An Efficient Trajectory Representative Generation Moving Web-Based Data Prediction Using Different Clustering Algorithms

Vishnu Kumar Mishra, Shri Shankaracharya Institute of Engineering and Technology, India

Megha Mishra, Shri Shankaracharya Institute of Engineering and Technology, India & Chhattisgarh Swami Vivekanand Technical University, India

Bhupesh Kumar Dewangan, O.P. Jindal University, Raigarh, India*

D https://orcid.org/0000-0001-8116-7563

Tanupriya Choudhury, University of Petroleum and Energy Studies, India

ABSTRACT

This paper highlighted moving and trajectory object cluster (MOTRACLUS) algorithm and analyzed the sub-trajectories and real-trajectories algorithm for moving web-based data and suggested a new approach of moving elements. This paper evaluates the hurricane data measure and mass less data measure entropy of trajectories objects of moving data of Chhattisgarh location. The paper covered prediction generation with their distance cluster minimum description length (MDL) algorithm and other corresponding distance cluster (CLSTR) algorithm. This paper highlighted the k-nearest algorithm with least cluster section (LCSS) model and dimensional Euclidean of MDL algorithm. The algorithm consists of two parts, that is, partitioning and grouping phase. This paper develops and enhances a cluster of trajectory objects and calculates the actual distance of moving objects. This algorithm works on the CLSTR algorithm and calculates trajectory movement of the object. In this, the authors evaluate the entropy of moving objects by consideration of the heuristic parameter.

KEYWORDS

Density-Based Cluster, Framework Grouped, Partition and Group Work, Trajectory Clustering, Web-Mining

INTRODUCTION

This papers we inspect the problem of discover similar trajectories of moving web-based objects. The trajectory of a moving web-based object is typically modeled as a sequence of consecutive locations in a multidimensional (generally two or three dimensional) Euclidean space. We are facing an unprecedented proliferation of mobile devices, many equipped with positional technologies such as Global Positioning System (GPS).

DOI: 10.4018/IJISMD.316132

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Moreover, the accessibility as well as the wide-ranging collection of applications these knowledge apparatus, has maintained their inclusive dispersion of everyday life, facilitate the frequent aggregation of moving web-based objects data into large wide-ranging datasets. These promising datasets are stressing the implication of supervision and organize cluster of representative data. Technique used for remove information from database or cluster set data is called trajectory data mining.

The current developments in peripatetic database need the different electronic device like GPS system, Cellular mobile system and sensor technology and different tools for awareness, also perform very vital role for data analysis for performing various task related to data. For real life example, the mining computing the users equipped with movable devices that move in everywhere space and provide the record, which registered with device location at different time moments via some movable temporal databases.

RELATED WORK

In this Section, we study previous work based on two major discovery techniques, Markov chain models and spatiotemporal data mining, for extracting movement patterns of an object from historical trajectories (Zhegown, 2020) (Wang et.al, 2020) (Wang, 2020). Markov chain models have been widely used in order to estimate the probability of an object's movements from one region or state to another at next time period. Ishikawa et al. derive the Markov transition probabilities between cells from indexed trajectories (Kuliko, 2017) (Mehrotra and K. Chakrabarti, 2018) (Shende, M. P., 2012).

In their further study (Zheng, 2020) (Yuan et al., 2019), a special type of histogram, called mobility histogram, is used to describe mobility statistics based on the Markov chain model. They also represent the histogram as cube-like logical structures and support an online analytical processing (OLAP)-style analysis. Authors (Wang et al., 2018) classify an object's mobility patterns into three states (stationary state, linear movement, and random movement) and apply Markov transition probabilities to explain a movement changes one state to another consider the location tracking problem in PCS networks (Wang et al., 2016) (Sozio et al., 2018) (Al-Sharif et al., 2017) (Venkatadri, M., & Pasricha, A., 2019). All studies are based on the same Markov process in order to describe users' movements from one or multiple Personal Communications Services (PCS) cells to another cell. However, they have different ways to model users' motilities using Morkov models, thus, show distinct results to each other (Lander et al., 2018) (Yang et al, 2017).

Data mining play important role to discover the knowledge and some time it is also called knowledge discovery, sometime this technique helps to extracting the data from large database (Han et al., 2016) (Kuliko, 2017). Data mining clustering approach generate answer and classifying movement analysis of data by using some clustering and prediction algorithm (Lander et al., 2018) (Wu, et al., 2017).

Data mining only a tool and framework which help to mine temporal and pictorials trajectory moving web-based data. Through the classified trajectory moving web-based data can be represent by cubes and relational transaction approach (Wang et al., 2018) (Mishra et al., 2017). The survey of instance of data mining methods based on classical relational and transactional data can be found in traditional clustering trajectory movement. Based on that trajectory mining cluster many researchers suggest the many technologies Global Navigation Satellite System (GNSS) and Radio frequency identification (RFID) (Wang et al., 2016).

The trajectory data mining generation is analysis tasks based, suggest visualization of Twodimension (2D) and Three-dimension (3D) data movement. The trajectory mining moving webbased data classify three domains: these are operational, dynamic and social trajectory motion of data (Kuijpers and Othman, 2007). This survey represents different algorithm efficient clustering generation with respective density based clustering algorithm (Jacinto et al., 2010) (Pasricha, A., 2016).

The trajectory data and Clustering are the generally measured as the field of data mining. In data mining classification and Clustering play main role. The Clustering algorithms can be divided

into four different types (Tseng et al., 2013): The first method generally based on density monitoring method that real life example Density-based spatial clustering of applications with noise (DBSCAN) and other example is ordering points to identify the clustering structure (OPTICS) algorithm. The second type is grid-based algorithm; the example of this algorithm is Statistical Information Grid (STING) (Chen et al., 2014). The third algorithm *k*-means which is example of partitioning algorithm and fourth is hierarchical algorithm, one of the best examples of hierarchical algorithm is Balanced iterative reducing and clustering using hierarchies (BIRCH) algorithm. Suggested algorithm Trajectory clustering (TRACLUS) follow the rule group of density-based methods (Kuliko et al., 2014).

Figure 1 Shows probability of trajectory data movement of different spot with respective time span. The different spot colors represent actual predictive generation of object. The spot black and brown represent the maximum data movement of object and center of attention the same cluster generation. The dataset DBM based clusters algorithm is provinces of high density disjointed by provinces of low-density algorithm that is example of k-means (Weibel et al., 2016) (Baghel, S. K., et al., 2012). It is also observed that density-based clustering algorithm a part of DBSCAN algorithm. As mainstream preceding investigation has take up be partitioning and group of apex statistics. Trajectory representative generation moving web-based data prediction has maximum and given projected data clustering suggested by (Guo et al., 2016) (Shende, P., 2013).

Figure 2 corresponds to the overall similar trajectory data object movement at density based clustering at pier 500 kilometer range with respect to latitude and longitude. This latitude and longitude help for actual movement of object (Dewangan B.K. et al., 2020) in each direction with classify distance 500 per kilometer. This emphasis the modification of the representative generation moving probabilistic modeling algorithm monthly basis i.e. is called TRACLUS algorithm. Properly, the probability calculation based on representative generation prediction.

ALGORITHM CRITERIA

The CLSTR Algorithm

The TRACLUS trajectory clustering algorithm contains the three different phases. These are trajectory planning (line 2), Line Segments Clustering (line 4) and Trajectory generations. These phases execute in sequentially with minimum trajectory movement i.e. the first part we calculate the



Figure 1. Periodic data movement pattern generation

Figure 2. Trajectory data monthly generation prediction (movement with time): a. Dandkaran area Hurricane data measure; b. Dandkaran area Massless data measure



partitioning and then second phase we execute the trajectory clustering. We comprehensive clarify these algorithms below.

Algorithm Generation CLSTR (Clustering Trajectory)

```
Input: Find trajectories generation data T = \{G_1, \dots, G_{nmn}\}
Output: (1) set prediction data data R = \{S_1, \dots, Snun_{obs}\}
        (2) A S={set of trajectories moving set of web-based data}
/* CLSTR clustering prediction */
01: \forall (G \in T) recursive until termination ;
02:
       Predictive generation Trajectory clustering;
          find Ls and N\varepsilon (L) ;
         Accumulate Ls into a set E;
03:
/* Grouping Phase */
04:
           Execute Line Segment Clustering for E;
             Find value of R clusters ;
05:
           search belong (S \in R) do;
06: Repeat Predictive generation Trajectory clustering;;
```

Partitioning Trajectory (Partitioning Phase)

This portion covers the *Approximate Trajectory Partitioning* algorithm for trajectory clustering for data. The algorithm *Approximate Trajectory Partitioning* shows below. This work computed MDL_{nopar} and MDL_{par} for respectively argument in a trajectory. Compare actual value of each MDL_{par} and MDL_{nopar} , out of them find out maximum value then put in the instantly value to MDL_{par} (pmdlstaIndex, pmdlcurrentIndex). Set the current appxomately value in stating and repeat CP₁ as startmdlP1.

Algorithm Prediction generation Trajectory MDL

```
Input: A prediction generation PG, = g1g2....gn ...ppgin,
Output: Prediction generate upto last Cp ={ C1, c2..C ppgin, }
/* stating: Prediciton Generation */
    Initialize the value CP<sub>i</sub> in srt P1,p2, ..;
01:
02: set Indexvaluestr:= 1, maximumlength:= MDL<sub>par</sub> miminum= Pcurr+currIndex;
03:
     for (maxIndex start+ maximunvalue £ currentIndxggu); repeat
      unitl termination
04:
         maxcuurrvalue= pccgggstart + current start;
05:
         max<sub>par</sub>:= CDMDL total <sub>par</sub>(maxcuur srart, max curr index);
06: min_nopar:= CDMDLtotal_nopar (maxcuur sart, max curr index);
/*Find max and min patterned the separating existing argument
that generation as CDMDL totalnonpa */
07:
    CDMDL total par= pccgggstart -1
/* Prediciton generation of previous point */
08:
      Average= pccgggstart -1; fing generation each;
                   pccgggstart:= pcccurrIndexst-1, maxlength:=1;
09:
10:
              recursive
11:
              total max:= pccgggstart-1;
      Assign the value plen, i nto Cp /* the last */
12:
```

Clustering Trajectory (Grouping Phase)

This portion, we recommend a different line algorithm (Dewangan B.K. et al., 2019) especially emphasis on line *segment clustering algorithm* for moving web-based data trajectory clustering. We also focus on trajectory density prediction generation algorithm for trajectory clustering (Agarwal A et al., 2019). Suppose a there are segmentation of line belong to S and produces a set of clusters that is O. For defining the cluster set parameter we need *index value cluster Id* two data set i.e. is $S_{\varepsilon}(L)$ nd Lnr₁,..., L_{nmcls} . These approaches suggest clustering trajectory algorithm provides different appearances of algorithm DBSCAN with many parts in trajectory clustering.

Algorithm: K- nearest prediction generation density Clustering

```
(1) K-nearest Line prediction generation segments D = \{G_1, \dots, G_{nmln}\},\
Input:
         (2) set K-nearest prediction S_{\varepsilon}(L) nd \operatorname{Lnr}_{1}, \cdots, L_{nmcls}
Output: K-nearest Line prediction generation segments S = \{s_1, \dots, s_{nmcls}\}.
/* prediction starting point generation */
01: Assign generation cluster Id to be -1;
02: Identify all unbrokable cline and setcluster Id value;
03:
     iterative belong (L \in D) until
04:
            go (S each unbroken Id) jump
05:
                   assign S_{g}(D);
06:
                   put (|S_{a}(D)| \geq GNmax) repeat
                          provide sum of rS \{A\} \in Sr(D) on Id;
07:
                    Addition N_{g}(I) - \{A\};
08:
/* 2 Step prediction generation*/
09:
                 assign cluster Id (S, cluster Id, r, MLmax);
10:
                      put cluster Id as GNmax; /* generate new id */
11:
                   do;
12:
                            drop unstasidied generation (remark);
```

International Journal of Information System Modeling and Design

Volume 13 · Issue 7

```
/* 3 Step prediction generation */
13: Assign r \ Gs \in D, S_{clusterID};
/* Cluster predication trajectory */
14: if (D \in G) till
/* from stating check clusterid*/
            for (| KCN (G) | > MLmax)
15:
                  check unsatisfied value and remove from cluder id;
16:
17: Increase value(S, D, G, MLins) {
18:
            for each (D=q) till
19:
                  each pass value assign in Sg;
20:
                  add G \alpha (S);
                  for \{A\} \in S_r(D) on Id) do
21:
                          ITERATION (NS\alpha (g\ D) reperat
22:
23:
                           set (G is actual cluster value) print
24:
                                       value of cluster Id to G;
                                          set (S is removed cluster id) print
25:
                                       value of cluster Id to D;
26:
27:
            Stop removing value from G to value p1,p2...Pn;
                           Stop prediction generation from S;
28:
29: }
```

Algorithm: DBSCAN Trajectory partitioning clustering

```
Input: (1) Create new cluster generation predication P,
       (2) Find MLins value for cluster generation
       (3) Set prediction \varphi.
Output: Set demonstrative trajectory T, prediction generation for maximum P,
01: Find max value of direction vector field ec{v} ;
02:
      Alternate the hatchets X axis is equivalent to \vec{v};
03:
      Now set starting and ending value for cluster P_i;
/* coordinate of the c axis donation */
04:
      average value G with new generation p'-values;
05:
          if (T \in G) recursive
/* calculate all cluster predicition_ */
06:
          sum mov<sub>p</sub> segments that contain the p'- value of the predication r;
         if (mov_n \ge pLincuus) then
07:
         rel in p'- values between t and its immediately previous point f;
08:
09:
         put (max \geq r) get
10:
         total max generation movavgC_;
11:
         get max and min value mov,;
12:
         Append avg to the end of RTR;;
```

Formalization Using the MDL Principle

This area suggests a formalization trajectory prediction of optimum trade between precision versus terseness. This portion also emphasis the accept the minimum prediction generation description density length (MDL) standard extensively.

The MDL classify into two different parts, these are G(P) and L(GIT) where T emphasis trajectory movement and G belongs maximum prediction data. There binary apparatuses are casually stated as follows [if (diff³ α)]: we formulate G(P) by Formula (1). Here, len (pcj+1) denotes the length of a line segment pcj+1, i.e., the Euclidean distance between pcj and pcj+1. Hence, L(H) signifies the amount of the distance of all trajectory partitions:

$$L(H) = \sum_{j=1}^{par i^{-1}} \log_2\left(len\left(\mathbf{p}_{cj}\mathbf{p}_{cj+1}\right)\right)$$
(1)

Approximate Solution

The algorithm Approximate Trajectory Partitioning shows below. Here we calculate GTR for cluster prediction of moving web-based data. Here value of CLCSTR_{par} is assign maximum trajectory area i.e. belong $\{A\} \in S_r(D)$ on Id _{nopar}, assign actual data with its adjacent value of moving web-based data, for every point $\{A\} \in S_r(D)$ on Id . Then, we recurrence generation data practice for approximate solution i.e. start with initial point (startInex: = currIndx-1, length: = 1). Representative CLCSTR_{par} present provides best generation approximation clustering.

Algorithm: New Representative CLCSTR_{par present} Generation

```
(1) Consider PLT_i as CLCSTR_{par present}
Input:
                                                 ={c1,c2...Cnindexstart);
         (2) MLins (3) A smoothing parameter \alpha
Output: The demonstrative PLR_i as CLCSTR_{par present} = \{p1, p2...pnindexstart\};
01: Find max value of direction vector field ec{v} ;
02:
      Replace the hatchets X axis is equivalent to ec{v} ;
      set value CLCSTR<sub>par present</sub> =currentindex;
03:
      iterative belong (L \in D) until;
04:
05:
            if (G \in r) repeat
      Let num be the number of the line segments that contain the
06:
      X'- value of the point p;
          if (num_p \ge G MLins) then
07:
08:
          calu difference in X' - values between p and its nearest point;
          if (diff \geq \alpha) then
09:
         Compute the average coordinate avg '_;
10:
11:
         Unwrap the spin and find out avg_{p};
12:
          Append avg to the end of RTR;;
```

Clustering Neighbors K-Clustering (NK-CN)

```
The set od points (p, int k, real num \alpha)
// where \alpha is greater than zero (\alpha>0).
BEGIN
p = Find out core value P (int p, k, \alpha);
if pointp <> null then
        Cul 1 set = Set1 (int p, k, \alpha) set at initial point;
        Clu1 = Set and GetCluId1 ();
         C1 = Set and GetInitial valueClu1 (for p, tp, Set1, k, cluID1);
        Release Cluster1 (p, C1, Core1, \alpha,K);
Start clusterting;
D = D \cup nk-distance (point) (point);
CoreSet1 = Coreset1 (object); Release clur1 (Set p, C1,Set1, int k, float \alpha)
BEGIN
        Seedset1 = CoreSet1;
        while not SeedSet1.empty() DO
         Pointp = GetOutPointp1 (SeedSet1);
```

```
CoreSet1(object1) = D

SeedSet1 = SeedSet1 ∪ (object);

CoreSet1 = Coreset1 ∪ (object);

terminating;

close each iteration;

D =D ∪ nk-distance (point) (point);

terminate loop;

end Expandclus1.
```

The *TRCLS* trajectory clustering algorithm contains the three different phases. These are trajectory planning (line 2), Line Segments Clustering (line 4) and Trajectory generations. These phases execute in sequentially with minimum trajectory movement i.e. the first part we calculate the partitioning (Dewangan B.K. et al., 2020) and then second phase we execute the trajectory clustering. We comprehensive clarify these algorithms below.

Algorithm Prediction generation Trajectory MDL

```
Input: A prediction generation PG, = g1g2....gn ...ppgin,
Output: Prediction generate upto last Cp ={ C1,c2..... C ppgin, }
/* stating: Prediciton Generation */
        Initialize the value {\rm CP}_{_{\rm i}} in srt P1,p2, ..;
01:
        set Indexvaluestr:= 1, maximumlength:= MDL_{max} miminum= Pcurr+currIndex;
02:
        for (maxIndex start+ maximunvalue \leq currentIndxggu);
03:
        repeat unitl termination
04:
        maxcuurrvalue= pccgggstart + current start;
05:
        max<sub>par</sub>:= CDMDL total <sub>par</sub>(maxcuur srart, max curr index);
06: min_nopar: = CDMDLtotal_nopar (maxcuur sart, max curr index);
/*Find max and min patterned the separating existing argument
that generation as CDMDL totalnonpa */
07:
        CDMDL total <sub>par=</sub> pccgggstart -1
/* Prediciton generation of previous point */
08:
        Average= pccgggstart -1; fing generation each;
09:
        pccgggstart:= pcccurrIndexst-1, maxlength:=1;
10:
        recursive
11:
       total max:= pccgggstart-1;
        Assign the value plen, i nto Cp /* the last */
12:
Algorithm: DBSCAN Trajectory partitioning clustering
Input:
         (1) Create new cluster generation predication P.
         (2) Find MLins value for cluster generation
         (3) Set prediction \varphi.
Output: Set demonstrative trajectory T_i prediction generation for maximum P_i.
```

```
01: Find max value of direction vector field \vec{v} ;
```

```
02: Alternate the hatchets X axis is equivalent to \vec{v};
```

```
03: Now set starting and ending value for cluster P_i;
```

```
/* coordinate of the c´ axis donation */
04: average value G with new generation p´-values;
```

```
05: if (T \in G) recursive
```

/*	calculate all cluster predicition */
06:	: sum mov segments that contain the p - value of the predication $r;$
07:	: if (mov ≥ pLincuus) then
08:	rel in p' -values between t and its immediately previous point f;
09:	put (max ≥ r) get
10:	total max generation <i>movavgC_n;</i>
11:	give max and min value mov,;
12:	Append avg_p to the end of \tilde{RTR}_i ;

EXPERIMENTAL EVALUATION

The experiment performed using the approximation algorithm. The main aim of our research work is to compare the result which we obtain by using CLSTR algorithm and the previous traditional approaches. The following points found in Tables 1-3 are considerably important for performing experimental work.

The distance vector selected for Euclidean value will always remain unchanged throughout the research work. The comparison (Dewangan B.K. et al., 2018) is performed foe each and every pair of sequence value taken and tried out to uncover the minimum possible gap between the paired values. The value accepted for the minimum distance vectors. Table 1 shows the Correspondence values of Prediction Generation with Probability, Table 2 shows Correspondence values of distance cluster MDLdis, and Table 3 shows the Correspondence distance cluster CLSTRalgo.

	Prediction	Algorithm							
Distance Vector (Km)	generation / probability	MDL	SCAN	DBSCAN	CLCSTR _{par present}	NK-CN	K- nearest		
500	4.05/0.08	0.42	0.736	0.0043	0.00490	23.905	0.454		
1000	6.01/.14	0.506	0.673	0.00341	0.0042	18.907	0.876		
1500	5.04/0.06	0.358	0.631	0.0026	0.0042	45.452	0.0048		
2000	5.08/0.12	0.323	0.588	0.00271	0.00271	42.454	0.451		
2500	4.02/0.15	0.617	0.453	0.0032	0.0032	94.256	0.349		
3000	4.06/0.03	0.806	0.5935	0.0043	0.0043	93.245	8.651		
3500	5.01/0.54	0.7	0.7	0.323	0.588	0.7	0.67		

Table 1. Correspondence values of Prediction Generation with Probability

Table 2. Correspondence values of distance cluster MDLdis

Con	npare	Actual time	Representative generation					
	MI	DLdis	5	15	20	26		
5	0.45	0.42	0.42	0.736	0.0043	0.00490		
5	0.6	0.876	0.506	0.673	0.00341	0.0042		
3	0.4	0.631	0.358	0.631	0.0026	0.0042		
3	0.6	0.588	0.323	0.588	0.00271	0.00271		
6	0.45	0.42	0.736	0.0043	0.00490	0.0032		
6	0.7	0.876	0.806	0.5935	0.0043	0.0043		

			Cluster generation time cycle in (sec)					
Com	pare	Actual time	Representative generation					
CLSTR algo			5	10	26	52		
4	0.67	0.67	0.358	0.631	0.67	0.0043		
4	0.7	0.876	0.806	0.5935	0.0043	0.0043		
3	0.4	0.631	0.5935	0.0043	0.0043	0.7		
3	0.6	0.00271	0.00271	0.588	0.00271	0.00271		
5	0.45	0.00490	0.0032	0.0043	0.00490	0.0032		
5	0.7	0.67	0.806	0.5935	0.0043	0.0043		

Table 3. Correspondence distance cluster CLSTRalgo

Next step we applied for DTW and applied the CLSTR algorithm for finding the more feasible values of co-ordinates. Distance vector is calculated for the above mentioned two traditional approaches but it seems to take the optimized value of k means vector. The other hand in CLSTR algorithm no needs to manually optimize the parameter it automatically selects the best suited one. The reason behind this is that it compares each and every possible sequence of pairs and there distances also best algorithm. Table 4 shows the correspondence distance best algorithm for values of Prediction Generation with Probability.

The value selection of ω is very important for representation of LCSS. The experiment is approximately performed 15 times on an average to find out the most suitable clustering sequence with and without CLSTR algorithm and then the result is compared. CLSTR algorithm need not require any kind of sampling. Table 5 shows the correspondence values and running times cycle between two sequences from our ANIMALS dataset.

RESULT AND DISCUSSION

The main objective of this work is to carry out the analytical discussion of data and also watching the entropy particulars of it. The work emphasizes the capacity of object when it is on movable condition and its actual efficiency to carry out the k-measure capacity. The value of φ we obtain from above work we can see that is 0.43 which is the lowest one observed during the analysis part. The Time complexity of our MOTRACLUS algorithm is O(n) as liner. Prediction generation Trajectory

Distance	Prediction	Algorithm (distance density clustering)							
Vector (Km)	Generation / probability	MDL	SCAN	DBSCAN	CLCSTR _{par present}	NK-CN	K- nearest		
500	4.05/0.08	yes	yes	yes	yes	yes	no		
1000	6.01/.14	yes	yes	no	yes	no	no		
1500	5.04/0.06	yes	yes	yes	yes	yes	no		
2000	5.08/0.12	yes	no	yes	yes	yes	no		
2500	4.02/0.15	yes	no	yes	yes	yes	yes		
3000	4.06/0.03	yes	no	yes	yes	yes	yes		
3500	5.01/0.54	yes	yes	no	yes	no	no		

Table 4. Correspondence distance best algorithm for values of Prediction Generation with Probability

	Correspondence							Runr	ning time cycl	e in (sec)	
ω	φ	Actual time	No of tries movement						No of trie	s movement	
			5	10	26	52		5	10	26	52
4	0.43	0.319	0.42	0.736	0.0043	0.00490	23.905	0.454	0.617	0.453	0.0032
4	0.7	0.451	0.506	0.673	0.00341	0.0042	18.907	0.876	0.806	0.5935	0.0043
6	0.67	0.349	0.358	0.631	0.0026	0.0042	45.452	0.0048	0.0032	0.0032	0.00491
6	0.7	8.651	0.323	0.588	0.00271	0.00271	0.7	0.451	0.506	0.673	0.00341
7	0.45	0.454	0.617	0.453	0.0032	0.0032	94.256	0.349	0.358	0.631	0.0026
7	0.6	0.876	0.806	0.5935	0.0043	0.0043	93.245	8.651	0.323	0.588	0.00271
5	0.7	0.451	0.506	0.673	0.00341	0.7	0.451	0.506	0.673	0.00341	0.0042
5	0.45	0.454	0.358	0.471	0.00341	0.0042	0.358	0.631	0.5935	0.0043	0.0032

Table 5. Correspondence values and running times cycle between two sequences from our ANIMALS dataset

MDL and K- nearest prediction generation density Clustering is $O(n \log n)$ - Linearithmic algorithm while DBSCAN Trajectory partitioning clustering and DBSCAN Trajectory partitioning clustering $O(n^2)$ – linear Quadratic time.

From the Figure 3, we can observe that the value which we have calculated for $\varphi = 41$ and $\omega = approx(15)$ not showing much similarity and also represent the different aspect when we will increase the quality value of $\omega = approx(15)$. Also we can see that if we will be taking the very small values of k it will give the better result as compared to large volume values. The range which is in between the values of 37 and 43 represent the most optimistic result of T φ .

Figure 4 shows massless data measures. The on an average value we got for above analysis is $AVG[\omega(\phi)] = 7.26$ when we set the values as $\Omega = 0.43$ and $\omega = approx(15)$. So from above calculation the calculated value we achieved using the best suited parameter values are $\phi = 41$ which is the most significant value obtained during the analysis phase and also matching with the value which we want to carry out for this work.

Figure 5 and Figure 6 show hurricane data measure entropy and massless data measure entropy. The graph also represents the clustering outcome value which is $\omega = 11$ and it is not possible for all k points that clusters value should be 10. So from the above calculation and observation we can say that it is the most optimistic value which we got from the calculation and it is always good enough and produce the most quality clustering output for always $\omega = 11$. It also the close observation from above graph and calculation that no other value of ω can produce such a result which we got from above analysis.

Figure 3. Hurricane data measure



Figure 4. Massless data measure



Figure 5. Hurricane data measure entropy

Figure 6. Massless data measure entropy

CONCLUSION

This research work provides a way of 41 variations which we have used during the research work to form a more novel and optimistic approach for clustering configuration. The work which is carried out using different parameters and moving towards the most optimistic approach is based on the algorithm which we have used here CLSTR. The algorithm CLSTR is the best one which is designed over here taking the different value of parameters and choosing the most optimistic one. The work carried out is started with the division segments of different lines and parameters and then grouping

the parameters on the basis of similar point values. The main objective and advantage of the designed algorithms CLSTR is to find out the similar points of clusters from different database for moving web-based objects based on some common values. Two datasets used for carrying out the research work which is based on the analytical data sets and animal datasets which is also real-life dataset and costly one. The research work which is carried out has the main focus to find out the values most suitable and provide the optimistic result. One more effort which we applied here is that to find out the actual distance between clusters to validate their occurrences. The graphical aspect also represent the output which is based on the grouping of clusters that we have already stated in our algorithm CLSTR concluding the result which we got from the designed algorithm CLSTR we can say that this is the new approach towards the clustering paradigm and the objects which are movable. Data analytics which has been used is also insight the new approach towards the analysis of data. During the working period of this algorithm we found many issues and some other parameters for which we are working till now taking as a challenge.

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

FUNDING INFORMATION

No fund received for this research work.

REFERENCES

Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Sla-based autonomic cloud resource management framework by antlion optimization algorithm. *International Journal of Innovative Technology and Exploring Engineering*, 8(4), 119–123.

Al-Sharif, Z. A., Jararweh, Y., Al-Dahoud, A., & Alawneh, L. M. (2017). Autonomic based cloud computing resource scaling. *Cluster Computing*, 4(1), 1-10.

Baghel, S. K., & Dewangan, B. K. (2012). Defense in Depth for Data Storage in Cloud Computing. *International Journal of Technology*, 2(2), 58–61.

Chen, X., Kordy, P., Lu, R., & Pang, J. (2014). Machine Learning and Knowledge Discovery in Databases MinUS: Mining User Similarity with Trajectory Patterns. *European Conference, ECML PKDD, Nancy, France, Proceedings, Part III*, 436-439.

Dewangan, B. K., Agarwal, A., Choudhury, T., & Pasricha, A. (2020). Cloud resource optimization system based on time and cost. International Journal of Mathematical. *Engineering and Management Sciences*, 5(4), 758–768.

Dewangan, B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2018). Resource scheduling in cloud: A comparative study. *International Journal on Computer Science and Engineering*, 6(8), 168–173.

Dewangan, B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Design of self-management aware autonomic resource scheduling scheme in cloud. *International Journal of Computer Information Systems and Industrial Management Applications*, 11, 170–177.

Dewangan, B. K., Jain, A., & Choudhury, T. (2020). AP: Hybrid Task Scheduling Algorithm for Cloud. *Rev. d'Intelligence Artif.*, *34*(4), 479-485.

Guo, D., Kupper, J. A., & Zhu, X. (2016). Quantifying animal trajectories using spatial aggregation and sequence analysis – a case study of differentiating trajectories of multiple species. *Geographical Analysis*, 48(3), 275–291. .10.1111/gean.12098,2016

Han, J., & Kamber, M. (2016). Data Mining: Concepts and Techniques (2nd ed.). Morgan Kaufmann.

Jacinto, Mário, & Jorge. (2010). Trajectory Classification Using Switched Dynamical Hidden Markov Models. *IEEE Transactions on Image Processing*, 19(5).

Kuijpers & Othman. (2007). Trajectory Databases- Data Models, Uncertainty and Complete Query Languages. *LNCS*, 4353, 224–238.

Kuliko. (2017). Computer Systems That Classification and Prediction Methods from Statistics. Neural Networks.

Kuliko, Sun, & Song. (2014). That Classification and Prediction Methods from Statistics, An analytical method for multi-class molecular cancer classification. *SIAM Rev*, *45*, 726–734.

Lander, E. S., Golub, T. R., & Mesirov, J. P. (2018). An analytical method formulti-class molecular cancerclassification. *SIAM Review*, 45, 706–723.

Mehrotra & Chakrabarti. (2018). Local dimensionality reduction: A new approach to indexing high dimensional spaces. *Proc. of LDB*, 189–200.

Mishra. (2017). Clustering the Movement of Trajectory Based Data. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(11).

Pasricha, A. (2016, October). Credential and security issues of cloud service models. In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT) (pp. 888-892). IEEE.

Shende, M. P. (2012). Survey on user behavior trust evaluation in cloud computing. International Journal of Science. *Engineering and Technology Research*, *1*(5), 113.

Shende, P. (2013). The sliding window method: An environment to evaluate user behavior trust in cloud technology. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(2), 1158–1162.

Sozio, T.-h., Chan, H., & Guerquin, A. (2018) Fully dynamic k-center clustering. WWW, 579-587.

Tseng, V.S., Lee, W-C., & Jia-Ching Ying, J. (2013). Intelligent Mobile Knowledge Discovery and Management Systems, Special Issue, Special Section, Web Mining version. *Transactions on Intelligent Systems and Technology*, 5(1).

Venkatadri, M., & Pasricha, A. (2019). Self-characteristics based energy-efficient resource scheduling for cloud. *Procedia Computer Science*, *152*, 204–211. doi:10.1016/j.procs.2019.05.044

Wang, G. G., Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2016). A new hybrid method based on krill herd and cuckoo search for global optimization tasks. *International Journal of Bio-inspired Computation*, 8(5), 286–299. doi:10.1504/IJBIC.2016.079569

Wang, H., & Yi, J. H. (2018). An improved optimization method based on krill herd and artificial bee colony with information exchange. *Memetic Computing*, *10*(2), 177–198. doi:10.1007/s12293-017-0241-6

Wang, Y. Shen, Z.Bao, and X.Qin.Intelligent traffic analytics: from monitoring to controlling to controlling. In WSDM, pages 778-781, 2020.

Wang, Z., Bao, J.S., Culpepper, Z., Xie, Q., Liu, & Qin, X. (2018). Torch: A search engine for trajectory data. SIGIR, 535-544.

Weibel, W., Speackmann, B. F., Safi, K., & Wengh, D. E. (2016). An interdisciplinary review. *Movement Ecology, R. Analysis and visualisation of movement*, 1–24. 10.1186/s40462-015-0032-y

Wu, G. (2017). Supporting real-time analytic queirs in big and fast data environment. LNCS, 10178, 477-493.

Xin & Cui. (2020). A trajectory Collaboration Based Map Matching Approach for Low-Sampling –Rate GPS trajectories. *MDFI Sensors*.

Yang, C.S., Jensen, L., Chen, & Gao, L. (2017). Pivot based mertix indexing. PVLDB, 10(10), 1058-10969.

Yuan, S., Xia, S., Zhang, L., Zhou, Y., & Ji, C. (2019). An efficient trajectory-clustering algorithm based on an index tree. *Transactions of the Institute of Measurement and Control*, 34(7), 850–861. doi:10.1177/0142331211423284

Zhegown. (2020). Trajectory data mining of moving data. Inderscience Publication.

Zheng. (2020). Trajectory data mining overview. ACM Transactions on Intelligence System and Technology, 6(3), 29-51.

Vishnu Kumar Mishra guided more than ten research scholar for Doctor of Philosophy and M.Phil. His research area are Artificial Intelligence, Data Mining, NLP & Operating System. He has published many papers in Scopus & IEEE and other reputed journals including UGC and other referred journal. He has published more than 70 papers in different journal.

Megha Mishra completed Bachelor of Engineering from CSVTU University, Bhilai. She completed Master in Computer Science & Engineering from CSVTU University Bhilai. She received her Ph.D in (CSE) Computer Science & Engineering from SOA University, Bhubaneswar. Her research area are Natural Language processing, Artificial Intelligence & Operating System. She published many papers in Scopus & IEEE. Presently she is working as Professor at SSGI-SSTC affiliated to CSVTU University, Bhilai.

Bhupesh Kumar Dewangan pursued PhD in computer science and engineering and working as an Associate Professor, in Department of Computer Science and Engineering, OP Jindal University, Raigarh, India, Bachelor of Technology form Pandit Ravi Shankar Shukla University (State University), Raipur and Master of Technology from Chhattisgarh Swami Vivekananda Technical University (State Technical University), Bhilai in computer science and engineering. He has more than 40 research publications in various international journals and conferences with SCI/SCOPUS/UGC indexing. He has three Indian patents on Cloud computing and resource scheduling. His research interests in Autonomic Cloud Computing, Resource Scheduling, Software Engineering, and Testing, Image processing and Object detection. He is a member of various organizations like ISTE, IAPFE, etc. Currently he is editor in special issue journals of Inderscience & IGI publication house, and editor/author of two books of Springer & Taylor and Francis publication house.

Tanupriya Choudhury received his bachelor's degree in CSE from West Bengal University of Technology, Kolkata, India, and master's Degree in CSE from Dr. M.G.R University, Chennai, India. He has received his PhD degree in the year 2016. He has Ten years experience in teaching as well as in Research. Currently he is working as an Associate Professor in dept. of CSE at UPES Dehradun. Recently he has received Global Outreach Education Award for Excellence in best Young researcher Award in GOECA 2018 His areas of interests include Human computing, Soft computing, Cloud computing, Data Mining etc. He has filed 14 patents till date and received 16 copyrights from MHRD for his own software. He has been associated with many conferences in India and abroad. He has authored more than 85 research papers till date. He has delivered invited talk and guest lecture in Jamia Millia Islamia University, Maharaja Agersen college of Delhi University, Duy Tan University, Vietnam etc. He has been associated with many conferences throughout India as TPC member and session chair etc. He is a lifetime member of IETA, member of IEEE, and member of IET(UK) and other renowned technical societies. He is associated with Corporate and he is Technical Adviser of DeetyaSoft Pvt. Ltd. Noida, IVRGURU, and MydigitaI360 etc. He is holding the post of Secretary in IETA (Indian Engineering Teacher's Association-India), He is also holding the Advisor Position in INDO-UK Confederation of Science, Technology and Research Ltd., London, UK and International Association of Professional and Fellow Engineers-Delaware-USA.