



Predicting Co-movement patterns in mobility data

Andreas Tritsarolis¹ · Eva Chondrodima¹ · Panagiotis Tampakis² · Aggelos Pikrakis¹ · Yannis Theodoridis¹

Received: 23 November 2021 / Revised: 2 August 2022 / Accepted: 8 September 2022 /

Published online: 22 September 2022

© The Author(s) 2022

Abstract

Predictive analytics over mobility data is of great importance since it can assist an analyst to predict events, such as collisions, encounters, traffic jams, etc. A typical example is anticipated location prediction, where the goal is to predict the future location of a moving object, given a look-ahead time. What is even more challenging is to be able to accurately predict collective behavioural patterns of movement, such as co-movement patterns as well as their course over time. In this paper, we address the problem of *Online Prediction of Co-movement Patterns*. Furthermore, in order to be able to calculate the accuracy of our solution, we propose a co-movement pattern similarity measure, which facilitates the comparison between the predicted clusters and the actual ones. Finally, we calculate the clusters' evolution through time (survive, split, etc.) and compare the cluster evolution predicted by our framework with the actual one. Our experimental study uses two real-world mobility datasets from the maritime and urban domain, respectively, and demonstrates the effectiveness of the proposed framework.

Keywords Machine learning · Predictive analytics · Co-movement patterns · Trajectory prediction · Cluster evolution

✉ Andreas Tritsarolis
andrewt@unipi.gr

Eva Chondrodima
evachon@unipi.gr

Panagiotis Tampakis
ptampakis@imada.sdu.dk

Aggelos Pikrakis
pikrakis@unipi.gr

Yannis Theodoridis
ytheod@unipi.gr

¹ Data Science Lab., Department of Informatics, University of Piraeus, Piraeus, Greece

² Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

1 Introduction

The vast spread of GPS-enabled devices, such as smartphones, tablets, and GPS trackers, has led to the production of large amounts of mobility related data. By nature, this kind of data is streaming and there are several application scenarios where the processing needs to take place in an online fashion. These properties have posed new challenges in terms of efficient storage, analytics, and knowledge extraction out of such data. One of these challenges is online cluster analysis, where the goal is to unveil hidden patterns of collective behaviour from streaming trajectories, such as co-movement patterns [4, 9, 10, 13, 35]. What is even more challenging is predictive analytics over mobility data, where the goal is to predict the future behaviour of moving objects, which can have a wide range of applications, such as predicting collisions, future encounters, traffic jams, etc. At an individual level, a typical and well-studied example of such analytics is future location prediction [11, 24, 25, 29, 33], where the goal is to predict the future location of a moving object, given a look-ahead time. However, prediction of future mobility behaviour at a collective level has not been addressed at the degree of its individual counterpart.

Concerning the definition of co-movement patterns, there are several approaches in the literature, such as [4, 9, 10, 13]. However, most of the above are either offline and/or operate at predefined temporal snapshots that imply temporal alignment and uniform sampling, which are not realistic assumptions. For this reason, we adopt the approach presented in [35], which, to the best of our knowledge, is the first online method for the discovery of co-movement patterns in mobility data that does not assume temporal alignment and uniform sampling. The goal in [35] is to discover co-movement patterns, namely *Evolving Clusters*, in an online fashion, by employing a graph-based representation. By doing so, the problem of co-movement pattern detection is transformed into identifying *Maximal Cliques* (MCs) (for spherical, Flock-like clusters) or *Maximal Connected Subgraphs* (MCSs) (for density-connected, Convoy-like clusters).

In particular, the problem that we address in this paper is the *Online Prediction of Co-movement Patterns*. Informally, given a look-ahead time interval Δt , the goal is to predict the clusters of moving objects that are anticipated to be shaped after Δt . Figure 1 illustrates such an example, where in blue and green we have the information at hand (past and current locations / clusters, respectively) whereas in orange we have the predictions. The problem we address is quite challenging, since, apart from the inherent difficulty of predicting the future, we also need to define how the error between the actual and the predicted

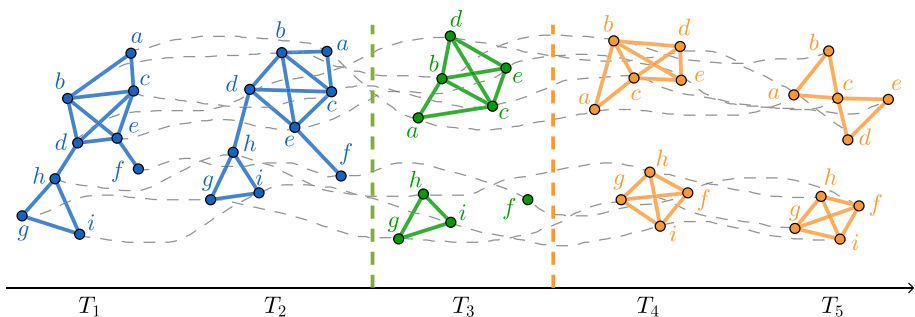


Fig. 1 Predicting evolving clusters via trajectory prediction – blue, green, and orange graphs correspond to past, current, and future (i.e., predicted) evolving clusters, respectively

clusters will be measured. This further implies that a predicted cluster should be correctly “matched” with the corresponding actual cluster, which is not a straightforward procedure at all. To the best of our knowledge, the problem we aim to address has not been addressed in the literature yet.

Several mobility-related applications could benefit from such an analytics task. In the urban traffic domain, predicting co-movement patterns could assist in detecting future traffic jams, which, in turn, could help the authorities take the appropriate measures (e.g. adjusting traffic lights) in order to minimize them. In the maritime domain, a typical problem is illegal transshipment, where groups of vessels move together “close” enough for some time duration and with low speed. It becomes obvious that predicting co-movement patterns could help in predicting illegal transshipment events.

Our main contributions are the following:

- We provide an efficient solution to the problem of *Online Prediction of Co-movement Patterns*.
- We propose a co-movement pattern similarity measure, which helps us “match” the predicted with the actual clusters.
- We exploit on MONIC/FINGERPRINT [22, 27], a state-of-the-art cluster evolution discovery method, in order to track the clusters’ evolution through time.
- We extensively evaluate the aforementioned framework using large-volume real-world data from two different mobility domains, namely, maritime and urban transportation.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, we formally define the problem at hand. Subsequently, in Section 4 we propose our methodology and we introduce a co-movement pattern similarity measure along with the cluster “matching” approach. Section 5, presents our experimental study and, finally, in Section 6 we conclude the paper and discuss future extensions.

2 Related work

The work performed in this paper is closely related to three topics, (a) co-movement pattern discovery, (b) future location prediction, and (c) co-movement pattern prediction.

Co-movement pattern discovery One of the first approaches for identifying collective mobility behaviour is the so-called flock pattern [17], which identifies groups of at least m objects that move within a disk of radius r for at least k consecutive timepoints. Inspired by this, several related works followed, such as moving clusters [15], convoys [14], swarms [19], platoons [18], traveling companion [32] and gathering pattern [39]. Even though all of these approaches provide explicit definitions of several mined patterns, their main limitation is that they search for specific collective behaviours, defined by respective parameters. An approach that defines a new generalized mobility pattern is presented in [9] where the general co-movement pattern (GCMP) is proposed. In [13], the authors propose a frequent co-movement pattern (f-CoMP) definition for discovering patterns at multiple spatial scales, also exploiting the overall shape of the objects’ trajectories, while at the same time it relaxes the temporal and spatial constraints of the seminal works (i.e. Flocks, Convoys, etc.) in order to discover more interesting patterns. The authors in [4, 10], propose

a two-phase online distributed co-movement pattern detection framework, which includes the clustering and the pattern enumeration phase, respectively.

Another perspective regarding co-movement pattern discovery is to reduce cluster types into graph properties and view them as such. In [35], the authors propose a novel co-movement pattern definition, called *Evolving Clusters (EC)*, that unifies the definitions of flocks and convoys and reduces them to Maximal Cliques (MC), and Connected Subgraphs (MCS), respectively. In addition, the authors propose an online algorithm, that discovers several evolving cluster types simultaneously in real time, without assuming temporal alignment, in contrast to the seminal works (i.e., flocks, convoys). In our work, we elaborate on *evolving clusters* [35] for co-movement pattern discovery. The reason why this is the most appropriate, is that we can predict the course of several pattern types at the same time, without the need to call several other algorithms, therefore adding redundant computational complexity.

Future location prediction The fact that the Future Location Prediction (FLP) problem has been extensively studied brings up its importance and applicability in a wide range of applications. Towards tackling the FLP problem, one line of work includes efforts that take advantage of historical movement patterns in order to predict the future location. Such an approach is presented in [33], where the authors propose MyWay, a hybrid, pattern-based approach that utilizes individual patterns when available, and when not, collective ones, in order to provide more accurate predictions and increase the predictive ability of the system. In another effort, the authors in [24, 25] utilize the work done by [31] on distributed subtrajectory clustering in order to be able to extract individual subtrajectory patterns from big mobility data. These patterns are subsequently utilized in order to predict the future location of the moving objects in parallel.

