Key Concepts of group pattern discovery algorithms from spatio-temporal trajectories

Karolina K Dluzniak [#1], Wei Jie [#2], Hai Wang [*3], Wei Xing [**4]

[#]*School of Computing and Engineering, University of West London, UK*
[1]`kainka.github@gmail.com`
[2]`wei.jie@uwl.ac.uk`

[*]*School of Engineering and Applied Science*
Aston University, UK
[3] `h.wang10@aston.ac.uk`

[**]*School of Computer Science and Technology*
Xi'an University of Posts and Telecommunications, China
[4]`xingwei@xupt.edu.cn`

Abstract— **Over the years, the increasing development of location acquisition devices have generated a significant amount of spatio-temporal data. This data can be further analysed in search for some interesting patterns, new information, or to construct predictive models such as next location prediction. The goal of this paper is to contribute to the future research and development of group pattern discovery algorithms from spatio-temporal data by providing an insight into algorithms design in this research area which is based on a comprehensive classification of state-of-the-art models. This work includes static, big data as well as data stream processing models which to the best of authors' knowledge is the first attempt of presenting them in this context. Furthermore, the currently available surveys and taxonomies in this research area do not focus on group pattern mining algorithms nor include the state-of-the-art models. The authors conclude with the proposal of a conceptual model of Universal, Streaming, Distributed and Parameter-light (UDSP) algorithm that addresses current challenges in this research area.**

I. INTRODUCTION

The growth of tracking technologies and development of location acquisition devices have generated big volumes of spatio-temporal data. This data is different from traditional relational data as it adds both spatial and temporal attributes to regular observations. Shekhar et al. differentiates three distinct types of data attributes in the context of spatio-temporal data [1]: spatial, temporal and non-spatiotemporal. The latter provides additional characteristics of the moving objects such as their name (nominal), or exchange rate among (ratio) others. Spatial attributes can either define a location by providing for instance its geo-coordinates, spatial size, area or shape while temporal attributes can be anything from timestamps, snapshots, or duration.

Due to the complexity of spatio-temporal data itself as well as its continuously growing and changing volume there are several challenges which have been re-occurring in the related literature. These challenges can be summarized as per below:

- **Stream analysis**: Growth of streaming technologies which generates high volumes of data require a new approach to extract valuable information from such data [2], [3], [4], [5], [6].
- **Incremental discovery**: Many current applications (especially monitoring or surveillance) require the output results simultaneously while receiving and processing the streamed data which means the algorithms should be able to discover group patterns in an incremental manner [5], [6], [7], [1], [8].
- **Efficiency**: Streaming means huge amount of data generated in a very short time which leads to extreme computational cost of processing. The algorithms should then be able to output the groups efficiently [6], [9], [10].
- **Big data processing**: Huge amounts of spatio-temporal trajectories data require a distributed or parallel approach to handle the volumes and to reduce the computational overload [11], [12], [13], [14], [15].
- **Parameter-free algorithms**: Many algorithms include a large number of parameters and each of them has a significant impact on the performance and the precision of the results. There is no standardized way of choosing the right set of parameters rather than by conducting experiments [16], [17], [18].
- **Group continuity**: Traveling together may not always mean the members of the group stay together during the lifespan of the group. This is particularly noticeable in human trajectories and in larger groups where members of the group can form sub-groups [6], [19], [7]
- **Effectiveness**: The number of members of the group can vary and the effective algorithm should be able to understand the wider picture and report long-term groups rather than short term ones [4], [6], [20].
- **High Precision**: The challenge lies with discovering the correct groups rather than more groups which can make a difference in certain applications based on personalized services and targeting the wrong group may be worse than not addressing it at all [5], [6], [2], [19].
- **Data mining theory and formalization**: Esling as well

as many other researchers noticed the lack of formalization of data mining theory such as through solid mathematical foundation [21], [22], [23].

