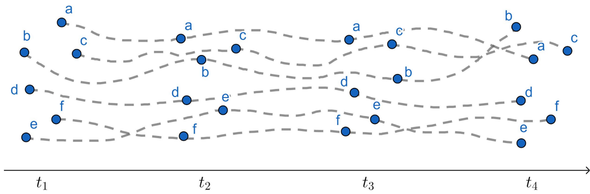
[source]

- The problem of *Online Prediction of Co-movement Patterns* is divided into two sub-problems
  - Future Location Prediction [Tampakis et al. 2020]
  - Evolving Clusters [Tritsarolis et al. 2020]

- Def. 1 (Trajectory):
  - A trajectory $T = \{p_1, \ldots, p_n\}$ is considered as a sequence of timestamped locations, where n is the latest reported position of T. Further, $p_i = \{x_i, y_i, t_i\}$, with $1 \leq i \leq n$.

- Def. 2 (Future Location Prediction):
  - Given an input dataset $D = \{T_1, \ldots, T_{|D|}\}$ of trajectories and a time interval Δt, our goal is $\forall$ $T_i \in D$ to predict $p^i_{pred} = \{x^i_{pred}, y^i_{pred}\}$ at timestamp $t^i_{pred} = t^i_n + \Delta t$.

Problem Definition (cont.)



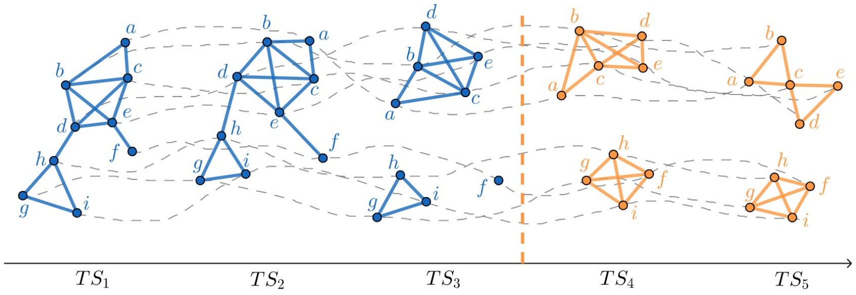
- Informally, group patterns can be regarded as "*a large enough number of objects moving close enough to each other, in space and time, for some time duration*".

- Def. 3 (Evolving Cluster) — Given:
  - Dataset *D* of trajectories,
  - Minimum cardinality threshold *c*,
  - Maximum distance threshold *θ*, and
  - Minimum time duration threshold *d*

- An Evolving Cluster $\langle C, t_{start}, t_{end}, tp \rangle$ is a subset $C \in D$ of the moving objects' population, $|C| \geq c$, which appeared at point t_{start} and remained alive until t_{end} (with $t_{end} - t_{start} \geq d$) during the lifetime $[t_{start}, t_{end}]$ of which the participating moving objects were spatially connected with respect to distance *θ* and cluster type *tp*.

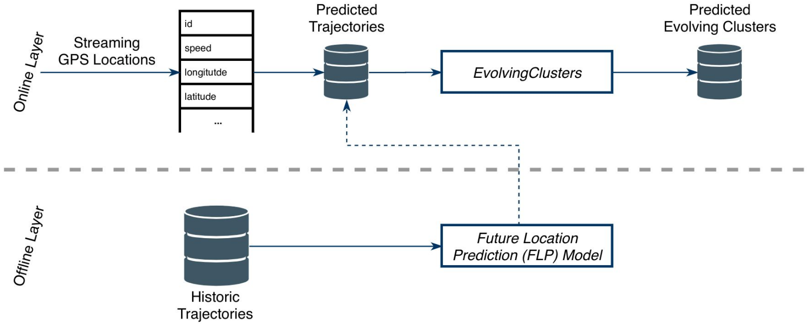
Problem Definition (cont.)