A different way of addressing the FLP problem includes machine learning approaches, such as Recurrent Neural Network (RNN) -based models [26] and Gated Recurrent Units (GRU) [5] models, which constitute the newer generation of RNN or static NNs [6, 28] [23]. In this line of research, Suo et al. [29] presented a GRU model to predict vessel trajectories based on the DBSCAN algorithm to derive main trajectories and a symmetric segmented-path distance approach to eliminate the influence of a large number of redundant data and to optimize incoming trajectories. Liu et al. [20] proposed a trajectory classifier called Spatio-Temporal GRU to model the spatio-temporal correlations and irregular temporal intervals prevalently presented in spatio-temporal trajectories. Particularly, a segmented convolutional weight mechanism was proposed to capture short-term local spatial correlations in trajectories along with an additional temporal gate to control the information flow related to the temporal interval information. Wang et al. [36] proposed a vessel berthing trajectory prediction model based on bidirectional GRU (Bi-GRU) and cubic spline interpolation. Hao et al. [12] proposed a vehicle trajectory prediction encoder-decoder model based on GRU with attention mechanism. Particularly, the proposed model is comprised of intention recognition module and trajectory prediction module. The former module recognizes driver's intention and calculates the probabilities of turning-left, lane-keeping, turning-right, whereas the trajectory prediction module predicts vehicle trajectory using GRU decoder with attention mechanism, which takes vehicle historical position as input and predicts future position. Zhang and Zheng [38] proposed a data-driven pedestrian trajectory predictor called MLP-social-GRU. The proposed mechanism first processes a pedestrian trajectory with a Multilayer Perceptron (MLP), and then, it adopts GRU to get hidden features of a pedestrian motion patterns, from which relationships between pedestrians can be simulated.

Co-movement pattern prediction The most similar to our work is presented in [16]. More specifically, the authors in [16], divide time into time slices of fixed step size and adopt a spherical definition of groups, where each group consists of moving objects that are confined within a radius d and their goal is to predict the centroid of the groups at the next timeslice. However, the group definition adopted in [16] is rather limited, since it identifies only spherical groups, as opposed to [8] where cylindrical clusters get detected and to [35] where both spherical and density-connected clusters can be identified. In addition, the Group Kalman Filter (GKF) they propose, while novel, it tracks only the centroid of each pattern, and not their shape and membership.

3 Problem definition

Before we proceed to the formulation of the problem, let us provide some preliminary definitions.

Definition 1 (Trajectory) A trajectory $T = \{p_1, \dots, p_n\}$ of a moving object is considered as a sequence of timestamped locations, where n corresponds to the latest reported position of T . Further, $p_i = \{x_i, y_i, t_i\}$, with $1 \leq i \leq n$.

Definition 2 (Future Location Prediction - FLP). Given a trajectory T_i and a time interval Δt , the goal of FLP is to predict $p_{pred}^i = \{x_{pred}^i, y_{pred}^i\}$ at timestamp $t_{pred}^i = t_{now}^i + \Delta t$.

Definition 3 (Evolving Cluster - EC) [35]. Given a dataset D of trajectories, a minimum cardinality threshold c , a maximum distance threshold θ , and a minimum time duration threshold d , an EC $\langle C, t_{start}, t_{end}, tp \rangle$ is a subset $C \in D$ of the moving objects' population, $|C| \geq c$, which appeared at time point t_{start} and remained alive until time point 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 (1 for maximal cliques - MC - or 2 for maximal connected components - MCS).

Definition 4 (Co-movement Pattern Prediction - CPP). Given a set D of trajectories, a set G of evolving clusters up to timeslice T_{now} , a time interval Δt , and a prediction step s (sub-multiple of Δt), the goal of CPP is to predict all valid evolving clusters G' up to $T_{now} + \Delta t$ with step s .

Definition 5 (Cluster Evolution Prediction - CEP). Given a set EC_{curr} of current ECs, a set EC_{pred} of predicted ECs up to a time interval Δt with step s (see Definition 4), a minimum cluster membership overlap threshold τ_{match} , a minimum cluster temporal overlap threshold τ_{temp} , and a minimum cluster split overlap threshold τ_{split} , the goal of CEP is to predict the evolution (emergence, disappearance, etc.) of the discovered clusters up to $T_{now} + \Delta t$ with step s , in accordance with the aforementioned three thresholds.

In particular, given two clusters C and C' from timeslices T_i and T_{i+1} , the transitions that a cluster might encompass, are as follows:

- Survival: $C \in T_i$ survives into $C' \in T_{i+1}$ iff C' is the match for C and there is no other cluster in T_i for which C' is the match.

- Absorption: $C \in T_i$ is absorbed into $C' \in T_{i+1}$ iff C' is the match for C and there is at least one additional cluster in T_i for which C' is the match.
- Split: $C \in T_i$ is split into $\{C'_1, C'_2, \dots, C'_n\} \in T_{i+1}, n > 1$ iff the overlap of C to each of the clusters C'_i is greater than a threshold τ_{split} and the overlap of all the clusters C'_i together with C is greater than a threshold τ_{match} .
- Disappearance: $C \in T_i$ disappears at T_{i+1} when C is not in the set of discovered clusters at T_{i+1} .
- Emergence: $C \in T_{i+1}$ emerges at T_{i+1} when C is not in the set of discovered clusters at T_i .

If we recall Fig. 1, it provides an illustration of Definition 4. More specifically, we know the movement of nine objects from T_1 until T_3 and (via EvolvingClusters with $c = 3$ and $d = 2$) the five evolving clusters that are formed: $P_1 = \langle \{a, b, c, d, e\}, T_1, T_3, 2 \rangle$, $P_2 = \langle \{a, b, c\}, T_1, T_3, 1 \rangle$, $P_3 = \langle \{b, c, d, e\}, T_1, T_3, 1 \rangle$, $P_4 = \langle \{g, h, i\}, T_1, T_3, 1 \rangle$, where $tp = 1(2)$ corresponds to MC (MCS, respectively). Our goal is to predict the anticipated evolution of ECs until T_5 , i.e., P_1, P_2, P_3, P_4 will continue to exist and two new patterns $P_5 = \langle \{f, g, h, i\}, T_4, T_5, 1 \rangle$, and $P_6 = \langle \{c, d, e\}, T_1, T_5, 1 \rangle$ will emerge.

4 Methodology

In this section, we present the proposed methodology in order to address the CPP and CEP problems, as defined in Definitions 4 and 5, respectively. The aforementioned framework follows the popular lambda architecture designed to handle massive quantities of data by taking advantage of both batch and stream-processing methods [21]. This architectural paradigm to balance latency, throughput, and fault-tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data. Figure 2 illustrates the architecture of our proposed methodology, consisting of an offline and an online layer. More specifically, at the offline layer we train our FLP-offline model, given a historical trajectory dataset, whereas at the online layer we receive the streaming GPS locations, predict the next objects' location (FLP-online module), and discover ECs at each timeslice. Finally, we compare the current with the predicted ECs, and output our prediction regarding the evolving clusters' evolution [22, 27] over time.

In the subsections that follow, we give more insights regarding the modules comprising our framework in Fig. 2. More specifically, in Sections 4.1 and 4.2 we describe two alternative models that can be used for FLP, while in Sections 4.3 and 4.4, we describe the modules in charge of EC discovery and cluster evolution, respectively.