II. RELATED WORK

There have been several distinct approaches to group pattern mining from spatio-temporal trajectories. One of the widely discussed approaches is flock pattern which is also considered one of the earliest group pattern discovery approaches [24], [25], [26]. The second approach, convoy, argues that the first model is not sufficient for correct group discovery due to its limitation of pre-defined group search region which leads to group omissions (also known as so-called lossy-flock problem). Instead, this second approach proposes to capture trajectory pattern of any shape by applying the density-based clustering [22], [27]. The third approach, swarm identifies another challenge with both aforementioned approaches as they have a strict requirement on the k-consecutive time points for group discovery [28]. This may result in ignoring certain groups depending on the k-consecutive time set up. For instance if k is set up to 9 but the objects would travel together during k=7, then this group would have been missed. Instead, the swarm approach introduces a cluster of objects that can last for at least k but non-consecutive time stamps. While all these approaches use static datasets and additionally convoy and swarm require loading the entire trajectories into memories in order to identify the clusters, the traveling companion approach implements a data structure called *traveling buddy* [4] which allows finding patterns from streamed data and therefore sets up a ground for further approaches. Lastly, the most recent approach - loose traveling companion pattern addresses certain limitations of the previous approaches by focusing on human and not object/animal trajectories as presented in the previous patterns and considering indoor environment.

III. GROUP AND MOVEMENT PARAMETERS

In order to identify and specify the group patterns that are the subject of this paper, certain parameters have been established which allow to distinguish them. The initial top-down approach to classify them has been based on the group and movement parameters. The first and most obvious criterion is the size parameter of the group and in the context of groups, the focus is on algorithms concerning groups sizes greater than 1. The movement parameters are derived from kinesiology which defines the physical movement of objects which matches the group pattern discovery from spatio-temporal trajectories as it is understood in the context of this work as moving point objects trajectories. This classification is based on two concepts that were extended for the purposes of this work. One called REMO (RElative Motion) which compares the motion attributes of moving point objects over time and space [26]. The second is following Dodge's classification who defined the set of moving parameters by grouping them into primitive, primary and secondary derivatives [23] based on the spatial, temporal and spatio-temporal dimensions. This set of movement parameters have been further extended and

modified to allow the classification of the patterns that are presented in this work. Primitive parameters in spatial dimension are simply the (x,y) position of the moving point object. In the context of the temporal dimension these are the instances and intervals of time. A primary derivative of spatial dimension is the function of position such as distance, and then a secondary derivatives which is the function of distance are density, radius and similarity parameters. In regards to temporal dimension, the primary derivative is duration, lifetime of a group and window while the secondary derivative in this context is a frequency which is a function of duration. In spatio-temporal dimension (x,y,t) the primary derivatives are speed and velocity both being functions of (x,y,t) while the secondary derivatives include for instance acceleration which is the function of speed. The aforementioned derivatives correspond to the thresholds that are set up for each of the considered algorithms as shown in the Table I.

Thresholds are the most significant differentiators between various algorithm approaches in group pattern mining from spatio-temporal trajectories. They also generate significantly different outcomes within the same algorithms depending on threshold set up. Zheng in the survey on trajectory data mining [22] mentions three key differentiating thresholds which are the group size, density or shape of the group and the duration of the pattern. Further review of the algorithms from this research area shows that many other thresholds play an important role in group discovery. The significant thresholds appearing in the subject literature are following:

- **Size threshold** $\delta_{\mathbf{s}}$ is the size of the targeted group to be discovered from pattern mining
- **Duration threshold** $\delta_{\mathbf{d}}$ is the minimum lifetime of a group.
- **Gap threshold** $\delta_{\mathbf{g}}$ refers to the time-gap between cluster-sets which breaks the strict k-consecutive continuity of the group and allows the group members to leave the group and re-join within the specified gap-threshold.
- **Density threshold** $\delta_{\mathbf{dn}}$ is derived from the density-based clustering method that is commonly used in the pre-processing stage (DBSCAN) where for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum of points so the density of this neighborhood needs to exceed the density threshold [29]
- **Distance threshold** $\delta_{\mathbf{ds}}$ is related to the same density-based clustering method and defines the shape of the neighborhood as a distance function of two points p and q dist(p,q) [29]
- **Frequency threshold** $\delta_{\mathbf{f}}$ defines the minimum time-slots when the group stays together
- **Radius threshold** $\delta_{\mathbf{r}}$ is specific to flock group of patterns where at least *m* number of moving objects *O* are found within a circular region of constant radius $\delta_{\mathbf{r}}$ ¿ 0 and heading in the same direction at some point of time
- **Similarity threshold** $\delta_{\mathbf{sm}}$ defines the minimum of overlapping moving point objects in any two consecutive clusters and the value of $\delta_{\mathbf{sm}} \in [0,1]$. Given the example of $\delta_{\mathbf{sm}} = 0.7$, it is required that 70% of the clusters are

TABLE I

TABLE OF NOTATIONS

Notation	Definition
O	moving object
O$_{\mathbf{set}}$	set of objects
V/V$_{\mathbf{next}}$	candidates/current candidates
Tr	trajectory
Ts	trajectory stream
C	cluster
t	time-slot index/snapshot
$\delta_{\mathbf{s}}$	size threshold
$\delta_{\mathbf{d}}$	duration threshold
$\delta_{\mathbf{g}}$	gap threshold
$\delta_{\mathbf{dn}}$	density threshold
$\delta_{\mathbf{ds}}$	distance threshold
$\delta_{\mathbf{f}}$	frequency threshold
$\delta_{\mathbf{r}}$	radius threshold
$\delta_{\mathbf{sm}}$	similarity threshold
P	group (pattern)

required to overlap.

IV. ALGORITHM DESIGN

Depending on data attributes, size and type of input data as well as the set of thresholds/parameters, the algorithms are represented by different models. The notations used in the pseudo-codes are shown in the Table I.

The models presented in this section are grouped in the following categories resulting from the classification based on the input data type and size:

- **static data algorithms:** this group is dominant as the majority of currently available algorithms in this research area use a static database and follow a more traditional KDD process such as flock, convoy, swarm, gathering, leadership, convergence, encounter, meet, MEMO, moving flock, group pattern (Apriori Group Pattern), moving convoy.
- **streamed data algorithms:** the development of this group became popular in the last few years and include loose and traveling companion models with a traveling buddy to store a reference rather than data for fast processing, LTCP and WCLTCP, notions of a continuous discovery of moving clusters [30], BFA(basic flock evaluation), or EVOCO (can be used for both static and streaming)
- **distributed/parallel algorithms:** Distributed algorithms in this research area are currently in minority with few examples such as distributed convoy by Orakzai [11], adaptive flock by Folino [31], or grid and spatial index methods such as TrajStore [32] for distributed trajectory mining rather than a group discovery

A. Static Data Algorithms

The generic model from trajectory static database is presented in Algorithm 1. It takes the input of a static, historical trajectory data as well as a set of thresholds. It is the most commonly represented within the group patterning discovery algorithms from spatio-temporal trajectories. This algorithm

model is relevant to majority of the classical algorithms such as convoy [27], [10], or swarm [28], as well as gathering [7], or group pattern (AGP) [33]. Flock-based patterns such as "classical" flock [25], [24], [26], moving flock [34], leadership [35], [25], [26], convergence [35], [25], [26], encounter [35], [25], meeting of moving objects(MEMO) [36] and meet [35], [25], [26] can also be grouped into this static model approach with one exception concerning the group formation. Unlike all previously mentioned models in this group that apply different methods of clustering to extract the group members, the flock-based patterns use a predefined, circular region with a constant radius to identify the members of the group, rather than via other clustering methods.

These static group discovery models can be split into three core stages as shown in the generic Algorithm 1 model:

1) pre-processing(clustering/region) (lines 1-2) // finds all possible combinations from trajectory (snapshot clustering)
2) extension of the candidates (lines 3-4) //validation to see if the candidates can form a pattern (group discovery)
3) adding new candidates (line 5) //final refinement (group pattern detection)

Algorithm 1 Generic static approach

Input: Tr, $\delta_{\mathbf{s}}$, $\delta_{\mathbf{d}}$/ $\delta_{\mathbf{g}}$/ $\delta_{\mathbf{dn}}$/ $\delta_{\mathbf{f}}$/ $\delta_{\mathbf{r}}$/ $\delta_{\mathbf{ds}}$ ▷ trajectory database data and a selection of thresholds

    Output: all closed groups of moving objects

1: **for** `<each time-slot, sub-trajectory/trajectory>` **do**
2:    `<cluster>`
3: **for** `<each potential candidate>` **do**
4:    `<extend current candidates list >`
5:    **if** extended candidates match thresholds **then**
      return group of moving objects

B. Streamed Data Algorithms

Different from data stored in static databases, streaming data is not constant and may have a variable data distribution which changes in time as the data is streamed [37]. Streaming trajectories are dynamic as the positions of objects are being continuously updated and therefore traditional ways of indexing such as R-tree or quad-tree require a high computational cost so new structures to store relevant data are required [4], [38]. In addition to this, the data from streaming trajectories may be subject to various delays and therefore the objects that potentially could be a part of the same group may not report their positions on time. When looking at data streams both offline streams and online streams can be distinguished [37]. Offline streams are usually including regular bulk additions of data while online streams includes real-time updated data that arrive sequentially one after another. Different from static models, this group of streaming data algorithms is designed to work with both offline and online streaming data inputs which imposed different solutions to those presented in the

aforementioned static examples. The static models require to load the whole trajectory of dataset into memory and therefore are not suitable for data streams which requires incremental processing. The algorithms in this streaming group include traveling companion and loose companion [4], [5], loose traveling companion pattern and weakly continuous loose traveling companion pattern [6], adaptive flock [31], online/incremental gathering and basic flock evaluation [39]. The generic streaming model is shown in Algorithm 2 and can be divided into the following steps:

1) initialize a temporary candidate set and cluster all objects from incoming stream/snapshot (lines 1 -3)
2) intersect existing candidates with temporary set and store the intersection result as new candidates (lines 4 -6)
3) validate new candidates (lines 7 -9)
4) return up-to-date group pattern to be processed in the next snapshot (line 10)

Algorithm 2 Generic streaming approach
___
    Input: Ts, $\delta_\mathbf{s}$, V $\delta_\mathbf{d}$/ $\delta_\mathbf{g}$/ $\delta_\mathbf{dn}$/ $\delta_\mathbf{f}$/ $\delta_\mathbf{r}$/ $\delta_\mathbf{ds}$ ▷ trajectory stream data, a selection of thresholds and candidates set
    Output: $\mathbf{O_{set}}$ ▷ all up-to-date groups of moving objects
1: **for** `<each snapshot> do`
2:    `<initialize a temporary candidate set>`
3:    `<cluster>`
4: **for** `<each potential candidate> do`
5:     **for** `<each cluster from Ts> do`
6:       `<new candidate = intersection of existing candidates with temporary candidates >`
7:         **if** new candidate matches minimum thresholds **then**
8:           add new candidate to temporary candidate set
9:         **if** temporary candidate set matches thresholds **then**
10:           add new candidates to existing candidates
      return up-to-date group of moving objects   ▷ to be processed in the next snapshots

---

C. Distributed/Parallel Algorithms

Continuously growing spatio-temporal data requires a big data processing approach. While the static models use memory processing which is not suitable for big trajectories, the streamed models propose a different approach to mine the streaming data. However, all of the algorithms in the latter group use small datasets for processing that can be processed on single machines. Distributed/parallel models for group pattern mining in this research area are relatively very few. The main algorithms in this group are distributed convoy [11], parallel adaptive flock [31], or another parallel approach using Swarm Intelligence - SPARROW [40]. Another way of handling group pattern mining from large datasets are grid-based models such as moving flock [34]. A generic model for

distributed algorithms is shown in Algorithm 3 and it can be divided into three key stages:

1) partitioning
2) local group pattern mining
3) merging results into a global result

Algorithm 3 Generic distributed approach
___
    Input: partitions of Tr, $\delta_\mathbf{s}$, $\delta_\mathbf{d}$/ $\delta_\mathbf{g}$/ $\delta_\mathbf{dn}$/ $\delta_\mathbf{f}$/ $\delta_\mathbf{r}$/ $\delta_\mathbf{ds}$ ▷ partitions of trajectory data and a selection of thresholds
    Output: $\mathbf{O_{set}}$ ▷ all groups of moving objects ▷ global result from all partitions
1: **for**   `<each moving object property/spatial/temporal dimension> do`
2:    `<partition data>`   ▷ results in disjoint subsets
3: **for** `<each disjoint subset> do`
4:    `<perform a local group pattern mining >`
      return global result of group of moving objects

D. Universal, Streaming, Distributed and Parameter-light Algorithm (UDSP)

Taking into consideration the classification criteria we based on the differentiation of the algorithms in this research area such as data type, data size and group and movement parameters/thresholds, the proposed conceptual model aims to address the following criteria in group pattern discovery from spatio-temporal trajectories:

- big data processing via distributed computing
- handling streaming data
- parameter-free approach (or parameter-light)
- universal group pattern model

Since there are not many group pattern mining algorithms in this research area that propose a distributed framework and the moving object data is continuously growing, a distributed model to extract group patterns is in high demand. Furthermore, historical and static data input is not enough to capture the patterns as there many real-live applications require an incremental pattern mining from streaming data. The streaming data algorithms are a growing trend in spatio-temporal data mining. However, there are still many unresolved issues such as handling the dynamically changing data where it is hard to find a stable, constant element that can be referred to, or stored and that is not computationally expensive to be temporarily stored or processed. Another challenge is related to error-prone data due to imprecision of physical devices so introducing a structure to capture these changes and improving data precision is also important. Currently available models introduce a structure to store in memory certain information for future processing. However, since the streaming data is potentially endless and storing and processing these temporary structures can be expensive from computational point of view, one of the possible solutions is to look at some form of

universal data structure to capture the incoming data stream and use for clustering accordingly

Parameter-free approach aims at reducing the bias coming from too many, or wrongly chosen parameters that can either cause the algorithm to fail, or to discover unwanted patterns Keogh2004. This area of research is not commonly applied to group pattern mining from spatio-temporal trajectories. However, it could be used in the pre-processing stage to extract the widest possible cluster-sets of meaningful data for group pattern mining.

Lastly, a universal group pattern model that does not depend on specific pattern criteria, can be achieved by creating a customisable model that can take any thresholds defining groups. An interesting line of research which is not directly related to spatio-temporal data looks at thresholds estimation in order to machine-select best thresholds for the pattern mining but could potentially be used as an option to arbitrary human selection of the group defining thresholds Mavridis2013.

Algorithm 4 Universal Streaming Distributed and Parameter-free approach

---

    Input: Tr, Ts   ▷ trajectory data, streaming and static
    Output: O_{set} ▷ global result from all partitions, all groups of moving objects

1: **if** input = Ts **then**   ▷ streaming data input
2:    `<follow Algorithm 5>`
3: **else**`<continue>`
4: **for**   `<each moving object's temporal dimension>` **do**
5:    **if** ¡input = macro-cluster¿ **then**
6:       `<partition by temporal dimension>`
7:    **else**`<continue>`
8:    `<partition data>`   ▷ results in disjoint subsets
9: **for** `<each node, each partition>` **do**
10:    `<parameter-free clustering>`   ▷ results in local clusters
11: **for** `<local cluster>` **do**
12:    `<perform a local group pattern mining >`
13: **for** `<each local candidates >` **do**
14:    `<merge closest candidates>`   ▷ checks local candidates with their closes candidates from closest nodes
15:    `<update global candidates>`
16: **for** `<each global candidate>` **do**
17:    `<apply thresholds >` ▷ checks if global result is a requested group
     return global result of group of moving objects

Taking into consideration the above criteria, a framework of a Universal, Streaming, Distributed and Parameter-free (USDP) model/algorithm 4 is proposed to include the following key stages:

1) Distributed parameter-light pre-processing
2) Distributed streaming

3) Local and global pattern mining

The distributed pre-processing is based only on one criterion of partitioning data into distributed nodes based on temporal dimension which makes it parameter-light. Following current research in this matter, the partition based on time attribute rather than the spatial attribute has a benefit of a quick identification of possible candidates that could have been split into different nodes as the time dimension is uni-directional and moves forward rather than in many directions like the spatial attributes of the moving objects Orakzai2016. In addition to this the streamed data will be easier to fit in the distributed model in order to check streamed candidates against the existing ones if we looked at the temporal dimension. Alternatively, a parameter-free incremental distributed hierarchical clustering can be considered which could free the initial partitioning of data from the temporal dimension criterion.

Depending on the size of streaming data there are two approaches. If the streaming data is considered big, the first approach is to add the incremental streaming data to the main distributed partitioning and process it locally over partitioned nodes so that it follows the whole partitioning distributed model. If the streamed input data is manageable for initial pre-processing, then the second approach is to use the *Micro-and-Macro-Cluster* framework where the streaming input is firstly pre-processed using refinement techniques such as line segmentation and then micro and macro-clustering data in order to eventually cross-check it against the global result coming from the main distributed model.

To distributed streaming process is following:

1) For each new streamed input
2) do initial pre-processing (cluster input (line segmentation and other techniques to refine data if missing)
3) start: micro-clustering
4) For each cluster-set find the nearest micro-cluster matching the cluster-set
5) macro-cluster
6) return macro-cluster to be partitioned
7) partition macro-clusters to include in the nodes and process together with the existing data

Alternatively, a new data structure can be introduced to store candidates sets from the nodes that is lightweight enough to be kept for certain period of time and then cross-checked with the streamed candidates.

Lastly, the local and global pattern mining will be based on a set of group and movement defining thresholds which can be either user-defined, or machine-chosen.

V. CONCLUSIONS

The focus of this paper was to provide an overview of the current research and its challenges in the area of group pattern discovery from spatio-temporal trajectories aiming at classifying the state-of-the-art algorithms. Despite of limitations resulting from the lack of access to full coding and tested datasets in majority of the cases, the classification was based on pseudo-codes. With continuously growing spatio-temporal

Algorithm 5 USDP - Streaming

    Input: Ts   ▷ streaming data
    Output: all valid macro-clusters

```
1: for <each new snapshot> do
2:     <line segmentation>
3:     <map-matching>
4: for <each line segment> do
5:     <find the nearest one within the
       sliding-window>
6:     <create micro-cluster>
7: for <each micro-cluster> do
8:     <check thresholds against other
       micro-clusters >
   return all valid macro-clusters
```

---

data there is a clear gap in group pattern discovery which does not scale up for big data and distributed computing. In addition to this many currently developed devices such as CCTV provide data streams while a significant majority of algorithms in spatio-temporal research area is designed to handle only static datasets. Due to the complexity of spatio-temporal data processing too many parameters add additional computational overhead and therefore there is a need for parameter-free or parameter-light models to optimise their performance. Taking all these key challenges into consideration, the authors have proposed a theoretical model of Universal, Streaming, Distributed and Parameter-free (USDP) algorithm to fill the gap in this area. The algorithm includes distributed big data processing, is optimized for data streams and includes only one parameter in the pre-processing stage. Seeing that there are many thresholds differentiating the group patterns in this research area, the presented concept aims at being universal where group thresholds can be either user or machine-defined and can include a range of various thresholds depending on the application requirements. The proposed model is in its early stage and requires further research, implementation and evaluation. The future implementation is considered via Spark platform as it is an efficient alternative to MapReduce via Hadoop which is currently being used for the distributed models in this research area. Spark includes GeoSpark and SpatialSpark which have spatial processing support inbuilt, however there is a gap in spatio-temporal tools available with some very few new options proposed such as STARK that complement the current tools with both spatial and temporal elements for big data processing.

REFERENCES

[1] S. Shekhar, Z. Jiang, R. Ali, E. Eftelioglu, X. Tang, V. Gunturi, and X. Zhou, "Spatiotemporal Data Mining: A Computational Perspective," *ISPRS International Journal of Geo-Information*, no. 4, pp. 2306–2338.

[2] J. Mao, Q. Song, C. Jin, Z. Zhang, and A. Zhou, "Online clustering of streaming trajectories," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 245–263, 2018.

[3] C. C. Aggarwal, T. J. Watson, R. Ctr, J. Han, J. Wang, and P. S. Yu, "A Framework for Clustering Evolving Data Streams," *Proc. of the 29th int. conf. on Very large data bases*, pp. 81–92.

[4] L. A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C. C. Hung, and W. C. Peng, "On discovery of traveling companions from streaming trajectories," *Proceedings - International Conference on Data Engineering*, no. 3, pp. 186–197, 2012.

[5] L.-A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, W.-C. Peng, and T. L. Porta, "A framework of traveling companion discovery on trajectory data streams," *ACM Transactions on Intelligent Systems and Technology*, no. 1, pp. 1–34.

[6] E. Naserian, X. Wang, K. Dahal, Z. Wang, and Z. Wang, "Personalized location prediction for group travellers from spatial temporal trajectories," *Future Generation Computer Systems*, pp. 278–292.

[7] K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, and X. Zhou, "Online discovery of gathering patterns over trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1974–1988, 2014.

[8] L. Wang, L. Ke, L. Peng, R. Rajiv, and L. Chen.

[9] J. Gudmundsson and M. van Kreveld, "Computing longest duration flocks in trajectory data," *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems - GIS '06*, p. 35.

[10] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of Convoys in Trajectory Databases."

[11] F. Orakzai, T. Calders, and T. B. Pedersen, "Distributed convoy pattern mining," *Proceedings - IEEE International Conference on Mobile Data Management*, vol. 2016-July, pp. 122–131, 2016.

[12] D. Matekenya, M. Ito, R. Shibasaki, and K. Sezaki, "Enhancing location prediction with big data," *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct - UbiComp '16*, pp. 753–762.

[13] P. Götze and S. Hagedorn, "The STARK Framework for Spatio-Temporal Data Analytics on Spark," *Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW)*, pp. 123–142, 2017.

[14] L. Wang, Y. Ma, J. Yan, V. Chang, and A. Y. Zomaya, "pipsCloud: High performance cloud computing for remote sensing big data management and processing," *Future Generation Computer Systems*, pp. 353–368.

[15] L. Wang, H. Shiyan, B. Gilles, and R. Rajiv.

[16] L. Mavridis, N. Nath, and J. B. Mitchell, "PFClust: A novel parameter free clustering algorithm," *BMC Bioinformatics*, no. 1, p. 213.

[17] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, Seattle, WA, p. 206.

[18] J. H. Faghmous, M. Le, M. Uluyol, V. Kumar, and S. Chatterjee, "A parameter-free spatio-temporal pattern mining model to catalog global ocean dynamics," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 151–160, 2013.

[19] P. Kalnis, N. Mamoulis, and S. Bakiras, "On Discovering Moving Clusters in Spatio-temporal Data," pp. 364–381.

[20] K. . Zheng, Y. . Zheng, J. . Yuan, S. Shang, K. Zheng, Y. Zheng, and N. J. Yuan, "On Discovery of Gathering Patterns from Trajectories," pp. 242–253.

[21] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys*, no. 1, pp. 1–34.

[22] Y. U. Zheng, "Trajectory Data Mining : An Overview," *ACM Trans. Intell. Syst. Technol.*, no. 3, pp. 1–41.

[23] S. Dodge, R. Weibel, and A. K. Lautenschtz, "Towards a taxonomy of movement patterns," in *Information Visualization*, no. 3-4, pp. 240–252.

[24] M. Benkert, J. Gudmundsson, F. Hübner, and T. Wolle, "Reporting flock patterns," *Computational Geometry: Theory and Applications*, vol. 41, no. 3, pp. 111–125, 2008.

[25] J. Gudmundsson, "Efficient detection of motion patterns in spatio-temporal data sets," *Proceedings of the 12th annual ACM international workshop on Geographic information systems - GIS '04*, pp. 250–257.

[26] P. Laube and S. Imfeld, "Analyzing Relative Motion within Groups ofTrackable Moving Point Objects." Springer, pp. 132–144.

[27] H. Jeung, H. T. Shen, and X. Zhou, "Convoy queries in spatio-temporal databases," *Proceedings - International Conference on Data Engineering*, pp. 1457–1459, 2008.

[28] Z. Li, B. Ding, J. Han, and R. Kays, "Swarm: Mining relaxed temporal moving object clusters," *VLDB International Conference on Very Large Data Bases*, no. 1-2, pp. 723–734.

[29] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise."

[30] C. S. Jensen, D. Lin, and B. C. Ooi, "Continuous clustering of moving objects," *IEEE Transactions on Knowledge and Data Engineering*, no. 9, pp. 1161–1173.

[31] G. Folino and G. Spezzano, "An Adaptive Flocking Algorithm for Spatial Clustering," *7th International Conference on Parallel Problem Solving from Nature*, no. May 2009, p. 924933.

[32] P. Cudre-Mauroux, E. Wu, and S. Madden, "TrajStore: An adaptive storage system for very large trajectory data sets," in *Proceedings - International Conference on Data Engineering*, pp. 109–120.

[33] Y. Wang, E. P. Lim, and S. Y. Hwang, "Efficient mining of group patterns from user movement data," *Data and Knowledge Engineering*, vol. 57, no. 3, pp. 240–282, 2006.

[34] Andres Oswaldo Calderon Romero, "Mining moving flock patterns in large spatio-temporal datasets using a frequent pattern mining approach," no. March, pp. 1–79.

[35] P. Laube, M. van Kreveld, and S. Imfeld, "Finding REMO detecting relative motion patterns in geospatial lifelines," *Developments in Spatial Data Handling: Proceedings of the 11th International Symposium on Spatial Data Handling*, pp. 201–214.

[36] H. H. Aung and K.-L. Tan, "PhD Showcase: Mining Multi-object Spatial-temporal\ Movement Patterns," *SIG SPATIAL Special*, no. 3, pp. 14–19.

[37] N. Jiang and L. Gruenwald, "Research issues in data stream association rule mining," *ACM SIGMOD Record*, no. 1, pp. 14–19.

[38] M. L. Lee, W. Hsu, C. S. Jensen, B. Cui, and K. L. Teo, "Supporting Frequent Updates in R-Trees," in *Proceedings 2003 VLDB Conference*, pp. 608–619.

[39] M. R. Vieira, P. Bakalov, and V. J. Tsotras, "On-line discovery of flock patterns in spatio-temporal data," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, Seattle, WA, p. 286.

[40] G. M. Jacob and S. M. Idicula, "Detection of flock movement in spatio-temporal database using clustering techniques - An experience," in *Proceedings - 2012 International Conference on Data Science and Engineering, ICDSE 2012*, 2012.