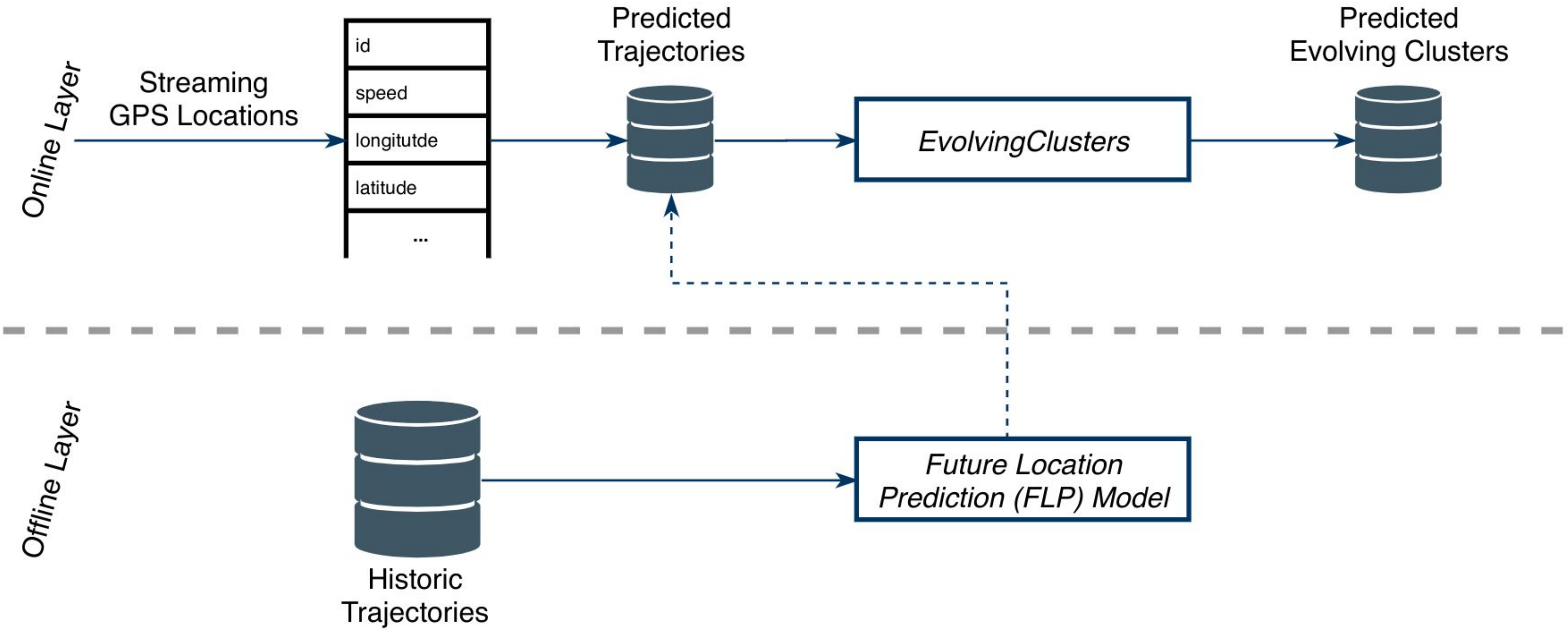
- Def. 4 (Group Pattern Prediction Online)
  - Given: a Dataset *D* of trajectories,
  - *G* of co-movement patterns up to time-slice TS_{now}, and
  - Lookahead threshold Δt, we aim to predict all the valid co-movement patterns $G' \in (TS_{now}, TS_{now} + \Delta t)$
- Our goal is to predict their respective locations until TS_5. Running EvolvingClusters with the same parameters for the predicted time-slices, reveals us (with high probability)
  - Some patterns will continue to exist (e.g. *{a,b,c,d,e}*); as well as
  - Some new patterns will be formed (e.g. *{f,g,h,i}*)



Methodology

- As mentioned before, we split the problem of *Online Prediction of Co-movement Patterns* into two parts
  - Future Location Prediction (FLP)
  - Evolving Cluster Discovery
- The FLP method consists of two parts:
  - FLP-offline → Model Training from Historic Trajectories
  - FLP-online → Model Prediction from Streaming Trajectories