4.1 GRU-based future location prediction

The first alternative to address the FLP problem (called GRU-FLP) employs a GRU-based method, utilizing the works done in [7, 30]. Subsequently, we briefly state the update rules for the employed GRU layer [5, 37].

$$\mathbf{z}_k = \sigma(\mathbf{W}_{\tilde{\mathbf{p}}z} \cdot \tilde{\mathbf{p}}_k + \mathbf{W}_{hz} \cdot \mathbf{h}_{k-1} + \mathbf{b}_z) \quad (1)$$

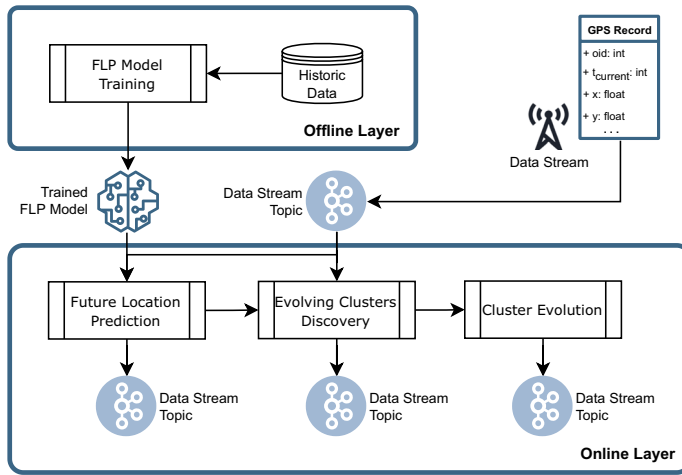


Fig. 2 Workflow for EC evolution prediction via FLP

$$\mathbf{r}_k = \sigma(\mathbf{W}_{\tilde{\mathbf{p}}r} \cdot \tilde{\mathbf{p}}_k + \mathbf{W}_{hr} \cdot \mathbf{h}_{k-1} + \mathbf{b}_r) \quad (2)$$

$$\tilde{\mathbf{h}}_k = \tanh(\mathbf{W}_{\tilde{\mathbf{p}}h} \cdot \tilde{\mathbf{p}}_k + \mathbf{W}_{hh} \cdot (\mathbf{r}_k * \mathbf{h}_{k-1}) + \mathbf{b}_h) \quad (3)$$

$$\mathbf{h}_k = \mathbf{z}_k \odot \mathbf{h}_{k-1} + (1 - \mathbf{z}_k) \odot \tilde{\mathbf{h}}_k \quad (4)$$

where $\mathbf{z}$ and $\mathbf{r}$ represent the update and reset gates, respectively, $\tilde{\mathbf{h}}$ and $\mathbf{h}$ represent the intermediate memory and output, respectively. Also, in these equations, the $\mathbf{W}_*$ variables are the weight matrices and the $\mathbf{b}_*$ variables are the biases. Moreover, $\tilde{\mathbf{p}}$ represents the input, which is composed of the differences in space ($\Delta x, \Delta y$), the difference in time (Δt), and the time horizon of the prediction (Δt_{next}).

Based on [7, 30], our GRU-based NN architecture (*GRU-FLP*) is composed of the following layers: a) an input layer of four neurons, one for each input variable, b) a single GRU hidden layer composed of 150 neurons, c) a fully-connected hidden layer composed of 50 neurons, and d) an output layer of two neurons, one for each prediction coordinate ($\Delta x_{next}, \Delta y_{next}$); a schematic overview of the proposed network architecture is presented in Fig. 3. Given the two most recent locations p_{i-1}, p_i of an object's trajectory, we predict Δx_{next} and Δy_{next} , which afterwards are summed with p_i , thus producing the predicted location of the aforementioned object.

4.2 Sub-trajectory based future location prediction

The second alternative to address the FLP problem (called ST-FLP) employs historical mobility patterns by utilizing the work done in [25]. In more detail, the problem is divided in its offline phase, which is responsible for identifying patterns of movement, and its online phase, which is responsible for predicting the future location of

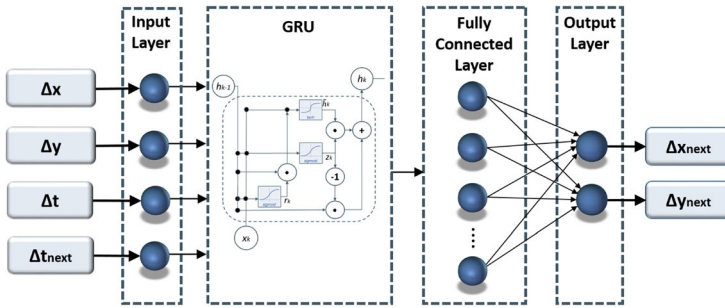


Fig. 3 GRU-based FLP architecture

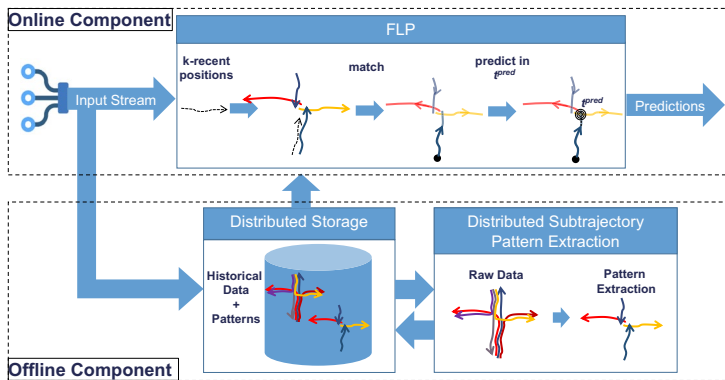


Fig. 4 Subtrajectory pattern-based FLP architecture

a moving object, given a look-ahead time and the set of patterns identified during the offline phase, as depicted in Fig. 4.

In more technical depth, the *offline* component consists of a distributed storage file system, which contains accumulated historical mobility data, and the *Distributed Subtrajectory Pattern Extraction* module, which takes as input a distributed trajectory dataset from the distributed file system and identifies a set of subtrajectory pattern. A predictor is built for each moving object by taking into account its individual past movement. Regarding the *online* component, it receives as input streams of mobility data, concerning the recent positions of moving objects and, for each moving object the corresponding subtrajectory pattern is retrieved, its k -most recent positions are matched with the most similar pattern, and the object's future location up to the given look-ahead time Δt is predicted.

4.3 Evolving clusters discovery

After receiving the predicted locations for each moving object, we use *EvolvingClusters* [35] in order to shape the predicted co-movement patterns. Because the sampling rate

may vary for each moving object, we use linear interpolation to temporally align the predicted locations at a common timeslice with a stable sampling (alignment) rate sr .

Given a timeslice T_{now} , EvolvingClusters works in a nutshell, as follows:

- Calculates the pairwise distance for each object within T_{now} , and drop the locations with distance less than θ ;
- Creates a graph based on the filtered locations, and extract its Maximal Connected Subgraphs (MCS) and Cliques (MC) with respect to c ;
- Maintains the currently active (and inactive) clusters, given the MCS and MC of T_{now} and the recent (active) pattern history; and
- Outputs the eligible active patterns with respect to c, t and θ .

The output of EvolvingClusters, and by extension of the whole predictive model, is a tuple of four elements (o_{ids}, t_s, t_e, tp) , where o_{ids} is the set of objects that form an evolving cluster, t_s and t_e , its corresponding start and ending timestamp, respectively, with tp noting its type.

For instance, the final output of the model for the example of Fig. 1 is a set of 4-element tuples, i.e., $\{(P_1, T_1, T_5, 2), (P_2, T_1, T_5, 1), (P_3, T_1, T_4, 1), (P_3, T_1, T_5, 2), (P_4, T_1, T_5, 1)\} \cup \{(P_5, T_4, T_5, 1), (P_6, T_1, T_5, 1)\}$. In other words, we predict that:

- P_1, P_2, P_3, P_4 will survive intact,
- P_3 will become inactive at timeslice T_4 , but it will remain active as an MCS at timeslice T_5 , and,
- two new clusters P_5, P_6 will be discovered at timeslice T_5

4.4 Cluster evolution

While the prediction of the course of a cluster is an important task, of equal importance is the tracking of the clusters' transitions, so as to derive some more salient conclusions regarding the clusters' movement.

In order to track the clusters' evolution through time, we use the MONIC/FINGER-PRINT framework [22, 27], which proposes a taxonomy of cluster transitions, differentiating between external and internal transitions, that concern the relationship of a cluster to other clusters, and itself, respectively. In contrast to other graph-based methods such as the Reeb graph [3] that are intended (mainly) for offline processing, MONIC can be used within online workflows, therefore rendering it a good option for predicting the clusters' evolution through time in real time fashion.

As already discussed in Section 3, given two clusters C and C' from timeslices T_i and T_{i+1} , respectively, the transitions that a cluster might encompass are among the following: survival, absorption, split, disappearance, and emergence.

To measure the similarity between C and C' , the authors in [22, 27] use the following formula, which in a nutshell, is a variant of the well-known Jaccard index, a metric which we will use extensively in the following sections.

$$\text{MONIC}_{sim}(C, C') = \frac{|C \cap C'|}{|C|}$$

In our running example (c.f. Fig. 1), the expected output of MONIC (for the discovered MCS) is that clusters P_1 , and P_4 emerge at T_1 and survive up to T_3 . From that point, after we use our FLP model, we predict that cluster P_1 will survive up to T_5 , cluster P_6 will emerge at T_5 , and cluster P_4 will be absorbed by cluster P_5 at T_4 , which in turn will survive up to T_5 .

