Online Co-movement Pattern Prediction in Mobility Data

Andreas Tritsarolis
Data Science Lab., Department of Informatics,
University of Piraeus
Piraeus, Greece
andrewt@unipi.gr

Eva Chondrodima
Data Science Lab., Department of Informatics,
University of Piraeus
Piraeus, Greece
evachon@unipi.gr

Panagiotis Tampakis
Data Science Lab., Department of Informatics,
University of Piraeus
Piraeus, Greece
ptampak@unipi.gr

Aggelos Pikrakis
Data Science Lab., Department of Informatics,
University of Piraeus
Piraeus, Greece
pikrakis@unipi.gr

ABSTRACT

Predictive analytics over mobility data are of great importance since they can assist an analyst to predict events, such as collisions, encounters, traffic jams, etc. A typical example of such analytics is future 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 being able to accurately predict collective behavioural patterns of movement, such as co-movement patterns. In this paper, we provide an accurate solution to the problem of *Online Prediction of Co-movement Patterns*. In more detail, we split the original problem into two sub-problems, namely *Future Location Prediction* and *Evolving Cluster Detection*. Furthermore, in order to be able to calculate the accuracy of our solution, we propose a co-movement pattern similarity measure, which facilitates the matching of the predicted clusters with the actual ones. Finally, the accuracy of our solution is demonstrated experimentally over a real dataset from the maritime domain.

KEYWORDS

Machine Learning, Predictive Analytics, Co-movement Patterns, Trajectory Prediction

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 are 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 [2, 5, 6, 8, 33]. 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 [23, 24, 27, 32], 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 and more specifically *Online Prediction of Co-movement Patterns*, has not been addressed in the relevant literature yet.

Concerning the definition of co-movement patterns, there are several approaches in the literature, such as [2, 5, 6, 8]. However, all of the above are either offline and/or operate at predefined temporal snapshots that imply temporal alignment and uniform sampling, which is not realistic assumptions. For this reason, we adopt the approach presented in [33], 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 [33] 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 to identifying *Maximal Cliques* (MCs) (for spherical clusters) or *Maximal Connected Subgraphs* (MCSs) (for density-connected clusters). Figure 1 illustrates such an example, where in blue we have the historical evolving clusters and in orange the predicted future ones.

Several mobility-related applications could benefit from such an operation. In the urban traffic domain, predicting co-movement patterns could assist in detecting future traffic jams which in turn can help the authorities take the appropriate measures (e.g. adjusting traffic lights) in order to avoid 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. Finally, in large epidemic crisis, contact tracing is one of the tools to identify individuals that have been close to infected persons for some time duration. Being able to predict these groups can help avoid future contacts with possibly infected individuals.

The problem of predicting the spatial properties of group patters has only been recently studied [12]. In more detail, the authors in [12] 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, this approach is offline and cannot be applied in an online scenario. Furthermore, the group definition adopted in [12] is rather limited, since the identify only spherical groups, as opposed to [33] where both spherical and density-connected clusters can be identified. Finally, the authors in [12] predict only the centroids of the clusters and not the shape and the membership of each cluster.

Inspired by the above, the problem that we address in this paper is the *Online Prediction of Co-movement Patterns*. Informally,