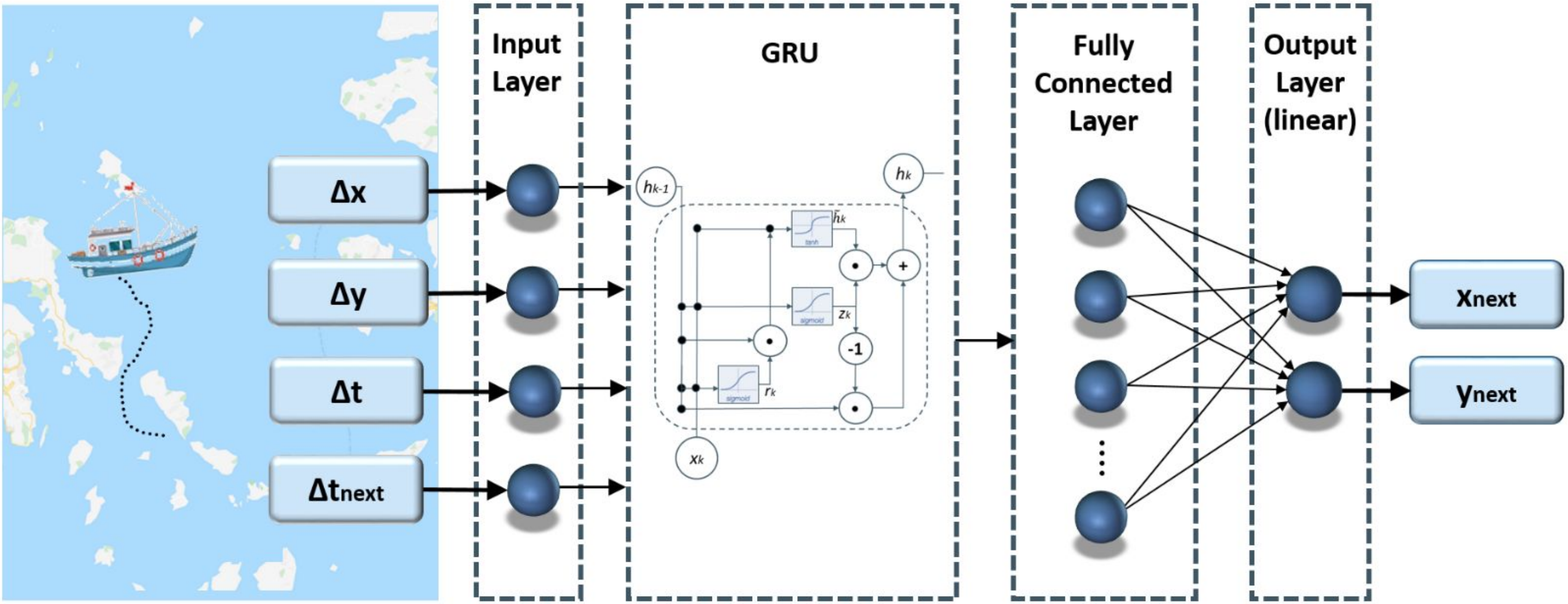




Workflow for evolving clusters prediction via (singular) trajectory prediction

Methodology (cont.)

- Trajectories can be considered as time-series
  - GPS Positions ordered by Time
- Over the past two decades, the research interest has been moved to RNN-based models
  - LSTM
  - GRU
- In this work, we employ a GRU-based model
  - Less complicated
  - Easier to modify
  - Faster to train
- Depending on the context, GRU-based networks perform better than LSTM-based ones

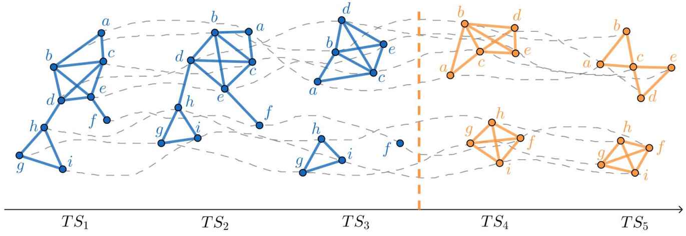


GRU-based neural network architecture

Methodology (cont.)