4.5 Evaluation measures

The evaluation of a co-movement pattern prediction approach is not a straightforward task, since we need to define how the error between the predicted and the actual clusters will be quantified. Intuitively, our aim is to match each predicted cluster C_{pred} with the most similar actual one cluster C_{act} . Towards this direction, we adopt, and then combine, three similarity measures, namely, spatial, temporal, and membership similarity. Concerning the spatial similarity, it is defined as follows:

$$Sim^{spatial}(C_{pred}, C_{act}) = \frac{MBR(C_{pred}) \cap MBR(C_{act})}{MBR(C_{pred}) \cup MBR(C_{act})} \quad (5)$$

where $MBR(C_{pred})$, ($MBR(C_{act})$) is the spatial coverage of the predicted cluster (actual cluster, respectively), in other words, the Minimum Bounding Rectangle (MBR) covering the objects' location in the given timeslice.

Regarding the temporal dimension, we adopt Allen's interval algebra [2] and calculate the temporal similarity as follows:

$$Sim^{temp}(C_{pred}, C_{act}) = \frac{Interval(C_{pred}) \cap Interval(C_{act})}{Interval(C_{pred}) \cup Interval(C_{act})} \quad (6)$$

where $Interval(C_{pred})$, ($Interval(C_{act})$) is the time interval when the predicted cluster (actual cluster, respectively) was valid.

for the membership similarity, we adopt the Jaccard similarity:

$$Sim^{member}(C_{pred}, C_{act}) = \frac{|C_{pred} \cap C_{act}|}{|C_{pred} \cup C_{act}|} \quad (7)$$

Finally, we define the overall similarity between two clusters as:

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

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $\lambda_i \in (0, 1)$, $i \in \{1, 2, 3\}$.

This further implies that a predicted cluster should be correctly matched with the corresponding actual cluster, which is not a straightforward procedure. Our methodology for matching each predicted cluster C_{pred} with the corresponding actual one C_{act} is depicted in (9).

$$Sim(EC_{pred}, EC_{act}) = \left\{ \max_{C_{pred} \in EC_{pred}} Sim^*(C_{pred}, C_{act}); C_{act} \in EC_{act} \right\} \quad (9)$$

In other words, we compare each C_{pred} with every C_{act} and “match” it with the most similar one. After all predicted clusters get traversed, we end up with the set $Sim(EC_{pred}, EC_{act})$ which holds all the “matchings”.

5 Experimental study

In this section, we evaluate our predictive model on two different mobility domains, namely maritime and urban transportation, and present our experimental results.

5.1 Datasets & preprocessing

From the maritime domain, we use the “Piraeus”¹ dataset [34], which consists of 103 million AIS positioning messages from 5,163 vessels (passenger boats, fisheries, cargo, containers, etc) in the wider region of Piraeus, Greece. The dataset ranges in time and space, as follows:

- temporal range: January 1st, 2018 – December 31st, 2018 (1 year);
- spatial range: longitude in [22.992, 24.031]; latitude in [37.437, 38.046].

During the preprocessing stage, we drop erroneous records (i.e. GPS locations) based on a speed threshold $speed_{max}$ as well as stop points (by removing records corresponding to speed less than 1 knot [7]). Although it may sound optional, the latter is a critical cleansing process in order for the ML model to avoid considering non-evolving parts of the trajectories; afterwards we organize the cleansed data into trajectories based on the temporal interval between two consecutive signals of the same vessel, given a threshold dt . Finally, in order to discover evolving clusters, we need a stable and temporally aligned sampling rate. For the aforementioned dataset, we set the following thresholds: $speed_{max} = 50$ knots, $dt = 30$ min., and $rate = 1$ min. The rationale behind these thresholds stems from the characteristics of the dataset, which were unveiled after a statistical analysis of the distribution of the $speed$ and dt between consecutive points of the same trajectory.

From the urban domain, we use the popular “GeoLife”² dataset [40–42], which contains information regarding urban movement (mainly) in Beijing, China. In particular, it consists of approximately 23.5 million records from 181 users organized in 18,454 trips, with some of their trips being semantically annotated with respect to the means of transportation they used at that time. The dataset ranges in time and space as follows:

- temporal range: April 12th, 2007 – July 27th, 2012 (approx. 5 years);
- spatial range: longitude in [73.50, 134.78], latitude in [17.96, 53.56] (restricted to China).

¹ The dataset is publicly available at <https://zenodo.org/record/4498410>

² The dataset is publicly available at <https://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/>



Fig. 5 Snapshots of Piraeus (left) and GeoLife (right) datasets

Regarding the GeoLife dataset, we omit the preprocessing step and accept it ‘as-is’. The main reason for this, is due to the fact that the raw points are already organized into trips and are neatly labeled with respect to the vehicle used. That being said, the only preprocessing we performed was to restrict its spatial range to the China region.

A map visualization of (parts of) the two datasets is illustrated in Fig. 5. In particular, Fig. 5 (left) visualizes Piraeus’ locations on July 10th, 2018, while Fig. 5 (right) visualizes GeoLife locations on October 20–22nd, 2011.

5.2 Experimental setup and results

The objectives of our experimental study are related to:

- the efficiency of our framework in terms of run time;
- FLP prediction quality;
- its effect on EC prediction quality;
- the sensitivity of (8) with respect to λ_i ; and
- quality assessment of the clusters’ predicted evolution.

The GRU-FLP and EC algorithms were implemented in Python3 (via Anaconda virtual environments), while the ST-FLP algorithm was implemented in Java. The experiments were conducted using Apache Kafka with 1 topic for the transmitted (loaded from a CSV file) and predicted locations, as well as 1 consumer for FLP and EC, respectively.

For the EC-related experiments, we utilized a single node with 8 CPU cores, 16 GB of RAM and 256 GB of HDD. For ST-FLP, we employed a 49 node Hadoop 2.7.2 cluster, where the master node consists of 8 CPU cores, 8 GB of RAM and 60 GB of HDD while each slave node is comprised of 4 CPU cores, 4 GB of RAM and 60 GB of HDD. The offline component was implemented over Apache Hadoop and the online component by utilizing Apache Kafka for messaging and Kafka Consumers interface for stream processing. All of the above are provided by okeanos-knossos,³ an IAAS service for the Greek Research and Academic Community. Finally, for the GRU-FLP, we used a single node equipped with 16 CPU cores, 64 GB of RAM, 1 TB of HDD, and an Nvidia GTX 2080Ti.

³ <https://okeanos-knossos.grnet.gr/home/>

Table 1 Datasets’ threshold values for Evolving Cluster (EC) Discovery

	Piraeus	GeoLife
Cardinality (c – objects)	5	5
Temporal (t – minutes)	15	10
Distance (d – meters)	1500	30

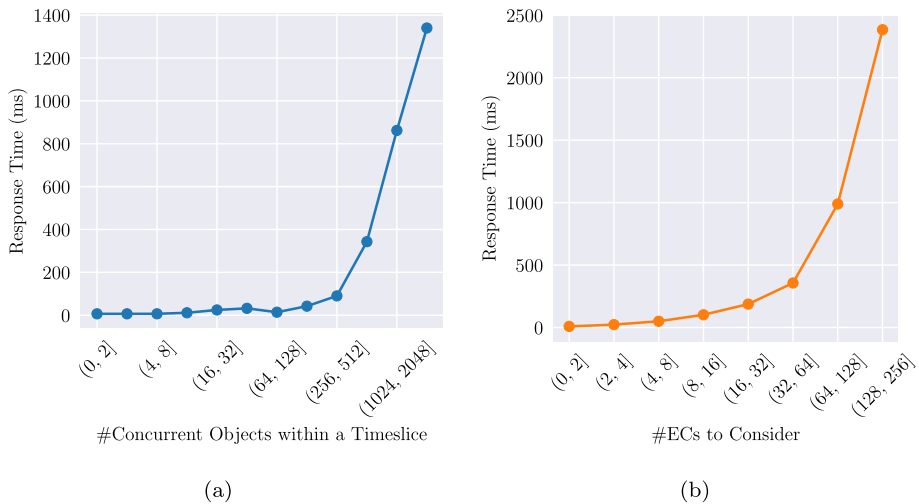


Fig. 6 Average time elapsed on (a) EC vs. #concurrent objects within a timeslice; and (b) CE vs. #ECs to consider

In the sections that follow, we provide the experimental results of our study, using the default parameters of EvolvingClusters per dataset, as defined by [35], as Table 1 illustrates. The respective source-code is available at <https://github.com/DataStories-UniPi/EvolvingClusters>, as a branch of the general Evolving Clusters -related library of operations.

5.2.1 Efficiency of our framework towards online processing

It is shown that EC, GRU-FLP, and ST-FLP algorithms are able to run in online mode in all corresponding real-world situations that are experimented with [7, 30, 35], with response time ≈ 250 ms for predicting the vessels’ locations at a single temporal instance. However, when combined, the time complexity of the proposed framework is overwhelmed by the most ‘expensive’ steps, namely EC and CE.