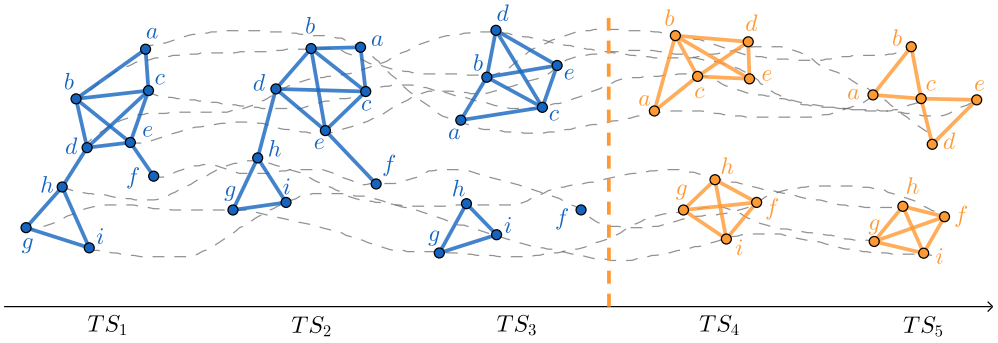

Figure 1: Predicting evolving clusters via (singular) trajectory prediction

given a look-ahead time interval Δt, the goal is to predict the groups, i.e. their spatial shape (spherical or density-connected), temporal coverage and membership, after Δt time. In more detail, we split the original problem into two sub-problems, namely *Future Location Prediction* and *Evolving Cluster Detection*. The problem of *Online Prediction of Co-movement Patterns* 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 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. To the best of our knowledge, the problem of *Online Prediction of Co-movement Patterns*, has not been addressed in the literature yet. Our main contributions are the following:

- We provide an accurate 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 perform an experimental study with a real dataset from the maritime domain, which verifies the accuracy of our proposed methodology.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, we formally define the problem of *Online Prediction of Co-movement Patterns*. Subsequently, in Section 4 we propose our two-step methodology and in Section 5, we introduce a co-movement pattern similarity measure along with the cluster "matching" algorithm. Section 6, presents our experimental findings and, finally, in Section 7 we conclude the paper and discuss future extensions.

2 RELATED WORK

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

Co-movement patterns. One of the first approaches for identifying such collective mobility behaviour is the so-called flock pattern [14], 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 [11], convoys [10], swarms [16], platoons [15], traveling companion [30] and gathering pattern [38]. 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 [5]. In more detail, the general co-movement pattern (GCMP), is proposed, which includes *Temporal Replication* and *Parallel Mining*, a method that, as suggested by its name, splits a data snapshot spatially and replicates data when necessary to ensure full coverage, and *Star Partitioning* and *ApRiori Enumerator*, a technique that uses graph pruning in order to avoid the data replication that takes place in the previous method. In [8], 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 [2, 6], propose a two-phase online distributed co-movement pattern detection framework, which includes the clustering and the pattern enumeration phase, respectively. During the clustering phase for timestamp t_s, the snapshot S_t is clustered using Range-Join and DBSCAN.

Another line of research, tries to discover groups of either entire or portions of trajectories considering their routes. There are several approaches whose goal is to group whole trajectories, including T-OPTICS [18, 19], that incorporates a trajectory similarity function into the OPTICS algorithm. However, discovering clusters of complete trajectories can overlook significant patterns that might exist only for portions of their lifespan. To deal with this, another line of research has emerged, that of *Subtrajectory Clustering*[20, 21, 28, 29], where the goal is to partition a trajectory into subtrajectories, whenever the density or the composition and its neighbourhood changes "significantly", then form groups of similar ones, while, at the same time, separate the ones that fit into no group, called outliers.

Another perspective into co-movement pattern discovery, is to reduce cluster types into graph properties and view them as such. In [31, 33], the authors propose a novel co-movement pattern definition, called *evolving clusters*, 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 using Apache Kafka®, without assuming temporal alignment, in constrast to the seminal works (i.e. flocks, convoys).

In the proposed predictive model, we will use the definition of *evolving clusters* [33] for co-movement pattern discovery. The

reason why is this 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 extensivelly 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 [32], 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 [23, 24] utilize the work done by [29] 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.

Recurrent Neural Network (RNN) -based models [26] constitute a popular method for trajectory prediction due to their powerful ability to fit complex functions, along with their ability of adjusting the dynamic behaviour as well as capturing the causality relationships across sequences. However, research in the maritime domain is limited regarding vessel trajectory prediction and Gated Recurrent Units (GRU) [3] models, which constitute the newer generation of RNN.

Suo et.al. [27] presented a GRU model to predict vessel trajectories based on a) the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to derive main trajectories and, b) a symmetric segmented-path distance approach to eliminate the influence of a large number of redundant data and to optimize incoming trajectories. Ground truth data from AIS raw data in the port of Zhangzhou, China were used to train and verify the validity of the proposed model.

