# Implementation of Mining Trajectory Patterns and Mobility Borders

Miss.Ashwini Bhingare[1], Mr. R.L.Paikrao[2]

[1]M.E,Computer,Amrutvahini college of engineering,sangamner,Maharashtra,India

[2]M.Tech,Ph.D(regi.), Computer,Amrutvahini college of engineering,sangamner,Maharashtra,India

## ABSTRACT

*Some traditional methods of UT pattern mining are inefficient as well as more complicated which only capable of identifying specific type of trajectory pattern from input dataset. We proposed UT-pattern mining framework to address the limitations of previous methods. It contains two phases to mine UT patterns such as, initial pattern mining & pattern forest construction. Clusters of various UT patterns are constructed using Traclus method. This cluster contains initial UT patterns. After pattern forest construction, patterns are classified into different categories such as, time relaxed, time constrained & time independent. Along with the UT pattern discovery & classification proposed system identifies Geographical Mobility Borders. With experimental result system demonstrates the efficiency of UT pattern framework in terms of time and memory.*

**Keywords**: *—Trajectory pattern mining, synchronous movement patterns, moving object trajectories, trajectory clustering*

---

## I.INTRODUCTION

Unifying trajectory mining framework consists of set of trajectory or set of moving objects closely related to their location and time.There are two types of movement discussed in existing work such as, synchronous and asynchronous movements. Synchronous movement can be defined as, objects interacting with each others with the small time interval whereas, asynchronous patterns can be defined as, objects are moving together. Unified trajectory patterns consist of a set of trajectories which closely related to the location & time hence they are included into asynchronous category. Unifying trajectory patterns have various application areas such as, deer migration, Wolf predation on wild ungulates. UT pattern mining is very useful in learning interactions between moving objects. Previously, lots of efforts have been conducted on the work of UT pattern mining such as, flock patterns [2], convoy patterns [2], swarm patterns [4], moving clusters [8], time-relaxed trajectory joins [10], hot motion paths [5], and sub-trajectory clusters etc. From analysis of existing system it is observed that only specific trajectory pattern can be identified from given input dataset which is more tedious & inefficient task. Other limitations could be that user is unaware of which types of trajectories are hidden into given dataset. Motivating from these limitations, UT pattern mining framework is proposed to discover UT patterns from large dataset and classified them into three types of categories such as, time-relaxed, time-constrained, and time-independent patterns. Line segmentation, vector creation strategies are implemented during initial cluster creation. Whereas, in pattern forest construction, data compression, reference movement extraction, pattern distribution is performed. In pattern forest construction, dill down and roll up approaches are implemented with MDL principle.

Two types of phases are included into proposed framework such as, initial pattern discovery & granularity phase. These both phases are guided by information-theoretic formula which is based on principle of minimum description length (MDL). In sub-trajectory cluster formation two phases are included such as, partitioning phase & grouping phase. In partitioning phase, trajectory is partitioned into a set of line segment which also known as, trajectory partitioned whereas, in grouping phase similar kind of trajectories are grouped into same cluster. Grouping of line segments are based on density based clustering method. The final stage of trajectory pattern grouping is also called as, representative trajectory. It is the sequence of points similar to the ordinary trajectory generated for each cluster. Trajectory partitioned are converted to location, time space which makes analysis simpler as well as easier almost without losing an accuracy. Reference movement is calculated which capture underlying patterns of

partitioned trajectory. Similarity measure is calculated between line segments is required for quantifying the degree of data compression. An approximation algorithm is used for initial UT pattern generation & it receives the set of trajectory partitioned that belonging to the same sub-trajectory cluster & returns a set of UT-patterns. In granularity phase, pattern forest is constructed using previous initial patterns.
OLAP-operations are used in pattern forest construction; it contains two types of operations such as, drill-down and roll up operations. Drill down operation decreases the cost of MDL principle by selecting dimensions automatically. Opposite to the drill down approach, roll up operation does not decrease the cost of MDL.
The system contributes mobility border identification.

## II. RELATED WORK

In [2], P. Laube and S. Imfeld et al., discussed about flock patterns which represents the concurrence motion. It contains subset of moving objects along with the close path to each other in some time intervals. Spatial temporal patterns are defined as, flock, leadership patterns. Flocking patterns mainly used for generic aggregation pattern [3]. It consists of REMO pattern concurence as well as spatial constraints. Effiecient pattern detection is the more complicated task which required to separate input data having equal time and equal motion so that n equal number of set of same direction and the same time can get.

For the seperation of MPO i.e. Modeling design patterns REMO is strictly designed in [4]. The MPO seperation is required to manage & track data seperatly from the task of analysis. MPO keeps the track of data & makes the computation of their motion attributes in analysis phase. Flock patterns have been proposed by M. Benkert et al. to determine the perfomance of tree-based algorithm which is well suitable for flock pattern identification. However it is much depend on the characteristics of input set. M. Nanni and D. Pedreschi defined a time-focused clustering for mining trajectories of moving objects[13].
In[9] J. Gudmundsson discussed the problem of computation of longest duration flock patterns.
Several problems in trajectory pattern mining have been identified such as, propose and design techniques for more complex patterns and implemented techniques that can manage spatio-temporal data with errors and missing values. Time relaxed trajectory joins manifested on basic symbolic join algorithms. Traditionally, there are two types of approaches are available.
From this first is based on the concept of multiple origins and the other is hueristic solution based on "divde and conquer" method[6]. It is only suitable with low memory resources. D. Sacharidis and K. Patroumpas suggested hot motion path. It is time relaxed trajectory joins to detect frequently traveled trails of numrous moving objects. A distributed setting is considered having co-ordinators maintaining hotness and geometrics of this paths. This scenario is only limited to freely moving objects.
In [7], Sacharidis et al, discussed about sub-trajectory clusters. It is the new framework called as partitioning & grouping framework. It is used for clustering of trajectories. For sub-trajectory cluster formation, TRACLUS algorithm is introduced. Main intension of TRACLUS algorithm is to detect sub-trajectories from large trajectory dataset. Sub-trajectory cluster can be defined as the set of clusters moving to similar direction.
T. Brinkhoff discussed about a framework for generating network-based moving objects [12]. It is used to evaluate spatio-based temporal database as many applications dealing with temoral data. The synchrouos movement patterns basically represent the moving objects as they interact with each others. Asynchrounous patterns can be defined as the set of moving objects which moves in the same direction which is known as, unifying trajectory patterns. UT patterns are closed together with the time-stamp, Geo-location or may both. Deer migation is the popular example of trajectory pattern.
As UT patterns useful to learn an interactions between object movement and possibly group dynamics, it is very useful in mining. In[2], flock patterns have been discussed by Patrick Laube and Stephan Imfeld. Flock pattern represents the concurrence motion. It also analyse the REMO i.e Relative Motion. Required trajectory patterns such as, flock, REMO, time relaxed trejectory joins, swarm pattern are also discussed in [2][3][4][5]and [6] respectively. Certain kind of limitations occurred in unifying trajectory pattern mining as user do not have knowledge that which type of trjectory patterns are hidden in the large database of trajectory pattern. Objects are moving arriving at several locations within one-minute interval, one-hour interval and so on therfore, while classifying trajectory patterns rigidity of temporal contraints on the patterns is considered.

An initial pattern discovery processed in two phases first is the sub-trajectory cluster and the other is search space limited to specific sub-trajectory cluster rather than whole dataset. Sub-trajectory clustering utilises hueristic solution based on divide and conquer method. Clustering moving objects is an interesting approach to catch regularities of the moving objects.In[10], the concept of micro-cluster moving is proposed. Theoretic clustering[11] is orgnised by RIC framework. It is implemented using greedy approach to prove goodness measure and efficiency of their propsed approach. RIC is a very flexible framework. It has several desired properties which is not in previous clustering mechanism.

Sub-trajectory clustering includes following phases:

a) Dividing or partitioning phase: In this phase each trajectory is partitioned into set of trajectory partitions or into line segment at the time of object moving in various directions rapidly. To search the set of partitioned point MDL principle [14-15]is utilised.
b) Conquer or groping phase: After partitioning all trajectories, another phase is grouping of obtained line segment into cluster. This phase utilises A density-based clustering method. In[9], density-based clustering method is described which is analogous to DBSCAN. Drill down operations occurred only if MDL i.e minimum description length descreases the

cost. The main purpose of drill down operation is to gain multiple smaller time relaxed patterns from large time relaxed patterns. This operation is automatically selected to minimise the cost of MDL.

Roll-up operations represents the reverse of the drill-down operations. In this operationa merger never decreases the MDL cost.

Lastly, in reasearch[1], author Jae-Gil Lee,J. Han and X. Li. Defined a framework for discovering trajectory patterns. They represent very useful learning interactions between moving objects. There is certain limitation in their defined framework as user don't have idea about type of patterns involved in large dataset.

## III. PROBLEM DEFINITION

"To identify and classify of trajectory patterns form given dataset. As per the tightness temporal constraints there are three types classified into trajectory patterns in which identified trajectories are classified."

## IV. SYSTEM ARCHITECTURE

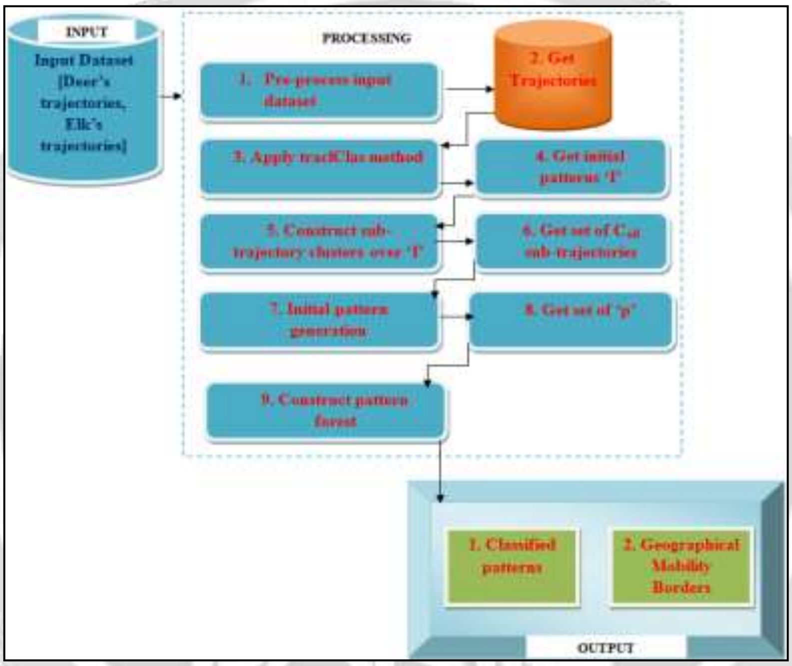


**Fig 1** System Architecture

1. Accept user input dataset i.e. Elk's or Deer trajectories dataset.
2. Pre-process dataset to get trajectories.
3. Apply TrakClas method: To perform sub-trajectory clustering over I based on the partition-and-group framework TrakClas method is used. In sub-trajectory clustering process,trajectory is partitioned into a set of line segments, and these trajectory partitions are grouped into a cluster according to their spatial similarities only.
4. Data compression: Data compression is required to measure the similiarity between line segements. To maximize data compression, a set of UT-patterns should possess two desirable properties: preciseness and conciseness.
5. Pattern forest construction: To construct patern forest apply drill down and roll up strategies of OLAP operation in such way that: -Drill down stratergy is done only if a split decreases the MDL cost same as intial pattern discovery. Its main purpose is to derive multiple time-constrained (or smaller time-relaxed) patterns from a time-relaxed pattern. -Roll up stratergy is the reverse operation of drill-down. Using both stratergies pattern forest is consructed which contains the various patterns.
6. Classification of patterns
7. Identification mobility borders
8. Display result

## V.ALGORITHMIC STRATEGY

1. **Initial Pattern Discovery[1][4][8][15]**

**Input:** A set L of trajectory partitions in a cluster C
**Output:** A set P of initial UT-patterns
**Processing steps:**

1. L1 $\leftarrow$ L, $\vec{R_1} \leftarrow$ DeriveRefMovement(L1);
2. P $\leftarrow$ {($\vec{R_1}$,$L_1$)};
3. Repeat
4. Choose the $m^{th}$ UT-pattern from P,
5. Split the $m^{th}$ UT-pattern into $\overrightarrow{\text{two}}$ splits
6. Choose the pair of trajectory partitions, where
7. $L^p_m \leftarrow Ø$, $L^q_m \leftarrow Ø$,
8. For each $L_i \in$ Lm do
9. If dist($L_i$,Lp)<dist($L_i$, Lq) then
10. $L^p_m \leftarrow L^p_m$ U {$L_i$};
11. Else
12. $L^q_m \leftarrow L^q_m$ U {$L_i$};
13. End if
14. End for
15. Derive reference movement for each split
16. $\vec{R^p_m} \leftarrow$ deriveRefMovement($L^p_m$)
17. $\vec{R^q_m} \leftarrow$ deriveRefMovement($L^q_m$)
18. Replace the $m^{th}$ pattern by new ones.
19. P' $\leftarrow$ P-{($\overrightarrow{Rm}$,Lm)}U{($R^p_m$, $\vec{L^p_m}$), {($R^q_m$, $\vec{L^q_m}$)}
20. Check if L(H)+L(D|H) decreases
21. If MDL(P')<MDL(P) then
22. P $\leftarrow$ P'
23. End if
24. Until MDL(P')>MDL(P)
25. Return the set P of initial UT-patterns
26. Function DeriveRefMovement($L_K$)
27. Consider each t-partition as a candidate
28. $R_k \leftarrow \{L|\forall L \in L_k\}$
29. Find one that minimizes the code length
30. Return $s^{th}$ candidate $R^s_k$, $\overrightarrow{\text{where}}$
31. End function

2. **Pattern Forest Construction[1][8][15]**

**Input**: A set $P_{all}$ of initial UT-patterns
**Output**: A pattern forest FR
**Processing steps:**

1. FR $\leftarrow P_{all}$ , Q $\leftarrow P_{all}$ where, Q is queue
2. Perform Drill-Down operation
3. While Q $\neq$ Ø do
4. Pop a UT-pattern $UT_i$ from Q
5. If $UT_i$ can be easily spilt into $UT^1_i$ and $UT^2_i$ then
6. Push $UT^1_i$ and $UT2_i$ into Q;
7. Update pattern forest
8. Add two vertexes for $UT^1_i$ and $UT^2_i$ into FR;
9. Add two edges for ($UT_i$, $UT^1_i$) and ($UT_i$, $UT^2_i$) into FR;
10. End if

11. End while
12. Perform Roll-up operation
13. $P_c$ is the set of UT-pattern in the $c^{th}$ cluster
14. $P_c \subseteq P_{all}$ do
15. For each pair of $UT_{i,} \in P_c$ and $UT_{j,} \in P_c$ do
16. If $UT_i$ and $UT_j$ merged into $UT_{ij}$ then
17. Add $UT_{ij}$ into $P_c$
18. Update pattern forest
19. Add one vertex for $UT_{ij}$ into FR;
20. Add two edges for ($UT_{ij,}$ $UT_i$) and ($UT_{ij,}$ $UT_j$) into FR.
21. End if
22. End for
23. End for
24. Return the pattern forest FR.

**3. UT Pattern Mining[1][5][8][15]**

**Input:** A set of Trajectories I= {$TR_1$, $TR_2$ …$TR_{numtra}$}
**Output:** A set of UT-patterns O= {UT1, UT2 …$UT_{numpat}$}
**Processing Steps:**

- Phase I-Initial pattern discovery

1. Perform sub-trajectory clustering over I based on the TRACLUS algorithm[8]
2. Get all sub-trajectory clusters $C_{all}$
3. For each $C \in C_{all}$ do
4. /Algorithm 1 */
5. Execute initial pattern generation over C
6. Get set P of UT-patterns as a result
7. Accumulate P into a set $P_{all}$
8. End for
9. Phase II-Granularity Adjustment
10. /Algorithm 2 */
11. Execute pattern forest construction over $P_{all}$
12. Return the set of UT patterns in the forest
13. Classify pattern into three type's i.e. time-constrained pattern, time-relaxed patterns and time independent patterns.

## VI. EXPERIMENTAL SET UP

A user friendly desktop application is developed for determination of trajectory patterns and its visual representation is also planned. The system will be designed using JAVA platform (Jdk1.7). For backend purpose Mysql 5.3 is used to manage local user information in database. Machine with Core i3 processor with OS having 4GB ram is used for development and testing. Eclipse indigo is used as an IDE using which code is develop.

**Dataset used: Dataset used:** are used to analyze the trajectory patterns. Deer trajectories and Elk's trajectories is considered. Starkey Project makes this dataset available which is having radio-telemetry locations of animals from north eastern Oregon.

A] Deer Dataset:

- This dataset is having 32 trajectories and 20,065 points.

B] Elk Dataset:

- This dataset is having 33 trajectories and 47,204 points.

## VII.MATHEMATICAL MODEL

Let S be UT-pattern mining system such that,
S = {I, F, O}

| | |
|---|---|
| **I: {I1, I2, I3 } , input to the system** | I1= User details for registration |
| | I2= User Login |
| | I3= Dataset of trajectories i.e. {TR1; ... ; $TR_{numtra}$ } |
| **F: {F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17} , functions of system** | F1= User registration |
| | F2= User login |
| | F3=Upload dataset |
| | F4=Create vector |
| | F5=Define line segments |
| | F6=Discover average direction |
| | F7=Search projection point |
| | F8=Apply data compression |
| | F9=Extract reference movement |
| | F10=Pattern splitting |
| | F11=Initial pattern creation $P_{all} \in P$ |
| | F12=Apply drill down strategy |
| | F13= Apply roll up strategy |
| | F14=Construct pattern forest |
| | F15= Identify mobility borders |
| | F16=Pattern classification |
| | F17=Display result |
| **O :{ O1, O2, O3, O4, O5}, Output of a system** | O1=Initial clusters |
| | O2=Initial pattern P |
| | O3=Pattern Forest |
| | O4=Mobility borders |
| | O5=Classified UT patterns |

## VIII. RESULT TABLE AND DISCUSSION

Figure 2, 4, 6 represents sample of cattle, deer and elk dataset respectively. It contains cluster ID, type of trajectories such as, time-constrained or time-independent and points number i.e. X and Y co-ordinates. Each point consists of a real coordinate (location) and a timestamp. Whereas, Figure 3, 5 and 7 represents UT-patterns for the cattle, deer and elk datasets respectively. In this red line indicates the clustering, blue line indicates clustered mobility border and boundry line indicates the dense mobility reason.

```
ClusterID: 0
Type :Time-independent
Points Number:  2
555.0 635.0   417.8664808793932 563.2618361975537

ClusterID: 1
Type :Time-independent
Points Number: 4
444.0 634.0   417.8817183493784 605.5508441778338
384.81090072460364 550.3516322911275   341.81329158293215
479.93664999766315
```

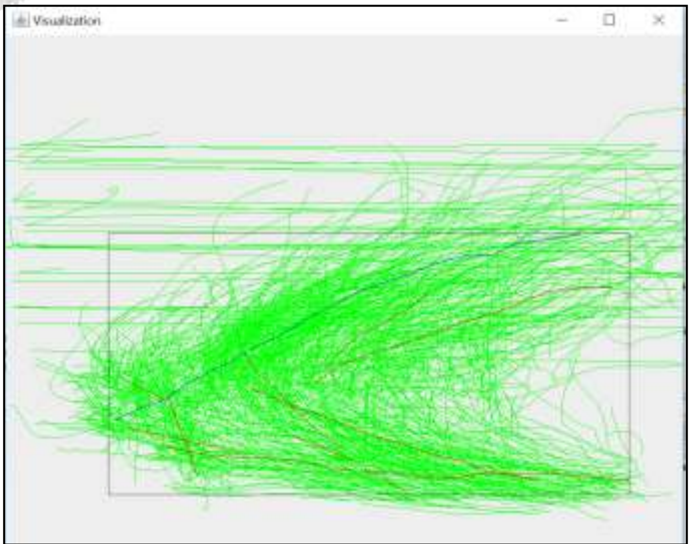

**Fig 2**: Sample of cattle dataset

```
ClusterID: 1
Type :Time-constrained
Points Number: 11
732.8000000000001 627.1   755.1023245950289 564.6399179637402
773.4071082159468 521.3800645149495   808.9218151762342
502.2954730808634   841.7530329312752 483.0042856762092
869.9905320699811 457.29719257220654   894.9415376883478
431.49138092224626   915.1784425350729 399.31850131583246
934.4804666458747 354.1274250126442   930.5849033222265
293.46339481330017   949.9936464082418 230.8026320511861

ClusterID: 2
Type :Time-constrained
Points Number: 2
858.8 365.4   917.4920150708067 385.4480213070027
```

**Fig 3**: Sample UT-pattern for the cattle dataset

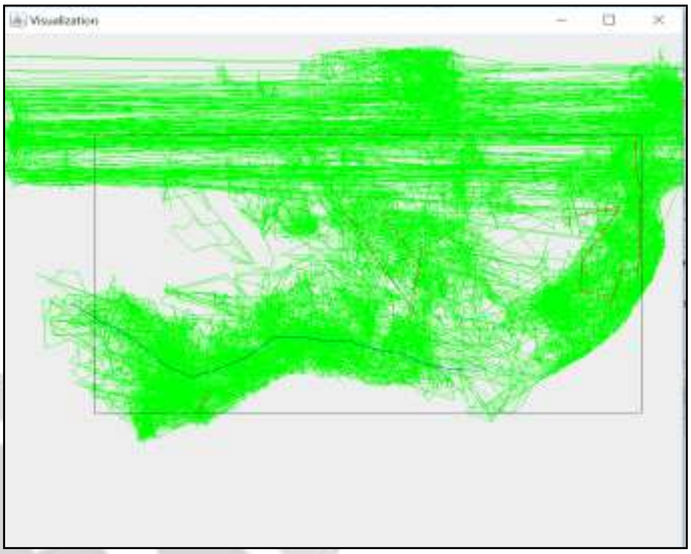

**Fig 4**: Sample of deer dataset

```
ClusterID: 0
Type :Time-constrained
Points Number: 2
772.522156446508 79.32301060537361   772.918760589134
159.95664295050548

ClusterID: 1
Type :Time-constrained
Points Number: 24
32.099999999999994 338.6   49.99690217016408 297.83967931344637
71.2171941489708 259.8837815094218   103.40032789148464
230.69032076434948   138.8931257267478 195.2817554878429
164.8361002785565 168.4344535636706   192.48345581039462
141.4611196297331   225.26786858228078 121.8898432498137
264.47054093711773 113.3207373712495   301.69137189808635
101.11484811311423   336.8943980832231 88.3835954755229
375.3375511669243 78.73138207463361   414.78615308064525
72.47858288973654   465.6151578942473 63.89278838923062
501.48205125164975 51.615129897432695   556.277216679209
43.569237170213796   600.9674968518694 37.11471479116858
645.0138557239349 37.60444353941307   687.9358144360666
33.018791881830055   735.7615301595196 39.030017502495014
799.3116176171889 50.54699274860474   892.5400373880055
84.31118112483978   1072.0479310554713 67.5445570609495
1132.9450726929754 37.71840702180532
```

**Fig 5**: Sample UT-pattern for the deer dataset

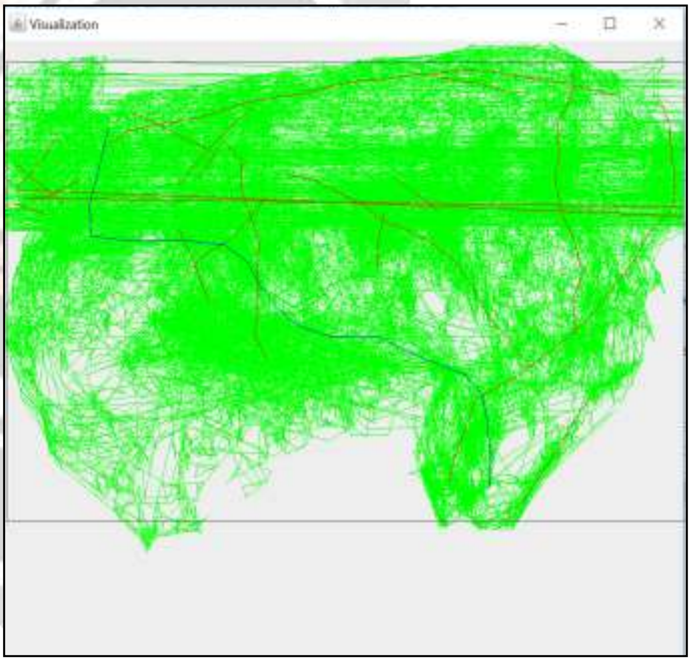

**Fig 6:** Sample of elk dataset

**Fig 7**: Sample UT-pattern for the elk dataset

## IX. CONCLUSION

UT-pattern mining framework is proposed to mine UT patterns from given input dataset. We proposed UT-pattern mining algorithm to detect trajectory patterns of various temporal tightness such as, time-constrained, time relaxed and time independent. Algorithm contains three phases such as, initial pattern discovery, pattern forest construction and pattern classification. For cluster formation Traclus method is implemented and for pattern forest construction drill down and roll up approach is used. Furthermore, system is mobility border identification. A general method is used to determine the influence of social and mobility behavior over a specific geographical area in order to evaluate to what extent the current administrative borders represent the real basin of human movement. Experiments using real-world data sets show that UT-Pattern Mine can easily discovers various types of trajectory patterns.

.

## REFERENCES

[1] J. Lee, J. Han and Xiaolei Li ,"A Unifying Framework of Mining Trajectory Patterns of Various Temporal Tightness", IEEE transaction on knowledge and data mining vol,27, No.6, june 2015.

[2] P. Laube and S. Imfeld, "Analyzing relative motion within groups of trackable moving point objects," in Proc. 2nd Int. Conf. Geograph. Inf. Sci., Boulder, CO, USA, Sep. 2002, pp. 132–144.

[3] P. Laube, M. J. van Kreveld, and S. Imfeld, "Finding REMO— Detecting relative motion patterns in geospatial lifelines," in Proc. 11th Int. Symp. Spatial Data Handling, Leicester, U.K., Aug. 2004, pp. 201–214.

[4] P. Laube, S. Imfeld, and R. Weibel, "Discovering relative motion patterns in groups of moving point objects," Int. J. Geograph. Inf. Sci., vol. 19, no. 6, pp. 639–668, Jul. 2005.

[5] M. Benkert, J. Gudmundsson, F. Hubner, and T. Wolle, "Reporting flock patterns," in Proc. 14th Eur. Symp. Algorithms, Zurich, Switzerland, Mar. 2006, pp. 660–671.

[6] P. Bakalov, M. Hadjieleftheriou, and V. J. Tsotras, "Time relaxed spatiotemporal trajectory joins," in Proc. 13th ACM Int. Symp. Geograph. Inf. Syst., Bremen, Germany, Nov. 2005, pp. 182–191.

[7] Sacharidis, K. Patroumpas, M. Terrovitis, V. Kantere, M. Potamias, K. Mouratidis, and T. K. Sellis, "On-line discovery of hot motion paths," in Proc. 11th Int. Conf. Extending Database Technol., Nantes, France, Mar. 2008, pp. 392–403.

[8] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in Proc. ACM SIGMOD Int. Conf. Manag. Data, Beijing, China, Jun. 2007, pp. 593–604.

[9] J. Gudmundsson and M. J. van Kreveld, "Computing longest duration flocks in trajectory data," in Proc. 14th ACM Int. Symp. Geograph. Inf.. Syst., Arlington, VA, USA, Nov. 2006, pp. 35–42.

[10] Y. Li, J. Han, and J. Yang, "Clustering moving objects," in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Seattle, WA, USA, Aug. 2004, pp. 617–622.

[11] C. Bohm, C. Faloutsos, J.-Y. Pan, and C. Plant, "Robust informa- € tion-theoretic clustering," in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Philadelphia, PA, USA, Aug. 2006, pp. 65–75

[12] T. Brinkhoff, "A framework for generating network-based moving objects," GeoInformatica, vol. 6, no. 2, pp. 153–180, Jun. 2002.

[13] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," J. Intell. Inf. Syst., vol. 27, no. 3, pp. 267– 289, Nov. 2006.

[14] T. C. M. Lee, "An introduction to coding theory and the two-part minimum description length principle," Int. Stat. Rev., vol. 69, no. 2, pp. 169–183, Aug. 2001.

[15] H. Bischof, A. Leonardis, and A. Selb, "MDL principle for robust vector quantization," Pattern Anal. Appl., vol. 2, no. 1, pp. 59–72, Apr. 1999.

[16] http://www.fs.fed.us/pnw/starkey/data/tables