Nevertheless, according to our experiments, our framework is able to run in online mode in all experimental real-world scenarios. In detail, focusing on the entire temporal horizon of both datasets, and running our framework with an activation step of 60 s, Fig. 6 illustrates the (average recorded) response time for its most “expensive” steps, namely, EC and CE.

As it appears in the chart, the response time for EC (c.f. Fig. 6a) remains below 400 ms for thousands of concurrent objects and increases up to (approx.) 1400 ms for 3,527 objects, which corresponds to the most crowded timeslice in our experimental study.

Similar results are also recorded for CE, albeit with slightly increased response, with up to 1000 ms when considering ≈ 128 clusters, and up to 2500 ms for ≈ 256 clusters.

Summarising, although it takes some time to train the underlying FLP models, the response time of our framework in its online part is typically below 4 sec per monitored timeslice, thus well-placing it within the online method family. Last but not least, it is obvious that the performance of our framework is directly correlated to the hardware of the experimental setup.

5.2.2 FLP prediction quality

In our method, predicting evolving clusters relies heavily on the FLP model. In this context, we compare two models, namely GRU-FLP (c.f. Section 4.1) and ST-FLP (c.f. Section 4.2), and assess their impact on EC discovery. Figure 7 illustrates the distribution of the predicted locations' Mean Absolute Error (MAE) versus the prediction lookahead Δt . We observe that as we increase Δt , the error increases rapidly for the former, going from hundreds of meters ($\Delta t = 1$) to tens of kilometers ($\Delta t = 10$). On the other hand, for the Sub-trajectory based model the error is well-maintained to at most 1,000 meters for the Piraeus

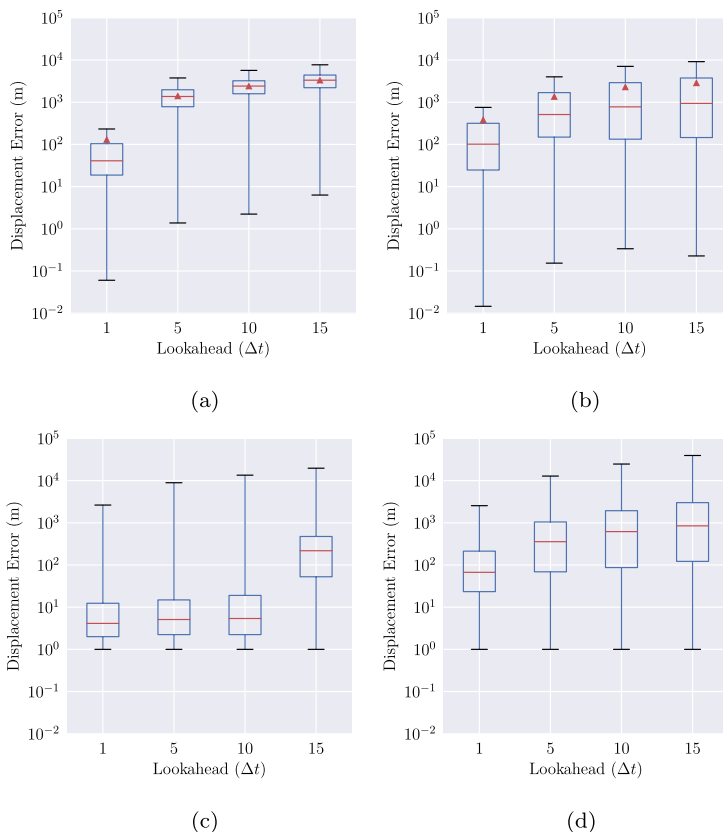


Fig. 7 Displacement error for GRU-FLP (top) and ST-FLP (bottom) models for Piraeus (a,c); and GeoLife (b,d) datasets, respectively

and GeoLife dataset, respectively, with slight increases due to increasing Δt , thus rendering the Sub-trajectory based model the better model for the datasets and task at hand.

5.2.3 EC prediction quality

Having evaluated the quality of our predictive models, we proceed to assess the quality of the predicted evolving clusters for our models on both mobility domains with respect to their corresponding “ground truth”. As “ground truth” we define the evolving clusters discovered using the objects’ corresponding actual positions, and for quantifying the accuracy of a predicted cluster C_{pred} against an actual cluster C_{act} we use (9), with $\lambda_i, i \in [1,3]$ set to their corresponding values. Without loss of generality, Fig. 8 illustrate the EC prediction quality of both predictive models over the Piraeus dataset, where solid lines correspond to the median accuracy, and the confidence intervals to the 25th and 75th quartile, respectively.

Observing the aforementioned figure, we clearly deduce that as we increase the look-ahead (Δt), ST-FLP consistently outperforms GRU-FLP (with median accuracy close to 70%) in all aspects of cluster matching, except for sim^{member} , where GRU-FLP outperforms ST-FLP by a significant margin. More specifically, the performance of ST-FLP on both sim^{temp} and $sim^{spatial}$ show similar behaviour, decreasing with similar rate as Δt increases, a behaviour which is seemingly correlated to the FLP models’ performance.

Because, in general, both datasets are quite challenging, as they contain a higher degree of unpredictability (due to the amount of manouvers each trajectory has), we observe that

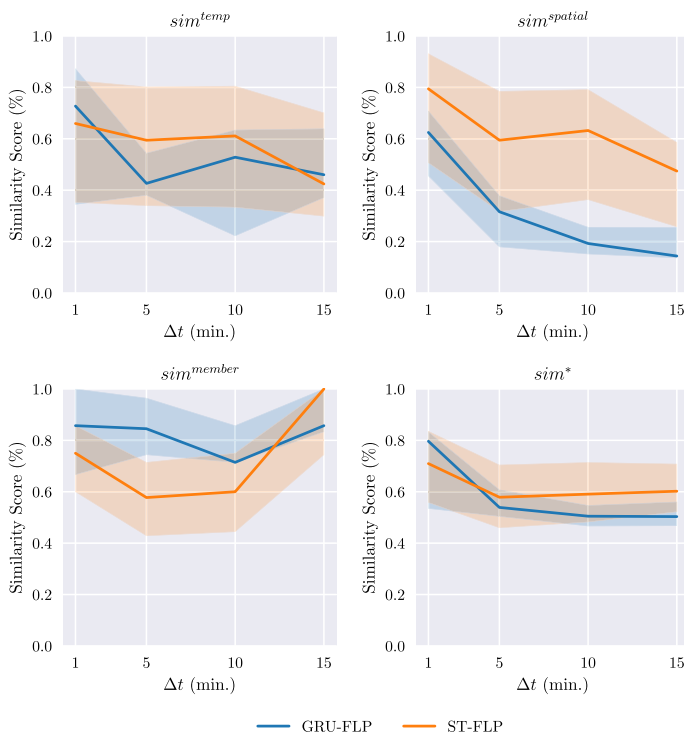


Fig. 8 EC prediction quality – assessing models’ performance on Piraeus dataset

both GRU-FLP and ST-FLP models have quite large confidence interval ranges, indicating that there is room for further tuning.

5.2.4 EC similarity score sensitivity

One of the main claimed contributions in our work is the proposed EC similarity score (c.f. (8)) for matching the predicted ECs with the actual ones, and this measure takes the form of a convex combination (weights $\lambda_1, \lambda_2, \lambda_3, \sum_i \lambda_i = 1$) of spatial, temporal, and membership similarities, respectively. The values of λ_i are fixed in their corresponding default values (i.e. $\lambda_i = 1/3$) by taking into account all matching aspects (namely, temporal, spatial, and membership) in a balanced manner (i.e., average).

To assess the effect of these weights, Fig. 9(a–c) illustrates the distribution of EC similarity score, on the Piraeus dataset ($\Delta t = 1$ min.) with respect to (a) λ_2, λ_3 , (b) λ_1, λ_3 , and (c) λ_1, λ_2 . Observing the aforementioned figure, we deduce that the temporal aspect (λ_1) is the most dominant, as all other aspects (namely, spatial and membership) have minor impact on the final score, while $\lambda_1 = 1/3$.