- Having predicted the moving objects' future locations, we employ EvolvingClusters → Present the predicted co-movement patterns
- Variable sampling rate → Temporal alignment (via Linear Interpolation)
- Given a timeslice TS_{now}:
  - Creates its adjacency matrix given the objects' pairwise distance with respect to the threshold θ
  - Extract the Maximal Connected Subgraphs (MCS) and Cliques (MC) with respect to c
  - Maintains the currently active (and inactive) clusters, given the MCS and MC of TS_{now} and the recent (active) pattern history
  - Outputs the eligible active patterns with respect to c, t, and θ.

Methodology (cont.)



- The output of Evolving Clusters, and by extension of the whole model, is a tuple of four elements for each respective evolving cluster
  - The set o_{ids} of objects that it consists of
  - Its corresponding start and end times *st*, *et*; as well as,
  - Its corresponding type *tp* (1: MC, 2: MCS)
- In the previous example, and assuming c=3 and d=2,
  - $\{(P_2, TS_1, TS_5, 2), (P_3, TS_1, TS_5, 1), (P_4, TS_1, TS_4, 1), (P_5, TS_1, TS_5, 1)\} \rightarrow$ The clusters from the historic trajectories that continue to exist
  - $\{(P_4, TS_1, TS_5, 2),(P_6, TS_4, TS_5, 1)\} \rightarrow$ The newly predicted clusters
  - Furthermore,
    - P_4 becomes inactive at timeslice TS_5, but it remains active as an MCS
    - At timeslice TS_5, a new evolving cluster, P_6, is discovered

P_2	$\{a,b,c,d,e\}$
P_3	$\{a,b,c\}$
P_4	$\{b,c,d,e\}$
P_5	$\{g,h,i\}$
P_6	$\{f,g,h,i\}$

Evaluation Measures

- Evaluating predicted co-movement patterns → Not a straightforward task!
  - We need to quantify the deviation between the actual and predicted co-movement pattern

- Intuition → Match each predicted co-movement pattern with the most similar actual one

- The overall similarity is calculated as the weighted average of the below metrics

$$Sim^*(C_{pred}, C_{act}) = \begin{cases} \lambda_1 \cdot Sim^{spatial} + \\ \lambda_2 \cdot Sim^{temp} + \quad & Sim^{temp} > 0 \\ \lambda_3 \cdot Sim^{member} & \\ 0 & Else \end{cases} \tag{8}$$

$$Sim^{spatial}(C_{pred}, C_{act}) = \frac{MBR(C_{pred}) \bigcap MBR(C_{act})}{MBR(C_{pred}) \bigcup MBR(C_{act})} \tag{5}$$

$$Sim^{temp}(C_{pred}, C_{act}) = \frac{Interval(C_{pred}) \bigcap Interval(C_{act})}{Interval(C_{pred}) \bigcup Interval(C_{act})} \tag{6}$$

$$Sim^{member}(C_{pred}, C_{act}) = \frac{|C_{pred} \bigcap C_{act}|}{|C_{pred} \bigcup C_{act}|} \tag{7}$$

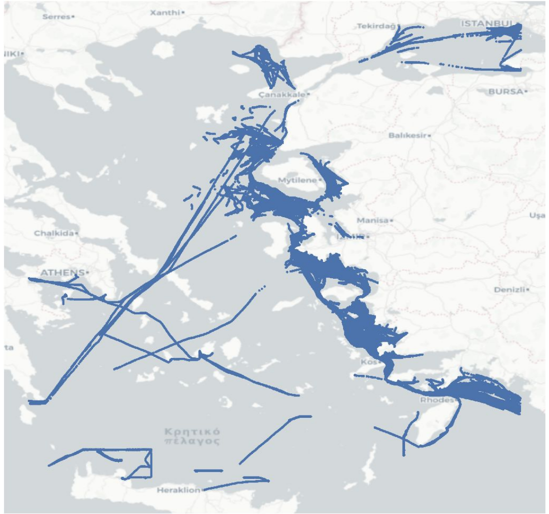
Algorithm 1: CLUSTERMATCHING. Matches the predicted with the actual evolving clusters

Input: Evolving Clusters discovered using the predicted EC_p; and actual EC_a data-points; Measures' weights $\lambda_i, i \in \{1, 2, 3\}$
Output: "Matched" Evolving Clusters EC_m