Liu et.al. [17] 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. [34] aiming at predicting the movement trend of vessels in the crowded port water of Tianjin port, proposed a vessel berthing trajectory prediction model based on bidirectional GRU (Bi-GRU) and cubic spline interpolation.

Co-movement pattern prediction. The most similar work to ours has only been recently presented in [12]. More specifically, the authors in [12], 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, this approach is offline and cannot be applied in an online scenario. Furthermore, the group definition adopted in [12] is rather limited, since the identify only spherical groups, as opposed to [33] where both spherical and density-connected clusters can be identified. Finally, the authors in [12] predict only the centroids of the clusters and not the shape and the membership of each cluster.

3 PROBLEM DEFINITION

As already mentioned, we divide the problem into two sub-problems, namely *Future Location Prediction* and *Evolving Clusters Detection*. Before proceeding to the actual formulation of the problem, let us provide some preliminary definitions.

Definition 3.1. (Trajectory) A trajectory $T = \{p_1, \dots p_n\}$ 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 $1 \leq i \leq n$.

Definition 3.2. (Future Location Prediction). Given an input dataset $D = \{T_1, \dots, T_{|D|}\}$ 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 $t^i_{pred} = t^i_n + \Delta t$.

An informal definition regarding *group patterns* could be: "a large enough number of objects moving close enough to each other, in space and time, for some time duration". As already mentioned, in this paper we adopt the definition provided in [33].

Definition 3.3. (Evolving Cluster). Given: a set D of trajectories, a minimum cardinality threshold c, a maximum distance threshold θ, and a 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 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.

Definition 3.4. (Group Pattern Prediction Online). Given: a set D of trajectories, G of co-movement patterns up to timeslice TS^{now} and a look-ahead threshold Δt, we aim to predict all the valid co-movement patterns $G' \in (TS^{now}, TS^{now} + \Delta t]$.

Figure 1 provides an illustration of Definition 3.4. More specifically, we know the movement of nine objects from TS_1 until TS_3 and via EvolvingClusters with $c = 3$ and $d = 2$ that they form four evolving clusters $P_1 = \{a, b, c, d, e, f, g, h, i\}$, $P_2 = \{a, b, c, d, e\}$, $P_3 = \{a, b, c\}$, $P_4 = \{b, c, d, e\}$, $P_5 = \{g, h, i\}$. Our goal is to predict their respective locations until TS_5. Running EvolvingClusters with the same parameters for the predicted timeslices, reveals us (with high probability) that P_2, P_3, P_4, P_5 will continue to exist as well as the creation of a new pattern $P_6 = \{f, g, h, i\}$.

4 METHODOLOGY

In this section we present the proposed solution to the problem of *Online Prediction of Co-movement Patterns*, composed of two parts: a) the FLP method, and b) the Evolving Cluster Discovery algorithm. Also, an example is presented illustrating the approach operation.

4.1 Overview

Figure 2 illustrates the architecture of our proposed methodology. First we split the problem of *Online Prediction of Co-movement Patterns* into two parts, the FLP, and the Evolving Cluster Discovery. The FLP method is, also, divided to two parts: a) the FLP-offline part, where the training procedure of the model is taking place, and b) the FLP-online part, where the trained FLP model is applied to streaming GPS locations to predict the next objects' location.

Thus, our proposed approach is further divided in the offline phase and the online one. Particularly, at the offline phase, we

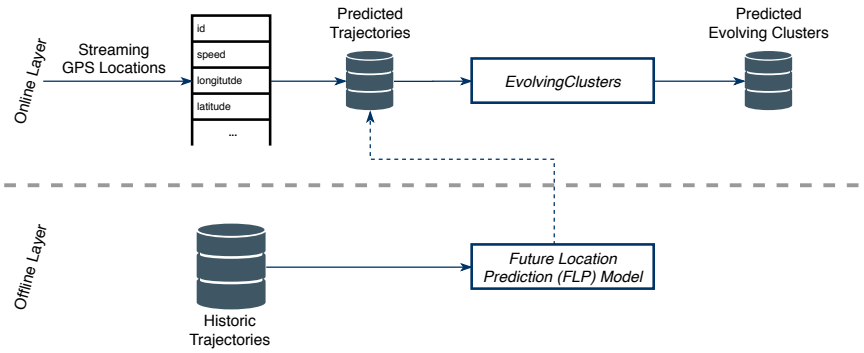

Figure 2: Workflow for evolving clusters prediction via (singular) trajectory prediction

train our FLP model by using historic trajectories. Afterwards, at the online phase we receive the streaming GPS locations in order to use them to create a buffer for each moving object. Then, we use our trained FLP model to predict the next objects' location and apply EvolvingClusters to each produced timeslice.