This behaviour is expected, as the predicted co-movement pattern may demonstrate a temporal and/or membership overlap with its actual one, but, due to the models' displacement error, exhibit minor spatial overlap, an increasing behaviour, especially for higher Δt , e.g., 15 min. On the other hand, fixing λ_2 , and λ_3 , we observe that as we increase the value of λ_1 , the similarity score decreases with an - almost - steady

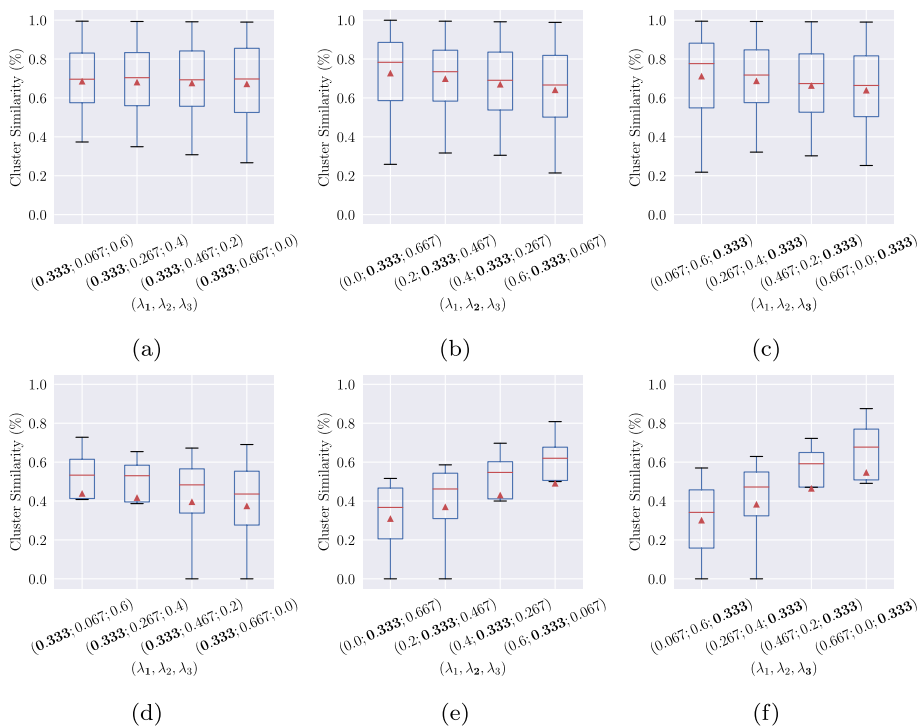


Fig. 9 Assessing the sensitivity of EC similarity (c.f. (8)) on Piraeus (a,b,c) and Geolife (d,e,f) datasets, with respect to (a,d) λ_2, λ_3 , (b,e) λ_1, λ_3 , and (c,f) λ_1, λ_2

rate, further showing the trade-off of focusing more on a single similarity aspect over another. Similar observations can be observed on the GeoLife dataset (c.f. Fig. 9(d-f)), albeit with steeper increase/decrease rates, as we increase λ_2 and λ_3 , respectively.

5.2.5 Cluster evolution accuracy

After evaluating the EC prediction within the spatiotemporal dimension, we may assess the prediction accuracy regarding the clusters' evolution. As defined in Section 4.4, a cluster C will emerge, survive for some time duration Δt , and then disappear, or be absorbed into another cluster C' or split into n clusters $C_1, C_2, \dots, C_n$.

Exploiting on the MONIC/FINGERPRINT framework [22, 27], we create a graph $G = (V, E)$, with its nodes corresponding to the aforementioned ECs ($\langle P_i, T_i, T_j, tp \rangle$ $j > i$) and its edges corresponding to the event (i.e., emerged, survived, absorbed, split, disappeared) which occurred between these two clusters. For instance, consider the setting that appears in Fig. 1 for $tp = 2$ (MCS); the respective graph G of cluster evolutions is illustrated in Fig. 10a. This graph represents the actual situation of cluster evolution; hence it is G_{act} . On the other hand, let us assume that our framework predicted a different situation; more specifically, at T_5 all clusters were combined into a single entity $P_7 = \langle \{a, b, c, d, e, f, g, h, i\}, T_5, T_5, 2 \rangle$, which is represented in G_{pred} , as illustrated in Fig. 10b.

In order to compare the evolution over time between a G_{act} and G_{pred} , we exploit on the Graph Edit Distance (GED) measure. In particular, GED is a graph similarity measure analogous to Levenshtein distance for strings, defined as the minimum cost of edit path (sequence of node and edge edit operations) transforming a graph G_1 to another graph G_2 , such that G_1 and G_2 are isomorphic [1]. Because the resulted graphs may significantly vary in their corresponding size, thus resulting in confusion regarding the final score, we use the normalized GED (nGED – c.f. (10)) in order to bound the scores between zero and one, or in other words, between full and none isomorphism, respectively.

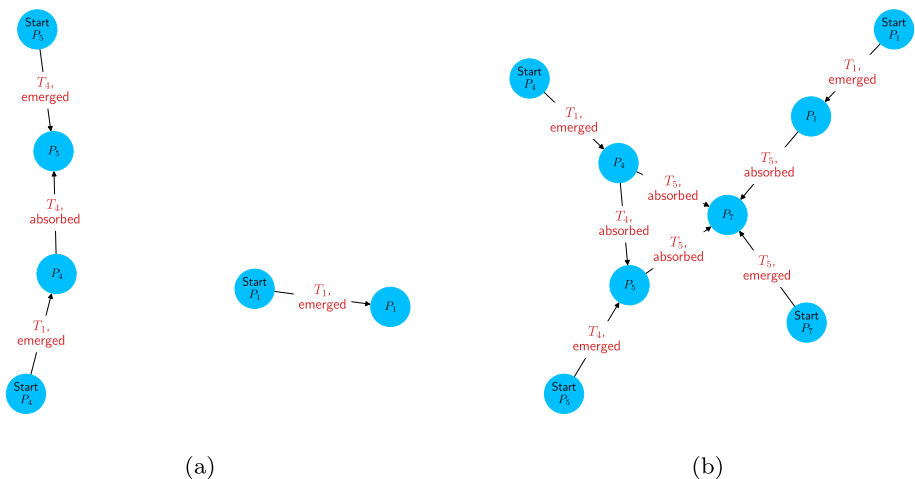


Fig. 10 Discovered ECs' evolution graph on (a) actual (C_{act}); and (b) predicted (C_{pred}) situation, respectively

$$nGED = \frac{GED(G_{act}, G_{pred})}{2 * \max(|V_{act}| + |E_{act}|, |V_{pred}| + |E_{pred}|)}, \quad (10)$$

where G_{actual} , G_{pred} the evolution graphs of C_{act} and C_{pred} with V , and E , their corresponding vertices and edges, respectively.

After comparing the evolution graphs for each predicted EC with its corresponding actual one, Table 2 illustrates the minimum, median, and maximum nGED, respectively. For the Piraeus dataset, we can clearly observe that in both FLP models, as Δt increases so does the nGED, albeit with some minor noise, between the actual and predicted EC evolution graphs.

Moreover, for the GeoLife dataset, we observe similar behaviour but with less isomorphic graphs, a result correlated with the models' performance over the aforementioned datasets. These conclusions are in accordance with the ones derived from Fig. 8, further strengthening the (intuitively obvious) correlation between FLP accuracy and EC prediction, even at higher levels of abstraction (i.e., cluster evolution).

5.3 Towards exploiting the predicted clusters

A very important stage in the data analytics hierarchy is data storytelling; in other words, how to deduce meaningful and easy to interpret outcomes from a (curated) dataset at hand. After executing the EC module on both datasets for the actual and predicted locations, respectively, we present some further findings, aiming to reach some insightful conclusions regarding the objects' future activity.

Towards this direction, we construct heatmaps of the discovered ECs using the cardinality of each cluster as the measurable magnitude and setting the parameters of EC to their corresponding default values (c.f. Table 1). In particular, Figures 9 and 10 illustrate two heatmaps of actual (left) and predicted ECs (right) in indicative regions of Piraeus dataset (cells of size 225x225 m) and GeoLife dataset (cells of size 256x256 m), respectively. In order to avoid repetitions, we choose to focus our discussion on $\Delta t = 1$.

Starting with the Piraeus dataset, if we are interested in predicting the vessels' port entry/exit routes, Fig. 11 (top) provides an answer: it can be observed that our GRU-based model predicted that most traffic will be concentrated at the ferry lines connecting the ports at peak hours, with emphasis on the ferry line connecting Salamina with Perama. Even further, focusing on the ST-FLP model, we not only observe the same behaviour (albeit

Table 2 Evaluating ECs' evolution over time (via nGED), using GRU-FLP vs. ST-FLP models on Piraeus and GeoLife datasets (lower is better)

Δt	GRU-FLP		ST-FLP	
	Piraeus Dataset	Geolife Dataset	Piraeus Dataset	Geolife Dataset
	(min.; med.; max.)	(min.; med.; max.)	(min.; med.; max.)	(min.; med.; max.)
1	0; 0; 0.444	0; 0.684; 0.864	0; 0.333; 0.519	0.333; 0.588; 0.656
5	0; 0.333; 0.477	0; 0.532; 0.753	0; 0.473; 0.753	0; 0.493; 0.587
10	0; 0.396; 0.481	0.45; 0.538; 0.695	0; 0.464; 0.710	0; 0.532; 0.694
15	0; 0.357; 0.478	0.381; 0.517; 0.650	0; 0.400; 0.516	0; 0.250; 0.500

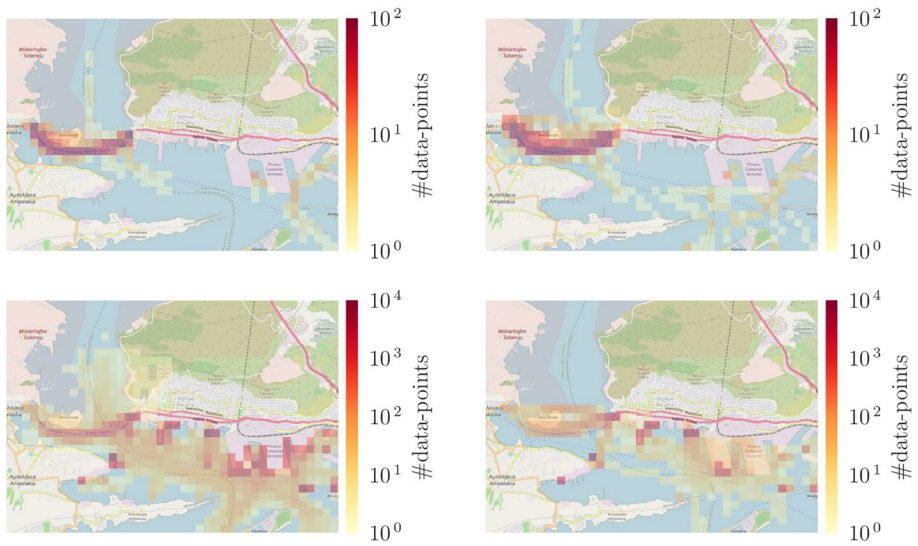


Fig. 11 Heatmap of ECs in a region of the Piraeus dataset using GRU-FLP (top) and ST-FLP (bottom) models (left: actual; right: predicted)

with higher resolution), but also we can accurately predict the intra-port activity, a conclusion that further empowers our confidence towards a predictive framework able to correctly anticipate incoming vessels, thus effectively managing port traffic. Except real-life

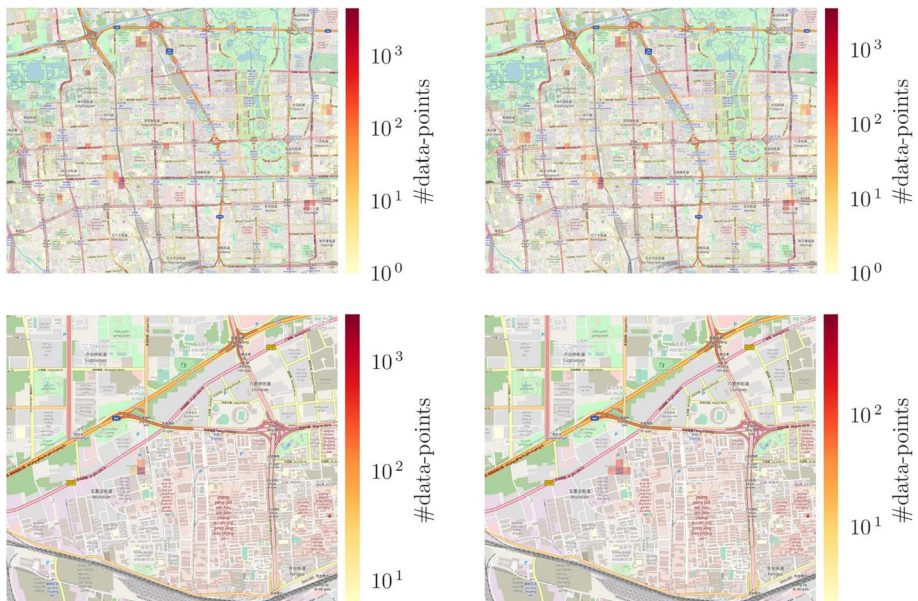


Fig. 12 Heatmap of ECs in a region of the GeoLife dataset using GRU-FLP (top) and ST-FLP (bottom) models (left: actual; right: predicted)

applications, the aforementioned findings may trigger domain experts into further investigating these occurrences and reach some meaningful conclusions.

Moving to the urban domain and regarding the GeoLife dataset, as Fig. 12 illustrates, most traffic lies around some places of interest (such as parks, shops, and university campuses, as it can be easily deduced when comparing the spatial footprint of the discovered evolving clusters with open spatial data sources, like Wikimapia.org, LinkedGeoData.org, etc.). More specifically, focusing on the northern part of Beijing, the GRU-FLP model predicts not only moderate traffic at the aforementioned areas, but also mild traffic on surrounding roads at peak hours. Similar results are also obtained at the south-western side of Beijing using the ST-FLP model. In general, combined with the hours they are created, a domain expert (e.g. transportation analyst) could easily determine where and when a congestion will occur, thus using our predictive models as a utility to effectively reroute urban traffic.

6 Conclusions and future work

In this paper, we propose a unified framework for the online prediction of evolving clusters and their evolution over time. The proposed method is based on a combination of FLP and EC algorithms and is evaluated through two real-world datasets from the maritime and urban domain, respectively, taking into account a novel co-movement pattern similarity measure, which is able to match the predicted clusters with the actual ones. Our experimental study on real-life datasets demonstrates the efficiency and effectiveness of the proposed methodology. Thus, based on the potential applications, as well as the quality of the results produced, we believe that the proposed model can be a valuable utility for researchers and practitioners alike.

In the near future, we aim to further tune our models with respect to their respective architecture and/or parameters, and experiment on more datasets from other domains (e.g. sports analytics) in order to demonstrate the versatility of our framework.

Funding Open access funding provided by HEAL-Link Greece This work was partially supported by project VesselAI (grant agreement No 957237; <https://vessel-ai.eu>), which has received funding by the EU Horizon 2020 R&I Programme.

Data availability The authors declare that all data supporting the findings of this study are available within the article.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abu-Aisheh Z, Raveaux R, Ramel J, Martineau P (2015) An exact graph edit distance algorithm for solving pattern recognition problems. In: ICPRAM (1). SciTePress, pp 271–278
2. Allen JF (1983) Maintaining knowledge about temporal intervals. *Commun ACM* 26(11):832–843
3. Buchin K, Buchin M, van Kreveld MJ, Speckmann B, Staals F (2013) Trajectory grouping structure. In: WADS, Lecture Notes in Computer Science, vol 8037. Springer, pp 219–230
4. Chen L, Gao Y, Fang Z, Miao X, Jensen CS, Guo C (2019) Real-time distributed co-movement pattern detection on streaming trajectories. *Proc VLDB Endow* 12(10):1208–1220
5. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1724–1734
6. Chondrodima E, Georgiou H, Pelekis N, Theodoridis Y (2022) Particle swarm optimization and rbf neural networks for public transport arrival time prediction using gtfs data. *Int J Inf Manag Data Insights* 2(2):100086
7. Chondrodima E, Mandalis P, Pelekis N, Theodoridis Y (2022) Machine learning models for vessel route forecasting: An experimental comparison. In: Proceedings of the 23rd IEEE international conference on mobile data management (MDM)
8. Dong Y, Pinelli F, Gkoulas Y, Nabi Z, Calabrese F, Chawla NV (2015) Inferring unusual crowd events from mobile phone call detail records. In: Proceedings of the joint european conference on machine learning and knowledge discovery in databases (ECML-PKDD), pp 474–492
9. Fan Q, Zhang D, Wu H, Tan K (2016) A general and parallel platform for mining co-movement patterns over large-scale trajectories. *Proc VLDB Endow* 10(4):313–324
10. Fang Z, Gao Y, Pan L, Chen L, Miao X, Jensen CS (2020) CoMing: A real-time co-movement mining system for streaming trajectories. In: Proceedings of ACM SIGMOD/PODS international conference on management of data, pp 2777–2780
11. Georgiou HV, Pelekis N, Sideridis S, Scarlatti D, Theodoridis Y (2020) Semantic-aware aircraft trajectory prediction using flight plans. *Int J Data Sci Anal* 9(2):215–228
12. Hao Z, Huang X, Wang K, Cui M, Tian Y (2020) Attention-based GRU for driver intention recognition and vehicle trajectory prediction. In: Proceedings of the 4th CAA international conference on vehicular control and intelligence (CVCI), pp 86–91
13. Helmi S, Kashani FB (2020) Multiscale frequent co-movement pattern mining. In: Proceedings of the 36th IEEE international conference on data engineering (ICDE), pp 829–840
14. Jeung H, Yiu ML, Zhou X, Jensen CS, Shen HT (2008) Discovery of convoys in trajectory databases. *PVLDB* 1(1):1068–1080
15. Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatio-temporal data. In: Proceedings of the 17th international symposium on spatial and temporal databases SSTD, pp 364–381
16. Kannangara S, Xie H, Tanin E, Harwood A, Karunasekera S (2020) Tracking group movement in location based social networks. In: Proceedings of the international conference on advances in geographic information systems (SIGSPATIAL/GIS), pp 251–262
17. Laube P, Imfeld S, Weibel R (2005) Discovering relative motion patterns in groups of moving point objects. *Int J Geogr Inf Sci* 19(6):639–668
18. Li Y, Bailey J, Kulik L (2015) Efficient mining of platoon patterns in trajectory databases. *Data Knowl Eng* 100:167–187
19. Li Z, Ding B, Han J, Kays R (2010) Swarm: Mining relaxed temporal moving object clusters. *PVLDB* 3(1):723–734
20. Liu H, Wu H, Sun W, Lee I (2019) Spatio-temporal GRU for trajectory classification. In: Proceedings of IEEE international conference on data mining (ICDM), pp 1228–1233
21. Marz N (2011) How to beat the CAP theorem. In: thoughts from the red planet. <http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html>. Accessed 4 Jul 2022
22. Ntoutsi E, Spiliopoulou M, Theodoridis Y (2012) FINGERPRINT: summarizing cluster evolution in dynamic environments. *Int J Data Warehous Min* 8 (3):27–44
23. Papadimitrakis M, Stogiannos M, Sarimveis H, Alexandridis A (2021) Multi-ship control and collision avoidance using mpc and rbf-based trajectory predictions. *Sensors* 21(21):6959
24. Petrou P, Nikitopoulos P, Tampakis P, Glenis A, Koutroumanis N, Santipantakis GM, Patroumpas K, Vlachou A, Georgiou HV, Chondrodima E, Doukeridis C, Pelekis N, Andrienko GL, Patterson F, Fuchs G, Theodoridis Y, Vouras GA (2019) ARGO: A big data framework for online trajectory prediction. In: Proceedings of the 16th international symposium on spatial and temporal databases (SSTD), pp 194–197

25. Petrou P, Tampakis P, Georgiou HV, Pelekis N, Theodoridis Y (2019) Online long-term trajectory prediction based on mined route patterns. In: Proceedings of the Multiple-Aspect Analysis of Semantic Trajectories - First International Workshop, MASTER 2019, Held in Conjunction with ECML-PKDD, pp 34–49
26. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323:533–536
27. Spiliopoulou M, Ntoutsi I, Theodoridis Y, Schult R (2006) MONIC: modeling and monitoring cluster transitions. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp 706–711
28. Stogiannos M, Papadimitrakakis M, Sarimveis H, Alexandridis A (2021) Vessel trajectory prediction using radial basis function neural networks. In: IEEE EUROCON 2021 - 19th international conference on smart technologies, pp 113–118
29. Suo Y, Chen W, Claramunt C, Yang S (2020) A ship trajectory prediction framework based on a recurrent neural network. *Sensors* 20(18)
30. Tampakis P, Chondrodima E, Tritsarolis A, Pikrakis A, Theodoridis Y, Pristouris K, Nakos H, Kalampokis P (2022) Dalamagas, T.: i4sea: a big data platform for sea area monitoring and analysis of fishing vessels activity. *Geo-spatial Inf Sci* 25(2):132–154
31. Tampakis P, Pelekis N, Doukeridis C, Theodoridis Y (2019) Scalable distributed subtrajectory clustering. In: Proceedings of 2019 IEEE international conference on big data (Big Data), pp 950–959
32. Tang LA, Zheng Y, Yuan J, Han J, Leung A, Hung C, Peng W (2012) On discovery of traveling companions from streaming trajectories. In: Proceedings of the 28th IEEE international conference on data engineering (ICDE), pp 186–197
33. Trasarti R, Guidotti R, Monreale A, Giannotti F (2017) MyWay: Location prediction via mobility profiling. *Inf Syst* 64:350–367
34. Tritsarolis A, Kontoulis Y, Theodoridis Y (2021) The piraeus ais dataset for large-scale maritime data analytics [data set]. <https://doi.org/10.5281/zenodo.5562629>
35. Tritsarolis A, Theodoropoulos GS, Theodoridis Y (2021) Online discovery of co-movement patterns in mobility data. *Int J Geogr Inf Sci* 35(4):819–845
36. Wang C, Ren H, Li H (2020) Vessel trajectory prediction based on ais data and bidirectional gru. In: Proceedings of international conference on computer vision, image and deep learning (CVIDL), pp 260–264
37. Werbos PJ (1990) Backpropagation through time: what it does and how to do it. *Proc IEEE* 78(10):1550–1560
38. Zhang Y, Zheng L (2021) Pedestrian trajectory prediction with MLP-Social-GRU. In: Proceedings of the 13th international conference on machine learning and computing (ICMLC), pp 368–372
39. Zheng K, Zheng Y, Yuan NJ, Shang S (2013) On discovery of gathering patterns from trajectories. In: Proceedings of the 29th IEEE international conference on data engineering (ICDE), pp 242–253
40. Zheng Y, Li Q, Chen Y, Xie X, Ma W (2008) Understanding mobility based on GPS data. In: Proceedings of the 10th international conference on ubiquitous computing (UbiComp), pp 312–321
41. Zheng Y, Xie X, Ma W (2010) Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng Bull* 33(2):32–39
42. Zheng Y, Zhang L, Xie X, Ma W (2009) Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web (WWW), pp 791–800

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Andreas Tritsarolis received his M.Sc. in Data Science and Machine Learning from the National Technical University of Athens (NTUA) in 2022, and his B.Sc. in Computer Science from the University of Piraeus in 2020. He is currently a junior researcher in the Data Science Lab at the University of Piraeus, collaborating in a number of research projects. He has co-authored publications in conference proceedings and journals, all in the topic of Mobility Data Analytics. He also received the SSTD 2021 Best Demo Paper Award for the paper entitled “MaSEC: Discovering Anchorages and Co-movement Patterns on Streaming Vessel Trajectories”. His research interests include Data Science for Mobility Data as well as Machine Learning, with a focus on Deep Learning Architectures and Federated Learning.

Eva Chondrodima received the Ph.D. degree in computational intelligence from the National Technical University of Athens (NTUA), Greece, in 2017, the M.Sc. degree in advanced information systems from the University of Piraeus, Greece, in 2013, and the B.Sc. degree in electronic engineering from the

Technological Educational Institute of Athens, Greece, in 2011. She is currently a Post-Doctoral researcher with the Department of Informatics, University of Piraeus. Her research interests include computational intelligence with emphasis on neural networks and evolutionary computation methods, and applications to geoscience, mobility data science, and semantic Web.

Panagiotis Tampakis is currently an Assistant Professor at the Department of Mathematics and Computer Science of the University of Southern Denmark. He has participated in a number of research projects related to Data Science and Big Data, with EU Horizon 2020 funded MASTER (2018–21) and Track-and-Know (2018–20) being the ongoing ones. His publication record includes publications in top venues and journals, such as ICDE, EDBT, PKDD and DMKD, and has been awarded with the SSTD 2021, Ralf H. Güting Best Paper Award for the paper entitled “A Novel Indexing Method for Spatial-Keyword Range Queries”. His research interests include mobility data management and mining, big data, and distributed data management.

Aggelos Pikrakis received the Diploma Engineering degree from the University of Patras, in 1993, and the Ph.D. degree in Computer Science from the University of Athens, in 2002. He is currently an Assistant Professor at the Department of Informatics, University of Piraeus, Greece, teaching courses related to machine learning and audio processing. He is the co-inventor of AI/ML patents and he has coauthored two international textbooks in the English language and more than 50 refereed papers in international peer-reviewed scientific journals/conferences. His research interests include audio analysis, with an emphasis on machine learning algorithms, including deep neural networks, hidden Markov models, and Bayesian architectures and sequence alignment methods. His research work has received award recognition and he was a recipient of the 2019 EURASIP Meritorious Service Award.

Yannis Theodoridis is Professor of Data Science with the Department of Informatics, University of Piraeus, Greece. He has participated on several boards, including the editorial board of ACM Computing Surveys (2016–19) and the Endowment of the Symposium on Spatial and Temporal Databases - SSTD (2010–). He has also served as general co-chair for SSTD’03 and ECML/PKDD’11, PC vice-chair for IEEE ICDM’08, and PC member for several conferences, including ACM SIGMOD/PODS, IEEE ICDE, ACM SIGKDD, IEEE ICDM, etc. Since 2001, he has been (co-) principal investigator in several research projects funded by various government agencies (EU, Greece) through open calls. His research interests include Data Science (big data management & analytics) for human mobility-related information. He has co-authored three monographs and over 100 refereed articles in scientific journals and conferences, with over 13,000 citations ($h=52$), according to Google Scholar.