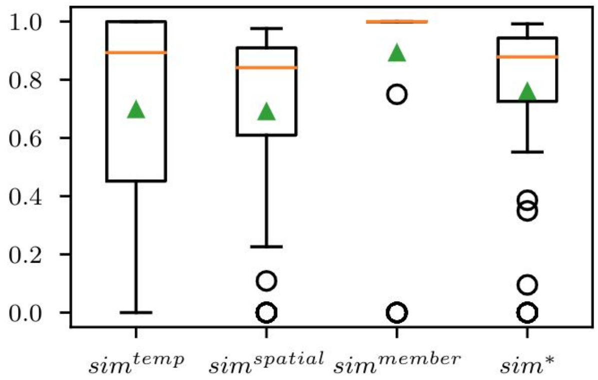
```
EC_m ← {}
for predicted pattern C_pred ∈ EC_p do
    similarity_scores ← {}
    topSim = 0
    for actual pattern C_act ∈ EC_a do
        calculate Sim*(C_pred, C_act)
        if Sim*(C_pred, C_act) ≥ topSim then
            topSim = Sim*(C_pred, C_act)
            match_best ← C_act
        end
    end
    EC_m ← EC_m ∪ match_best
end
```

Experimental Study

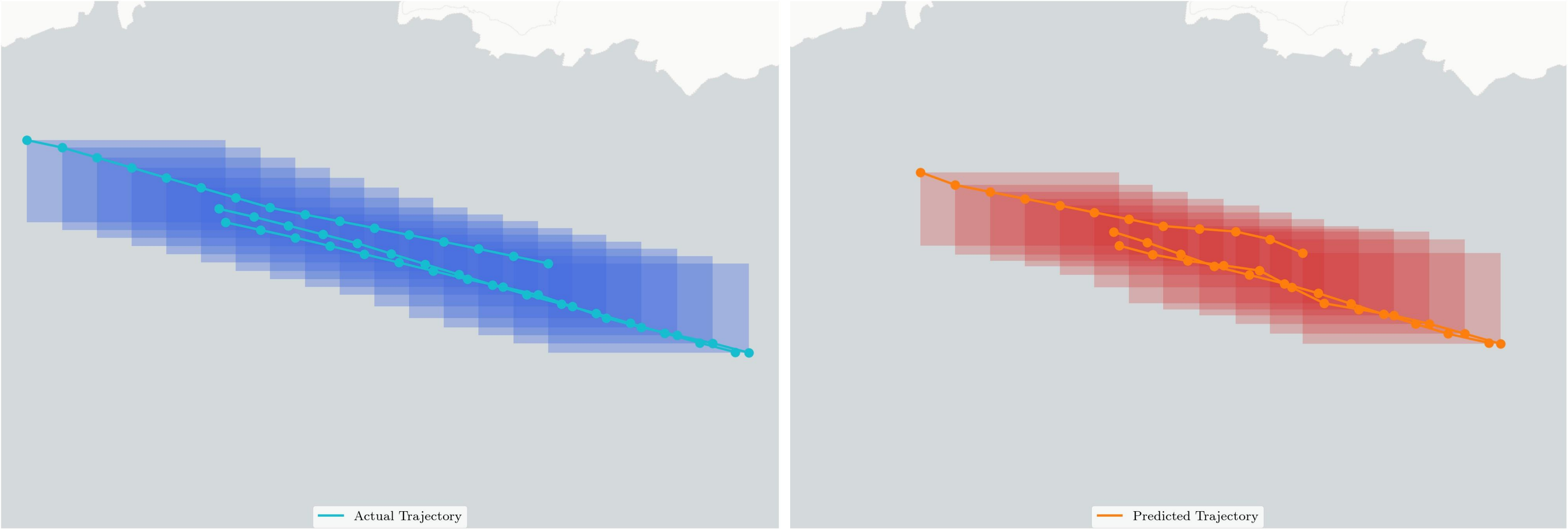
- For the experiments that will follow we use a real-life mobility dataset
    - Kindly provided by MarineTraffic
- Contains information on maritime traffic in Aegean Sea
    - Temporal Range: June 2nd, 2018 - August 1st, 2018
    - Spatial Range: longitude in [23.006, 28.996]; latitude in [35.345, 40.999]
- Preprocessing Pipeline:
    - Drop erroneous points w.r.t. a speed threshold $speed_{max}$ = 50 Knots
    - Drop stop points (i.e. locations with near-zero speed)
    - Organize remaining points into trajectories w.r.t. a temporal threshold *dt* = 30 min.
- The rationale behind these thresholds → stems from statistical analysis of the distribution of the locations' speed and dt