4.2 Future Location Prediction

Trajectories can be considered as time sequence data [37] and thus are suited to be treated with techniques that are capable of handling sequential data and/or time series [25]. Over the past two decades, the research interest on forecasting time series has been moved to RNN-based models, with the GRU architecture being the newer generation of RNN, which has emerged as an effective technique for several difficult learning problems (including sequential or temporal data -based applications) [4]. Although, the most popular RNN-based architecture is the well-known Long Short-Term Memory (LSTM) [9], GRU present some interesting advantages over the LSTM. More specifically, GRU are less complicated, easier to modify and faster to train. Also, GRU networks achieve better accuracy performance compared to LSTM models on trajectory prediction problems on various domains, such as on maritime [27], on aviation [7] and on land traffic [1]. Hence, this work follows this direction and employs a GRU-based method.

GRU includes internal mechanisms called gates that can regulate the flow of information. Particularly, the GRU hidden layer include two gates, a reset gate which is used to decide how much past information to forget and an update gate which decides what information to throw away and what new information to add. We briefly state the update rules for the employed GRU layer. For more details, the interested reader is referred to the original publications [3]. Also, details for the BPTT algorithm, which was employed for training the model, can be found in [35].

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

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

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

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

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 (longitude and latitude), the difference in time and the time horizon for which we want to predict the vessel's position; the differences are computed between consecutive points of each vessel.

In this work, a GRU-based model is employed to solve the future location prediction problem. The proposed GRU-based network architecture 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 (longitude and latitude). A schematic overview of the proposed network architecture is presented in Figure 3. Also, details for the Backward Propagation Through Time algorithm and for the Adam approach, which were employed for the NN learning purposes, can be found in [36] and [13], respectively.

4.3 Evolving Clusters Discovery

After getting the predicted locations for each moving object, we use *EvolvingClusters* in order to finally present 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 TS_{now}, EvolvingClusters works in a nutshell, as follows:

- Calculates the pairwise distance for each object within TS_{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 TS_{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, the set of objects o_{ids} that form an evolving cluster, the starting time st, the ending time et, and the type tp of the group pattern, respectively. For instance, the final output of the model at the example given at Section 3 would be a set of 4-element tuples, i.e., $\{(P_2, TS_1, TS_5, 2), (P_3, TS_1, TS_5, 1), (P_4, TS_1, TS_4, 1), (P_5, TS_1, TS_5, 1)\} \bigcup \{(P_4, TS_1, TS_5, 2), (P_6, TS_4, TS_5, 1)\}$, where $tp = 1(2)$ corresponds to MC (respectively, MCS). We observe that, the first four evolving clusters are maintained exactly as found in the historic dataset. In addition to those, we predict (via the FLP model) the following:

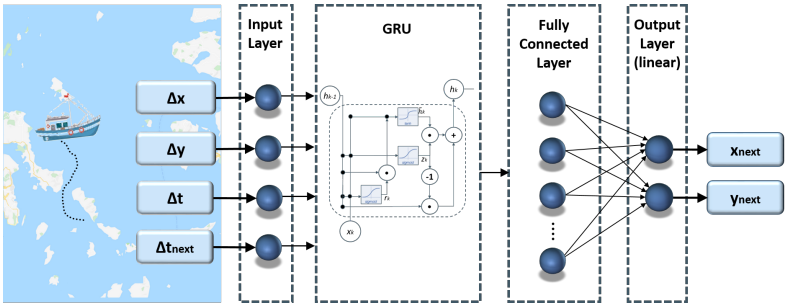

Figure 3: GRU-based neural network architecture.

- 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 at timeslice TS_5

In the Sections that will follow, we define the evaluation measure we use in order to map, each discovered evolving cluster from the predicted to the respective ones in the actual locations, as well present our preliminary results.

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 co-movement patterns will be quantified. Intuitively, we try to match each predicted co-movement pattern with the most similar actual one. Towards this direction, we need to define a similarity measure between co-movement patterns. In more detail, we break down this problem into three subproblems, the spatial similarity, the temporal similarity and the membership similarity. Concerning the spatial similarity this defined as follows:

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

where $MBR(C_{pred})$ ($MBR(C_{act})$) is the Minimum Bounding Rectangle of the predicted co-movement pattern (actual co-movement pattern, respectively). Regarding the temporal similarity:

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

where $Interval(C_{pred})$ ($Interval(C_{act})$) is the time interval when the the predicted co-movement pattern was valid (actual co-movement pattern, respectively). As for the membership similarity, we adopt the Jaccard similarity:

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

Finally, we define the co-movement pattern similarity as:

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

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 methdology for matching each predicted co-movement pattern C_{pred} with the corresponding actual one C_{act} is depicted in Algorithm 1.

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

Input: Evolving Clusters disovered 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

$EC_m \leftarrow \{\}$
for ***predicted pattern*** $C_{pred} \in EC_p$ **do**
  $similarity\_scores \leftarrow \{\}$
  $topSim = 0$
  for ***actual pattern*** $C_{act} \in EC_a$ **do**
    calculate $Sim^{*}(C_{pred}, C_{act})$
    if $Sim^{*}(C_{pred}, C_{act}) \geq topSim$ **then**
      $topSim = Sim^{*}(C_{pred}, C_{act})$
      $match_{best} \leftarrow C_{act}$
    end
  end
  $EC_m \leftarrow EC_m \cup match_{best}$
end

In more detail, we "match" each predicted co-movement pattern C_{pred} with the most similar actually detected pattern C_{act}. After all predicted clusters get traversed we end up with EC_m wich holds all the "matchings", which subsequently will help us in evaluate the prediction procedure by quantifuing the error between the predicted and the actual co-movement patterns.

6 EXPERIMENTAL STUDY

In this section, we evaluate our predictive model on a real-life mobility dataset from the maritime domain, and present our preliminary results.

6.1 Experimental Setup

All algorithms were implemented in Python3 (via Anaconda3[1] virtual environments). 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

[1]https://www.anaconda.com/

FLP and evolving cluster discovery, respectively. The machine we used is a single node with 8 CPU cores, 16 GB of RAM and 256 GB of HDD, provided by okeanos-knossos[2], an IAAS service for the Greek Research and Academic Community.

6.2 Dataset

It is a well-known fact that sensor-based information is prone to errors due to device malfunctioning. Therefore, a necessary step before any experiment(s) is that of pre-processing. In general, pre-processing of mobility data includes data cleansing (e.g. noise elimination) as well as data transformation (e.g. segmentation, temporal alignment), tasks necessary for whatever analysis is going to follow [22].

In the experiments that will follow, we use a real-life mobility dataset[3] from the maritime domain. The dataset, as product of our preprocessing pipeline, consists of 148,223 records from 246 fishing vessels organized in 2,089 trajectories moving within the Aegean Sea. The dataset ranges in time and space, as follows:

- Temporal range: 2nd June, 2018 – 31st August, 2018 (approx. 3 months)
- Spatial range: longitude in [23.006, 28.996]; latitude in [35.345, 40.999]

During the preprocessing stage, we drop erroneous records (i.e. GPS locations) based on a speed threshold $speed_{max}$ as well as stop points (i.e. locations with speed close to zero); afterwards we organize the cleansed data into trajectories based on their pairwise temporal difference, 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: $speed_{max} = 50knots$, $dt = 30min.$, and alignment rate equal to $1min$.

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 succesive points of the same trajectory.

6.3 Preliminary Results

In this section, we evaluate the prediction error of the proposed model with respect to the "ground truth". We define as "ground truth", the discovered evolving clusters on the actual GPS locations. For the pattern discovery phase, we tune *EvolvingClusters*, using $c = 3$ vessels, $d = 3$ timeslices, and $\theta = 1500$ meters. For the following experimental study, we focus – without loss of generality – on the MCS output of *EvolvingClusters* (density-connected clusters).

Figure 4 illustrates the distribution of the three cluster similarity measures, namely sim^{temp}, $sim^{spatial}$, and sim^{member}, as well as the overall similarity Sim^*. We observe that the majority of the predicted clusters are very close to their "ground truth" values, with the median overall similarity being almost 88%. This is expected however, as the quality of EvolvingClusters' output is determined by two factors; the selected parameters; and the input data. Focusing on the latter[4], we observe that the algorithm is quite insensitive to prediction errors, as deviations from the actual trajectory has minor impact to $sim^{spatial}$.

Figure 5 illustrates the previous discussion. More specifically, for the predicted and corresponding actual MCS with similarity

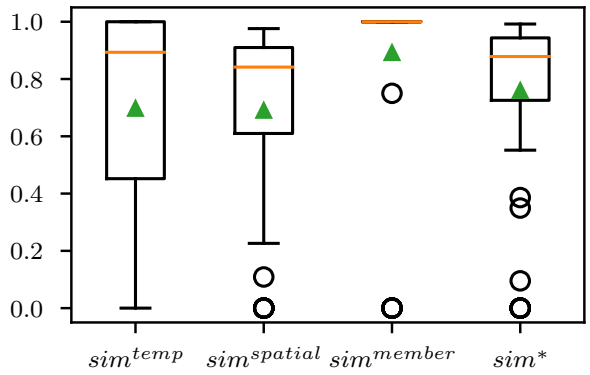

Figure 4: Distribution of Cluster Similarity Measures and Total Cluster Similarity

	Min.	Q25	Q50	Q75	Mean.	Max.
Record Lag	0	0	0	0	0.01	1
Consump. Rate	0	0	0	0	2.26	76.99

Table 1: Timeliness of the Proposed Methodology using Apache Kafka

close to the median, we visualize the trajectory of each participating object on the map, as well as the MBRs for each respective timeslice, in order to visualize the clusters' temporal and spatial similarity. It can be observed that deviations from the actual trajectories resulted in minor changes in the area of the points' MBR, and consequently to the overall similarity.

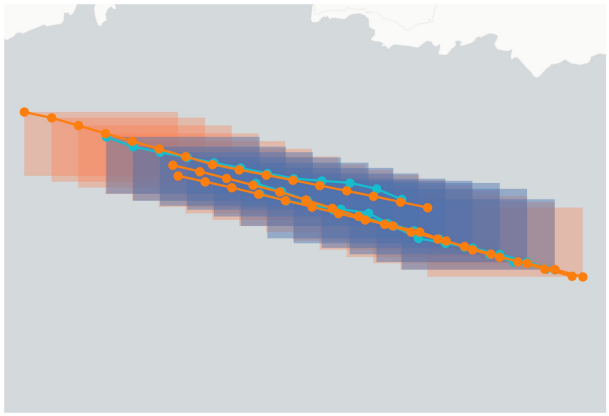

Figure 5: Trajectory of a predicted (blue) vs. an actual evolving cluster (orange)

Finally, Table 1 presents the metrics on the Kafka Consumers used for the online layer of our predictive model, namely, Record Lag and Consumption Rate. Observing the Record Lag, we deduce that our algorithm can keep up with the data-stream in a timely manner, while looking at Consumption Rate (i.e., the average number of records consumed per second) we conclude that our proposed solution can process up to almost 77 records per second, which is compliant with the online real-time processing scenario.

[2]https://okeanos-knossos.grnet.gr/home/
[3]Kindly provided to us by MarineTraffic.
[4]The parameter sensitivity of EvolvingClusters is out of the scope of this paper. For more details see [33]

7 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an accurate solution to the problem of *Online Prediction of Co-movement Patterns*, which is divided into two phases: *Future Location Prediction* and *Evolving Cluster Detection*. The proposed method is based on a combination of GRU models and Evolving Cluster Detection algorithm and is evaluated through a real-world dataset from the maritime domain taking into account a novel co-movement pattern similarity measure, which is able to match the predicted clusters with the actual ones. Our study on a real-life maritime dataset 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 develop an online co-movement pattern prediction approach that, instead of breaking the problem at hand into two disjoint sub-problems without any specific synergy (i.e. first predict the future location of objects and then detect future co-movement patterns), will combine the two steps in a unified solution that will be able to directly predict the future co-movement patterns.

ACKNOWLEDGMENTS

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.

REFERENCES

[1] A. Benterki, V. Judalet, M. Choubeila, and M. Boukhnifer. 2019. Long-Term Prediction of Vehicle Trajectory Using Recurrent Neural Networks. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, Vol. 1. 3817–3822.

[2] Lu Chen, Yunjun Gao, Ziquan Fang, Xiaoye Miao, Christian S. Jensen, and Chenjuan Guo. 2019. Real-time Distributed Co-Movement Pattern Detection on Streaming Trajectories. *Proc. VLDB Endow.* 12, 10 (2019), 1208–1220.

[3] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[4] R. Dey and F. M. Salem. 2017. Gate-variants of Gated Recurrent Unit (GRU) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243

[5] Qi Fan, Dongxiang Zhang, Huayu Wu, and Kian-Lee Tan. 2016. A General and Parallel Platform for Mining Co-Movement Patterns over Large-scale Trajectories. *Proc. VLDB Endow.* 10, 4 (2016), 313–324.

[6] Ziquan Fang, Yunjun Gao, Lu Pan, Lu Chen, Xiaoye Miao, and Christian S. Jensen. 2020. CoMing: A Real-time Co-Movement Mining System for Streaming Trajectories. In *SIGMOD 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 2777–2780.

[7] P. Han, W. Wang, Q. Shi, and J. Yang. 2019. Real-time Short- Term Trajectory Prediction Based on GRU Neural Network. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. 1–8. https://doi.org/10.1109/DASC43569.2019.9081618

[8] Shahab Helmi and Farnoush Banaei Kashani. 2020. Multiscale Frequent Co-movement Pattern Mining. In *36th IEEE International Conference on Data Engineering, ICDE 2020*. IEEE, Washington, DC, 829–840.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[10] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. 2008. Discovery of convoys in trajectory databases. *PVLDB* 1, 1 (2008), 1068–1080.

[11] Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. 2005. On Discovering Moving Clusters in Spatio-temporal Data. In *SSTD*. 364–381.

[12] Sameera Kannangara, Hairuo Xie, Egemen Tanin, Aaron Harwood, and Shanika Karunasekera. 2020. Tracking Group Movement in Location Based Social Networks. In *SIGSPATIAL '20: 28th International Conference on Advances in Geographic Information Systems*. ACM, 251–262.

[13] P. D. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

[14] Patrick Laube, Stephan Imfeld, and Robert Weibel. 2005. Discovering relative motion patterns in groups of moving point objects. *IJGIS* 19, 6 (2005), 639–668.

[15] Yuxuan Li, James Bailey, and Lars Kulik. 2015. Efficient mining of platoon patterns in trajectory databases. *Data Knowl. Eng.* 100 (2015), 167–187.

[16] Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kays. 2010. Swarm: Mining Relaxed Temporal Moving Object Clusters. *PVLDB* 3, 1 (2010), 723–734.

[17] H. Liu, H. Wu, W. Sun, and I. Lee. 2019. Spatio-Temporal GRU for Trajectory Classification. In *2019 IEEE International Conference on Data Mining (ICDM)*. 1228–1233. https://doi.org/10.1109/ICDM.2019.00152

[18] Mirco Nanni and Dino Pedreschi. 2006. Time-focused clustering of trajectories of moving objects. *J. Intell. Inf. Syst.* 27, 3 (2006), 267–289.

[19] Nikos Pelekis, Stylianos Sideridis, Panagiotis Tampakis, and Yannis Theodoridis. 2016. Simulating Our LifeSteps by Example. *ACM Trans. Spatial Algorithms Syst.* 2, 3 (2016), 11:1–11:39.

[20] Nikos Pelekis, Panagiotis Tampakis, Marios Vodas, Christos Doulkeridis, and Yannis Theodoridis. 2017. On temporal-constrained sub-trajectory cluster analysis. *Data Min. Knowl. Discov.* 31, 5 (2017), 1294–1330.

[21] Nikos Pelekis, Panagiotis Tampakis, Marios Vodas, Costas Panagiotakis, and Yannis Theodoridis. 2017. In-DBMS Sampling-based Sub-trajectory Clustering. In *EDBT*. 632–643.

[22] Nikos Pelekis and Yannis Theodoridis. 2014. *Mobility Data Management and Exploration*. Springer.

[23] Petros Petrou, Panagiotis Nikitopoulos, Panagiotis Tampakis, Apostolos Glenis, Nikolaos Koutroumanis, Georgios M. Santipantakis, Kostas Patroumpas, Akrivi Vlachou, Harris V. Georgiou, Eva Chondrodima, Christos Doulkeridis, Nikos Pelekis, Gennady L. Andrienko, Fabian Patterson, Georg Fuchs, Yannis Theodoridis, and George A. Vouros. 2019. ARGO: A Big Data Framework for Online Trajectory Prediction. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases, SSTD 2019, Vienna, Austria, August 19-21, 2019*. 194–197.

[24] Petros Petrou, Panagiotis Tampakis, Harris V. Georgiou, Nikos Pelekis, and Yannis Theodoridis. 2019. Online Long-Term Trajectory Prediction Based on Mined Route Patterns. In *MASTER@ECML-PKDD 2019*. 34–49.

[25] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri. 2020. Modelling Taxi Drivers' Behaviour for the Next Destination Prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 7 (2020), 2980 – 2989.

[26] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536.

[27] Yongfeng Suo, Wenke Chen, Christophe Claramunt, and Shenhua Yang. 2020. A Ship Trajectory Prediction Framework Based on a Recurrent Neural Network. *Sensors* 20, 18 (2020). https://doi.org/10.3390/s20185133

[28] Panagiotis Tampakis, Nikos Pelekis, Natalia V. Andrienko, Gennady L. Andrienko, Georg Fuchs, and Yannis Theodoridis. 2018. Time-Aware Sub-Trajectory Clustering in Hermes@PostgreSQL. In *ICDE*. 1581–1584.

[29] Panagiotis Tampakis, Nikos Pelekis, Christos Doulkeridis, and Yannis Theodoridis. 2019. Scalable Distributed Subtrajectory Clustering. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 950–959.

[30] Lu An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Chih-Chieh Hung, and Wen-Chih Peng. 2012. On Discovery of Traveling Companions from Streaming Trajectories. In *ICDE*. 186–197.

[31] George S. Theodoropoulos, Andreas Tritsarolis, and Yannis Theodoridis. 2019. EvolvingClusters: Online Discovery of Group Patterns in Enriched Maritime Data. In *MASTER@PKDD/ECML (Lecture Notes in Computer Science, Vol. 11889)*. Springer, 50–65.

[32] Roberto Trasarti, Riccardo Guidotti, Anna Monreale, and Fosca Giannotti. 2017. MyWay: Location prediction via mobility profiling. *Inf. Syst.* 64 (2017), 350–367.

[33] Andreas Tritsarolis, George-Stylianos Theodoropoulos, and Yannis Theodoridis. 2020. Online discovery of co-movement patterns in mobility data. *International Journal of Geographical Information Science* 0, 0 (2020), 1–27. https://doi.org/10.1080/13658816.2020.1834562

[34] C. Wang, H. Ren, and H. Li. 2020. Vessel trajectory prediction based on AIS data and bidirectional GRU. In *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*. 260–264. https://doi.org/10.1109/CVIDL51233.2020.00-89

[35] P. J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560. https://doi.org/10.1109/5.58337

[36] P. J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

[37] H. Xue, D. Q. Huynh, and M. Reynolds. 2018. SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision*. 1186–1194.

[38] Kai Zheng, Yu Zheng, Nicholas Jing Yuan, and Shuo Shang. 2013. On discovery of gathering patterns from trajectories. In *ICDE*. 242–253.