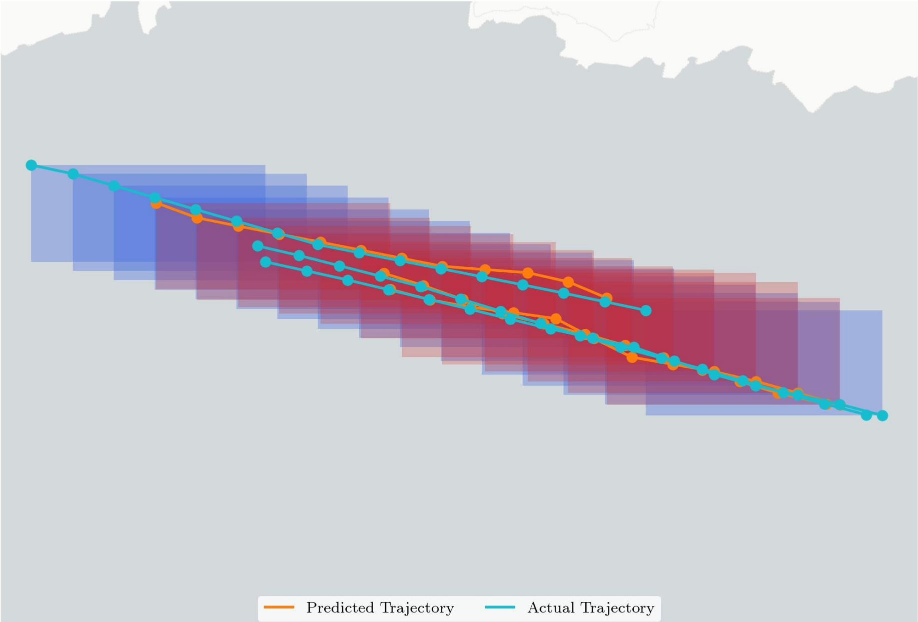
Experimental Study (cont.)



- Assuming thresholds for EvolvingClusters
  - *c = 3 objects*
  - *d* = 3 timeslices
  - *θ* = 1500 meters
- We observe that the majority of the predicted clusters is very close to the “ground truth”
  - Overall median similarity ~88%
- This is expected → the quality of EvolvingClusters’ output is determined by two factors
  - Selected parameters
  - Input data
- Focusing on the latter → the algorithm is quite insensitive to prediction errors
  - Deviations from the actual trajectory have minor impact to $sim_{spatial}$



Illustrating the actual vs. the predicted vessels' trajectories of an evolving cluster



Superimposing the actual vs. the predicted evolving clusters. Notice the temporal/spatial overlap.

Conclusions and Future Work

- Proposed an accurate solution to the problem of Online Prediction of Co-movement Patterns
  - Based on a combination of GRU models and Evolving Cluster Detection algorithm
  - A valuable utility for both researchers and practitioners alike
- Proposed a useful metric, able to match the predicted clusters with the actual ones
- In the near future we:
  - Aim to develop an online co-movement pattern prediction approach that, instead of two disjoint steps, will combine them in a unified solution
  - Able to directly predict the future co-movement patterns, in an accurate and timely manner

Acknowledgements

This work was partially supported by projects i4Sea (grant T1EDK-03268) and Track&Know (grant agreement No 780754), which have received funding by the European Regional Development Fund of the EU and Greek national funds (through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call Research-Create-Innovate) and the EU Horizon 2020 R&I Programme, respectively.

Thank You!

i4SEA Track & Know

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS