International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Reverse Path Nearby Cluster (Rpnc) Query Optimization using Trajectory Clustering With Ensemble Learning (Tcel) In Spatial Networks

G. Dona Rashmi, V. Narayani

Abstract*: : In spatial network, optimal path planning in real time is an open problem. For complex queries, Query optimization is intensively used. From given locations, most accessible locations can be found by Reverse Path nearby Cluster (R-PNC) query. This facility maximizes the commercial value. Distribution Clustering with Support Vector Machine is used to form a R-PNC clusters in the existing works. Trajectories with low positioning accuracy are used by this method. Trajectory Clustering with Ensemble Learning (TCEL) is proposed to overcome this problem. In this, Reverse Path nearby Cluster (R-PNC) query is used for effective path selection. Spatial datasets are handled using a two step process. Set of micro-clusters are maintained in the first step. Based on the input data, this micro clusters are continuously updated. Infinite data sources are compressed into a finite data set using Micro-clusters. More trajectory information can be stored using this finite dataset. In the second stage, ensemble learning is used to convert this micro-cluster into a macro-clusters of trajectories. Reverse the top k –RPNC. Based on their highest distribution they are stored and they are scanned to minimum from maximum. Improved Weight based Grey Wolf Optimization Algorithm (IWGWOA) is used to compute these values. The proposed system is implemented in MATLAB. The evaluation parameters like Normalized Mutual Information (NMI) and Rand Index (RI) are utilized for performance evaluation of proposed model.

***Keywords* : Spatial Networks, Trajectory Clustering with Ensemble Learning (TCEL), Prediction of Location, Improved Weight based Grey Wolf Optimization Algorithm (IWGWOA), Reverse Path Nearby Cluster (RPNC)query..**

I. INTRODUCTION

Spatial networks are ubiquitous in our everyday life and used in transport, navigation, and city planning as well as in Geographical Information Systems (GIS) and other geo-spatial applications [1]. The increasing interest in and importance of spatial networks has led to a large increase of generated spatial networks data. Consequently, database support is essential to store the large volumes of spatial network data and to utilize them in various GIS applications in an efficient way. Spatial databases, which form the data storage foundation of geographical and GIS applications, only offer spatial data types for single spatial objects like points, lines, and regions but are unable to adequately represent connectivity structures like spatial networks [2].The advent of location-based services has led to an increased demand for performing operations on spatial networks in real time. The largely increasing amount of generated data about spatial networks can only be efficiently stored and analyzed in a database system.

Location-based Service (LBS) is a kind of information service that provides users geographical positions located by mobile devices and wireless network. People can share their present location, record travel routes with GPS to share travel experiences [3]. There exists wealth information in the location data, such as user's interests, user's hobbies and user's behavior pattern. LBS may be employed in a number of applications, including: location-based advertising, personalized weather services, entertainment, personal life and so on. An effective location prediction or recommendation can make the users have good experience [4]. The advances in location-acquisition and mobile communication technologies empower people to use location data with existing online social networks in a variety of ways. In order to provide a better service for people, it is significant to discover valuable knowledge, such as interesting locations of individuals from historical trajectories. Therefore, extracting and recognizing interesting locations and predicting next location have been an essential task for remarkable LBS [5]. For now, despite many years of research on location prediction issue, there are still some problems: (1) using raw location data without semantic information makes it hard to study personal purpose of daily route; (2) uncleaned check-in data from social platforms increase the cost of data process and analysis despite dispersed semantic information. The Query optimization is well-known to be a computationally intensive process since a combinatorially large set of alternatives has to be considered and evaluated in order to find an efficient access plan for the query [6]. This is especially so for the complex queries that are typical in mining applications.

From a set of locations, the locations which are accessed very often can be found using a Reverse Path nearby Cluster (RPNC) query. In road network, optimum location can be computed using R-PNC [7].

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
G. Dona Rashmi*, Research Scholar, Bharathiar University. E-mail:donamca08@gmail.com
Dr.V. Narayani, Assistant Professor, Department of Computer Science, st. Xavier's college, Tirunelveli.

There is a reason which makes the R-PNC query as a difficult one. The relationship data point and trajectories is a most common difficulty. All the data points are taken into account for computing minimum distance o between data point and trajectory.

This process is more difficult than the one followed in Reverse Path nearby Cluster (R-PNC) search. The nature of the query input makes another difficulty. The input is not a single data point. It is a list of location candidate [8]. This problem is resolve by the use of effective pruning techniques in single point query models. Another difficulty is that the movement of objects in the road network. In spatial network, the optimization techniques used in free space will not yield better results.

In data analysis, an important role is played by Trajectory clustering. Moving objects trends are used by this clustering technique. The data is received in an incremental manner due to its sequential nature. GPS system is producing new point continuously. Because of two major requirements, trajectory clustering algorithms used in static datasets cannot be used for incremental clustering [9].The first requirement is frequent clustering in efficient manner as it is requested very often. The second one is, the amount of trajectory data to be accumulated. There are various algorithms are available for trajectory clustering of static data. But most of them rectify the problems in incremental huge trajectory data.

When the data set is large, existing techniques requires long time to compute the trajectory cluster as well as for data retrieval. In every minute, the data size is enlarging in a huge manner [10]. So it is not possible to have a real time clustering in every hour within the given time. Incremental accommodation of trajectory data can be used to resolve this issue. In this, local shifts are affected by new data. The area which is in long distance from the new data's local area is not affected by clustering done in this area. Maintaining large data requires more sensible techniques. The change in data source may affect the data due to its small size [11]. Input trajectories with same properties are summarized to save the space of clusters. So in order to overcome this issue, this method proposes the Trajectory Clustering with Ensemble Learning (TCEL) for effective path selection using Reverse Path nearby Cluster (R-PNC) query. It uses a two-step process to handle spatial datasets. The first step maintains a flexible set of micro-clusters that is updated continuously with the input data. Micro-clusters compress the infinite data source to a finite manageable size while still recording much of the trajectory information. The second step, which is on demand, produces the final macro-clusters of the trajectories using the micro-clusters as input and the ensemble based learning is used for increasing the positioning accuracy. Then the top k -RPNC are reversed and sorted according to their highest distributions and they are scanned from the maximum to the minimum. These values are found via the use of the Improved Weight based Grey Wolf Optimization Algorithm (IWGWOA). The paper is organized as follows. Section 1describes the importance of Spatial networks, Location-based services, Reverse Path nearby Cluster (R-PNC) query methods and presents our contributions. Section 2 describes existing methodologies for location recommendation. In Section 3, explained in detail the proposed methodology and its trajectory clustering techniques for query processing to identify the exact location. In Section 4, evaluate the simulation results and discussion. In Section 5, summarizes the conclusion and future work.

II. LITERATURE REVIEW

Lu et al [12] introduced a Reverse Spatial Textual k Nearest Neighbor (RSTkNN) query, i.e., finding objects that take the query object as one of their k most spatial-textual similar objects. And also proposed a hybrid index tree called IUR-tree (Intersection-Union R-Tree) that effectively combines location proximity with textual similarity. Based on the IUR-tree, designed a branch-and bound search algorithm. To further accelerate the query processing, proposed an enhanced variant of the IUR-tree called clustered IUR-tree and two corresponding optimization algorithms. Empirical studies show that the proposed algorithms offer scalability and are capable of excellent performance.

Taniar et al [13] implemented a k-Nearest Neighbor (kNN) queries.There are four perspectives: (i) Space (ii) Result (iii) Query-Point (iv) Relationship perspective. In spatial databases, all possibilities of NN queries are covered by these four perspectives of stationary objects. The simulation results show that the proposed k-Nearest Neighbor (kNN) queries provides the better location recommendation.

Cao et al [14] developed a functionality of spatial keyword query in order to use it easily and relevantly by the users. Efficient support can be made to this. The functionality will monitor the computation proper result and measurement of I/O and runtime performances in an easy manner. Finally the simulation result shows that the proposed spatial keyword queries are effective technique in large case of spatial dataset.

Chen et al [15] proposed a spatial keyword query performance for identification of location. Effective support for progress of geo-textual indexing can be provided the structure developed by this benchmark. The Boolean range query (BRQ), Boolean kNN query (BkQ), and top-k kNN query (TkQ) are supported by this. The BRQ are best served by the Quad that beats all the other indices for this type of query. The BkQ, if the dataset is relatively small and the number of keywords in queries is small, S2I is the most efficient index. If the query keyword count exceed 5, TkQ except at which point the decision tree starts to perform better. The query processing of the indices scales linearly with the number of objects and text length per object. And experimentally these techniques are shown to be a factor that makes a great difference on query performance.

Rocha-Junior et al [16] used road network to implement a top-k spatial keyword queries. The shortest path defines the distance between spatial object and query location. Queries are processed efficiently by novel indexing structures and algorithms. Wu et al [17] proposed two algorithms. They are used to compute safe zones that guarantee correct results at any time. The computation on server and communication between server and client are optimized. Safe zones conservative approximations are exploited and pruning is done in aggressive computational space.

De Felipe et al [18] implemented a top-k spatial keyword queries. Indexing structure IR2-Tree (Information Retrieval R-Tree) is introduced.
It introduced by combining an R-Tree and superimposed text signatures. The proposed algorithms construct and maintain an IR2-Tree, and top-k spatial keyword queries are answered using it.
Tao et al [19] developed a spatial inverted index. It extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time. Not only that the SI-index is fairly space economical, but also it has the ability to perform keyword-augmented nearest neighbor search in time that is at the order of dozens of milli-seconds.
Huang et al [20] proposed an effective method to support moving top-k spatial keyword queries. In addition to finding top-k answers of a moving query, we also calculate a safe region such that if a new query with a location falling in the safe region, we can directly use the answer set to answer the query. To this end, we propose an effective model to represent the safe region and devise efficient search algorithms to compute the safe region. We have implemented our method and experimental results on real datasets show that our method achieves high efficiency and outperforms existing methods significantly
Gao et al [21] introduced a new type of queries, namely, Reverse top-K Boolean Spatial Keyword (RkBSK) retrieval, which assumes objects are on the road network and considers both spatial and textual information. Given a data set P on a road network and a query point q with a set of keywords, an RkBSK query retrieves the points in P that have q as one of answer points for their top-k Boolean spatial keyword queries. Formalize the RkBSK query and then propose filter-and-refinement framework based algorithms for answering RkBSK search with arbitrary k and no any pre-computation. To accelerate the query process, several novel pruning heuristics that utilize both spatial and textual information are employed to shrink the search space efficiently. In addition, a new data structure called count tree has been developed to further improve query performance. An experimental evaluation using both real and synthetic data sets demonstrates the effectiveness of presented pruning heuristics and the performance of the proposed algorithms.
Zhao et al [22] introduced a new type of query, the Reverse Top-k Geo-Social Keyword (RkGSK) query. This query takes into account spatial, textual, and social information, and finds prospective customers for geo tagged objects. And also proposed a hybrid index, the GIM-tree, which indexes locations, keywords, and social information of geo-tagged users and objects, and then, using the GIM-tree, present efficient RkGSK query processing algorithms that exploit several pruning strategies. The effectiveness of RkGSK retrieval is characterized via a case study, and extensive experiments using real datasets offer insight into the efficiency of the proposed index and algorithms.

III. PROPOSED METHODOLOGY

In this propsed approach, Reverse Path nearby Cluster (R-PNC) query is introduced. Both textual and spatial information as well as objects in the raod are assumed. Reverse Path nearby Cluster (R-PNC) query retrieves the points in P by using a data set P on a road network and a query point q with a set of keywords. This method proposes the Trajectory Clustering with Ensemble Learning (TCEL) for effective path selection using Reverse Path nearby Cluster (R-PNC) query. Spatial dataset is handled by two-step process. Micro-cluster are maintained in the first and based on input data, they are continuously updated. Infinite data sources are compressed by micro-clusters. Micro-clusters are used to form macro-clusters of trajectories in the second step. Accuracy of positioning is increased by ensemble based learning. Then the top k -RPNC are reversed and sorted according to their highest distributions and they are scanned from the maximum to the minimum. These values are found via the use of the Improved Grey Wolf Optimization Algorithm (IGWO). A connected and undirected graph G(VE, ED, FU, WE, CON) is used to model a spatial network. Where, vertex set is given by VE and edge set is given by $ED \subseteq VE \times VE$. End of the road or intersection is given by vertex $v_i \in VE$ [22]. Two vertices and road segment defines an edge $e_k = (v_i, v_j) \in ED$. Geometrical information of spatial network G is given by a function $FU : VE \cup ED \rightarrow$ Geometries [23]. Every edge is assigned with a real valued weight by function $WE: ED \rightarrow R$. The weight $W(e)$ corresponds to the road segment's length or its travel time or fuel consumption of an edge e. Historic traffic data are mined to obtain this.The network shortest path betweentwo vertices p_a and p_b is denoted by $SP(p_a, p_b)$ in spatial domain and its length is given by $Sd(p_a, p_b)$.
Problem Definition 1
Given a query qe(So, key), a parameter k, and a dataset 'P' with each POI p ∈ P in the form of (So, key), let Pqe.key be the set of POIs in P that contain qe.key, i.e., Pqe.key= {p ∈ P | qe.key ⊆p.key}. A Top k PN query (on the road network) issued at q, denoted as TkPN(qe), returns the k POIs in Pqe.key having the minimal network distances to q, formally, TkPN(qe) ={S ⊆ Pqe.key| |S| = k ∧∀ s ∈ S, ∀ p ∈ (Pqe.key-S), ||qe,s||≤||qe,p||}. For any data point in TkPN(qe), that it is one of the Boolean spatial keyword nearest neighbors of 'qe'.S be the set of POIs for user given query.
Problem Definition 2 Reverse top-k Reverse Path Nearby query on the road network. Given a query qe(So, key), a parameter k, and a dataset P, an TkRPN query (on the road network) issued at qe, denoted as TkRPN(qe), retrieves all the POIs in P whose top-k spatial keyword queries include qe, formally, TkRPN (qe) = {p ∈ P | q ∈ TkPN (p)}. A sequence of vertices $(p_1, p_2, \ldots, p_n)$ represnts a reverse query route , where, adjacent vertices in G are given by pi and pi+1 are, i = (1, 2, . . . ,n – 1). The distance d(p,qe) betweenvertex 'p' and a given reverse query route 'qe' in a spatial network is defined in equation (2).
3.2. Query processing
Form the data set, exact PNCs of all trajectories are searched to solve Reverse Path nearby Cluster (R-PNC) query. The location which is having high value of influence factor is found by combining the results obtained in the search [23].
A sequence of vertices $(p_1, p_2, \ldots, p_n)$ is defined for a query route, where adjacent vertices in G are represented by pi and pi+1, where i = (1, 2, . . . ,n – 1). Distance d(p,qe) between vertex 'p' and query route 'qe' is defined in spatial network as,

$$dis(p, qe) = \min_{p_i \in qe} \{ sdis(p, p_i) \} \quad (1)$$

where pi is a vertex belonging to query 'qe'.

Distance d(p,qe) between given vertex 'p'and reverse query route 'qe' is defined in spatial network as,

$$dis(p, qe) = \min_{p_i \in qe} \{ sdis(p, p_i), con \}, con \in S_{ct} \quad (2)$$

where pi is a vertex belonging to query 'qe'.

3.3. Trajectory clustering with Ensemble learning

Trajectory clustering is the process of extracting the similarity, anomaly and valuable patterns from the trajectory data. The purpose of trajectory segmentation is to form sub-trajectories by dividing whole trajectory. In sub-trajectories, targets move in similar motion feature. So the key to the segmentation is to find the point where the target's motion feature changes rapidly [24]. The angle between two track segments can reflect the target's movement trend. Continuously receive the input data as shown in Figure 1. This method has two steps for effective identification of a location using a Reverse Path Nearby Cluster Query.

Micro-clustering is an initial step. Computing the clusters on request by preprocessing the input from infinite data source is a difficult task. Trajectory micro clusters are used in this proposed work to resolve this issue. The clusters extreme tightness corresponds to the term "micro". Very fine granularity clusters has to be formed. When compared to final trajectory clusters, micro-cluster count will be very high to form fine granularity clusters. Second row of Figure 1 shows the micro-clusters. Formation of macro-clustering has to be done after this. When new data is received micro-clusters are updated. But in macro-clustering, updating happens only after receiving request of trajectory clusters. Micro-clusters are used as an input in this stage.

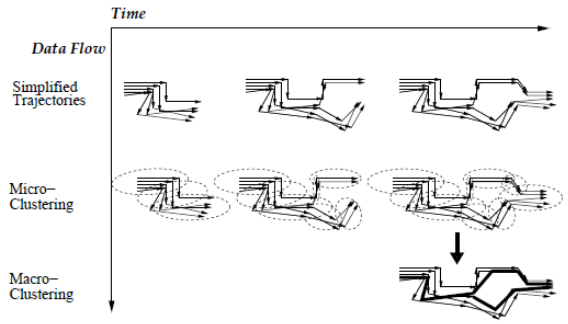


Figure 1. Proposed Trajectory clustering framework

3.3.1. Trajectory Micro-Clustering

Fine-granularity clustering is maintained by introducing trajectory micro clusters. Results of local clustering are affected by the arrival of new trajectories. The information about local partitioned trajectories is summarized in the micro-cluster. Final clusters are less restrictive than the micro-clusters [24]. Final clusters can be computed efficiently by using micro-clustering instead of computing it from line segments.

Algorithm 1: Trajectory Micro-Clustering

Input: New trajectories $I_{tcurrent} = \{TR_1, TR_2, \ldots, TR_{nTR}\}$ and existing micro-clusters $MC = \{MC_1, MC_2, \ldots MC_{nMC}\}$.
Parameter: d_{max}
Output: Updated MC with new trajectories inserted.
Algorithm:

```
forevery TRi ∈ Itcurrent do
forevery Li ∈ TRi do
    Find the closest MCk to line segment Li
if distance(Li, MCk) _ dmax then
        Add Li into MCk and update MCk accordingly
else
        Create a new micro-cluster MCnew for Li;
if size of MC exceeds memory constraint then
        Merge micro-clusters in MC
```

Generation and maintenance of micro-clusters are shown in algorithm 1. For the arrival of batch of new trajectories, closest micro-cluster MCk is computed for every trajectories each line segment Li. Liis inserted into MCk, if distance between Li and MCk is less than a distance threshold (dmax) else create a new micro-cluster MCnew. Micro-clusters are merged if the creation of new cluster makes the overloading of number of micro-clusters.

Micro-Cluster Definitions: Set of partitioned trajectories are holded and summarized by trajectory micro-cluster. Trajectory micro-cluster are line segments.

Definition 1 (Micro-Cluster).

A tuple is used to define the trajectory micro-cluster (or micro-cluster) of set of directed line segments $L_1, L_2, \ldots, L_N$. They are (N, LS_{center}, LS_θ, LS_{length}, SS_{center}, SS_θ, SS_{length}). Where, number of line segments in micro-cluster is given by N, Centr, angle and lengths linear sums of line segments are given by LS_{center}, LS_θ, and LS_{length} and squared sum is given by SS_{center}, SS_θ, and SS_{length}.

The cluster feature vector extension defines the trajectory micro-cluster. The line segment information is summarized and represented by a linear LS. Micro-cluster's tightness is computed by square sum SS. New line segments can be added to a micro-cluster and they can be merged because of definition's additive nature. But there is consistency in line segments distance measure. Representative line segment is associated with all trajectory micro-cluster. It is an "average" of sorts.

Definition 2 (Representative Line Segment). Starting and ending points (s and e) are used to represent the micro-cluster's representative line segment. From micro-cluster features, s and e are computed.

$$s = \left(center_x - \frac{cos\theta}{2} len, center_y - \frac{sin\theta}{2} len \right) \quad (3)$$

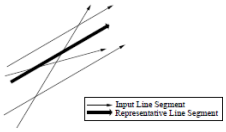
$$e = \left(center_x + \frac{cos\theta}{2} len, center_y + \frac{sin\theta}{2} len \right) \quad (4)$$

Where $center_x = \frac{LS_{center_x}}{N}, center_y = \frac{LS_{center_y}}{N}, len = \frac{LS_{length}}{N}, and\ \theta = \frac{LS_\theta}{N}$.

An example is shown in figure 2. Thin lines represents the line segments and there four of them. Thick line is used to represent the line segment of micro-cluster.

- **Creating and Updating Micro-Clusters**

For every new line segment L_i arrival, closest micro-cluster MC_k is computed to observe it. L_i is added to MC_k, if d_{max} is greater than distance between L_i and MC_k. Update MC_k accordingly. If d_{max} is less than distance between L_i and MC_k, form a new cluster. The execution this process is shown below.



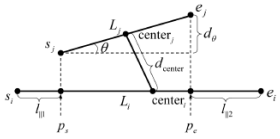


Figure 2. Representative Line Segment Figure 3. Line Segments Distance

Define the distance between micro-cluster and line segment. As representative line segments are in micro-clusters, the distance is defined in terms of distance between line segments. It has three components namely, angle distance (d_θ), parallel distance (dk) and center point distance (dcenter). Line segment Hausdorff distance is modified by pattern recognition which uses similarity measure. The adaptation of distance is done using this. Similar distance measure can also be used. Where, d id replaced by dcenter. Between d_θ and dk, dcenter is a balanced measure. The starting point of Li is given by siand ending point is given by ei. Li is assigned with longer line segment and Lj is given by shorter line segment. Distance function is shown in figure 3.

Definition 3. There are three components in the distance function and it is defined by sum of these components:

$$\mathrm{dist}(L_i, L_j) = d_{centre}(L_i, L_j) + d_\theta(L_i, L_j) + d_\parallel(L_i, L_j) \quad (5)$$

The distance of center is given by:

$$d_{centre}(L_i, L_j) = \|center_i - center_j\| \quad (6)$$

Where, Euclidean distance between center points of L_i and L_j is given by $\|center_i - center_j\|$.

The distance of angle is given by:

$$d_\theta(L_i, L_j) = \begin{cases} \|L_j\| \times \sin(\theta), & 0^\circ \le \theta < 90^\circ \\ \|L_j\|, & 90^\circ \le \theta \le 180^\circ \end{cases} \quad (7)$$

Where, length of Ljis given by $\|L_j\|$, small intersecting angle between Liand Lj is given by $\theta(0^\circ \le \theta \le 180^\circ$. θis measured by neglecting direction. So it will be less.

The distance in parallel is given by:

$$d_\parallel(L_i, L_j) = \min(l_{\parallel 1}, l_{\parallel 2}), \quad (8)$$

Where$l_{\parallel 1}$are the Euclidean distances of psto si and $l_{\parallel 2}$is that of peto ei. ps and pe are point's projection points sjand ejonto Li respectively.

If distance dmaxis greater than the distance from Li, insert Liinto it and update the square and linear sums of MCk after computing micro-clusters MCk. The update is made efficient because of the additive property of those sums. For Li, create a new micro-cluster, if dmax is less than the distance between Li and nearest micro-cluster. Line segments are used to derive the new micro-clusters initial measure value.

- Merging Micro-Clusters

In order to fulfill the space limitations, micro-clusters are merged if space requirement exceeds the space limit. The efficiency of the algorithm is highly affected by the micro-clusters count. Increase in the micro-cluster, requires more time to compute the nearest micro-cluster. There is no necessity to have all the micro-cluster as they become closer after updating in many rounds [24]. To reduce storage and increase the efficiency, close micro-clusters are merged by the algorithm. Micro clusters with same line segments can be merged in order to produce less information.

Distance between two micro-cluster's representatives line segments are computed to find the similarity among the clusters. Micro-clusters tightness are not considered for this computation.

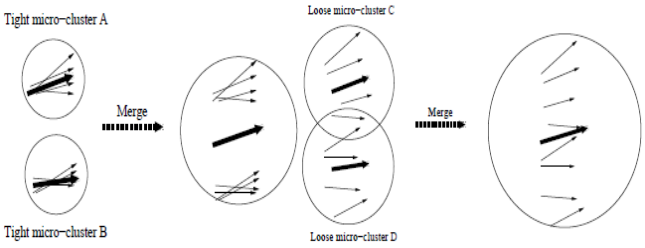


Figure 4(a). Merging tight micro-clusters
Figure 4(b). Merging loose micro-clusters

Figure 4. Merging micro-cluster

The effect of tightness on two-cluster distance is shown by figure 4. Two micro-clusters which are tight and their merging are shown in figure 4(a). Two loosly coupled micro-clusters are shown in figure 4(b). Micro cluster A and C has an equal representative line segment as B and D. If this representative line segments are used for measuring the distance between the micro-clusters, then the distance value between A and B will be equal to C and D. In this situation, merging chance of A and B will be same as merging chance of C and D. But micro-clusters C and D are merged.

The reason behind this merging is, merging of two tight micro-clusters will break the tightness and there is no effect in merging the loosely coupled micro-clusters. So extent of micro-clusters is used to solve this issue and distance between micro-clusters can be computed using this.

- Micro-Cluster Extent

The tightness of micro-cluster shows its extension. Tuples are used to represent the micro-clusters. Square and linear sums of angle, length and center are maintained by this tuples. The trajectory micro-cluster's tightness is measured by using the parameters called extent length, extent center and $extent_\theta$. From SS and LS standard deviations are computed to find the extent.

3.3.3. Trajectory Macro-Clustering

The trajectory clusters in overall is produced by Trajectory clustering framework in last step. The arrival of new data makes the micro-clustering to process whereas user request is required in macro-clustering. In spatial point, distance between micro-clusters enables the adaptation of different clustering method.

Distances between micro-clusters are used to replace the distance between spatial points [24]. Instead of using trajectory partitions, micro clusters are used to perform Trajectory clustering. Ensemble Learning based algorithm is used to cluster the micro-clusters. It computes the maximum "density connected" components, based on this macro-cluster is formed.

3.3.4. ENSEMBLE LEARNING

In ensemble learning, set of models are combined and same task is solved every model. This is done to a global model which is best as well as has better results when compared to single model. Enhanced Kernal based Support Vector Machine (EKSVM), Enhanced Deep Neural Network and Weighted Random Forest Algorithm are combined based on weighting method. This classifier performs very effectively compare to the single classifier for location prediction.

- Enhanced Kernel based Support Vector Machine (EKSVM)

Based on theories of discriminant function, a machine learning method SVM is formed. Support vectors are used by this to classify the data. Data patterns are represented by this support vectors. For a given N data samples $(x_1, y_1) \ldots (x_i, yi) \ldots (xN, yN)$, discriminant function f(x), such that $y_i = f(x_i)$ has to be find. It is a problem of general two-class classification. f(x) = sgn(w · x − b) represents the possible linear discriminant function. In data space, separating hyperplane is given by w · x − b = 0. Hyperplane with maximum separating margin between two classes has to be found to choose a discriminant function. f(x) = sgn(Pl i=1 αiyi(xi · x − b) is the final linear discriminant function. where, number of training records are given by l, label associated with training data are given by yi∈ {−1, +1} , 0 ≤ αi ≤ C (constant C > 0), and support vectors are given by xi.

The data points are transformed into high dimensional space, in order to make the data points are linearly separable, if surface separating two classes is not linear. The SVM's nonlinear discriminant function is expressed by,

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b\right), \quad (9)$$

Where, kernel function is given by K(xi, x) and data points are transformed using this. Linear, sigmoid, radial and polynomial are the most commonly used kernel functions. The variation among data feature are not considered by this kernel functions. Equal importance will be given to all the features of dataset as in the kernel function format K(xi, x) of SVM.

This fact of giving equal importance may degrade the performance of SVM. In order to resolve this, a kernel function is added with a weight [25]. Every feature's importance is measured by a weight. K(wxi, wx) represents the format of new kernel function. Weights are represented as w. With feature weight, nonlinear discriminant function is given by,

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(wx_i, wx) + b\right), \quad (10)$$

This enhanced kernel is an independent one. Based on the application, the kernel function can be chosen. From training data, the weights are computed as well as generated by using rough set theory.

The basic principles followed to compute the weights are, a. weight of a feature is 0, if it is not in any reducts, b. based on the appearance of feature in reducts, the importance will be given, c. if a reducts has only few features, then they will more important.

Algorithm 3 shows the algorithm based on principles discussed above. Feature weight computation and feature ranking are done by rough set theory. The feature with weight 0 is not important and they can be deleted. Selection of feature and ranking of it are combined in algorithm 3.

Algorithm 3: Feature Weights Calculation

```
Input:  micro clusters.
Output: macro clusters.
Find out all the reducts of D
using rough sets;
Ncluster ← cluster count in D;
Nreduct ← reducts count of D;
//Initialize the weight of each
clusters.
for (i ← 0 to Nfeature) do
wi ← 0;
end
// Calculate the weight of each
clusters.
    for (i ← 0 to Ncluster) do
     for (j ← 0 to Nreduct) do
          if (feature i in the
jthreductRj) then
               m ← cluster count in
Rj ;
wi← wi + 1 m ;
               end
    end
end
Scale the values of clusters
weights into the interval [0, 100];
```

- Enhanced Deep Neural Network

The representation of data can be learned by a multi-layer technique in deep learning. Multi-layer neural network is used for the same. Deep neural networks utilize feedback signal as well as transformations of data to map a feature with target in order to learn it. The importance of successive layer learning is shown by DNN. Wide range of problems can be solved by a simple and successful concept called overarching. When compared to brain, neural network operation is simple. It has input, output and hidden layers in its basic model [26]. User can give their data to input layer and most of the learning is done at the hidden layer. The results are projected by output layer.

Algorithm 1: Feed forward DNN algorithm

This training algorithm provides a high-level pseudocode for preparing a network.

Node: Computation's basic unit
Layer: A collection of nodes of the same type and index (i.e. input, hidden, outer layer)
Connection: A weighted relationship between nodes of one layer to the node of another layer
W: The weight of a connection
I: Input node (the neural network input)
H: Hidden node (a weighted sum of input layers or previous hidden layers)
HA: Hidden node activated (the value of the hidden node passed to a predefined function)
O: Output node (A weighted sum of the last hidden layer)
OA: Output node activated (the neural network output, the value of an output node passed to a predefined function)
B: Bias node (always a contrant, typically set equal to 1.0)
e: Total difference between the output of the network and the desired value(s) (total error is typically measured by estimators such as mean squared error, entropy, etc.)

Step 1: Initialization

The first step after designing a neural network is initialization:

- Using a random number weights W1 to W12 are initialized by normal distribution.
- Bias nodes are set to 1 (B1 = B2 = 1.0).

Note: The variance of distribution is different from this value and network's convergence rate is depends on this.

Step 2: Feed-Forward

For hidden and output layers, all the values are computed.

- Input node vaules are set.
- Value of hidden node is computed.

$$H_1 = I_1W_1 + I_2W_3 + B_1W_5 \quad (11)$$

$$H_2 = I_1W_2 + I_2W_4 + B_1W_6 \quad (12)$$

- For hidden laey, activation function (sigmoid function) is selected.

$$S(x) = \frac{1}{1+e^{-x}} \quad (13)$$

- Calculate hidden node activation values:

$$HA_1 = \frac{1}{1+e^{-H_1}} \quad (14)$$

$$HA_2 = \frac{1}{1+e^{-H_2}} \quad (15)$$

- Calculate output node values:

$$O_1 = HA_1W_7 + HA_2W_9 + B_2W_{11} \quad (16)$$

$$O_2 = HA_1W_8 + HA_2W_{10} + B_2W_{12} \quad (17)$$

- For oupt layer, activation function (linear function) is selected.

$$y = f(x) = x \quad (18)$$

- Calculate output node activation values:

$$OA_1 = O_1 \quad (19)$$

$$OA_2 = O_2 \quad (20)$$

- Total erroe is computed; For node i, obtained value of output is given by OAi and desired output is given by yi

$$e = \frac{1}{n}\sum_{i=1}^{4}(y_i - OA_i)^2 \quad (21)$$

There will be high value of error after this. This error between desired value and output can be reduced by adjusting the weights using a backpropagation algorithm.

Step 3: Backpropagation

In the forward pass, observation batch of is selected by DNN. Every node connection is given with a random weight and output is predicted [27]. The improvement in the accuracy by automatic adjustment of weights is the base of neural network. This is known as backpropagation. The weights are adjusted in incremental order to produce an output which is close to the expected one using training data.

The below expression can be used to adjust the weight of the network by backpropagation.

$$W_i^{k+1} = W_i^k - \eta\frac{\partial e}{\partial W_i^k} \quad (22)$$

Where, iteration number isrepresnted by k, learning rate is represnted by η, derivative of total error is given by $\frac{\partial e}{\partial W_i^k}$. This derivative is based on adjusted weight. For every weight in the network inclusive of bias connection, a formula is derived.

Step 4: Model training

Created base model is trained by the data. This training is done by fit fuction. Fit function gets the based model and data to train the model. DNN model is produced. The optimal model is computed by running DNN model, which tunes the various parameters like batch normalization, rate of learning, model capacity, dropout.

$$Y = F(y_{in}) \quad (23)$$

In training algorithm, decide the subtask execution and activation function application depends on Y value.

- Weighted Random Forest Algorithm

During the voting of classification, equal weight is given to the all RF decisions. In prediction of classification of unbalanced data, the weightage will reduce the efficiency of classification [28]. RF algorithm is incorporated with weighted F-measure to rectify this issue. Various decision trees are assigned with various weight values in this method to produce high performance in prediction of location.

Steps involved in this algorithm is given below,

Step 1. Test, validation and training sets are confirmed. Bootstrap method is used extract K+1 datasets with known categories randomly from original dataset. Like original dataset, extracted data set is also going to have the capacity of n. In that extracted dataset, one set is used for validation purpose and other K sets are used training purpose. Remaining data sets in the original dataset corresponds to test set.

Step 2. RF classifier is constructed. K training set are given as input. The model is constructed using RF algorithm. Combine K classification decision trees to apply combination classifier.

Step 3. Compute the F-measure of sub classifier in order find the weight value. Validation set is given as an input. Each sample in validation set are classified.

They are classified by considering each decision tree in the forest. They are assumed as an independent classifier. For each classifier, TP, TN, FP, FN, TN, recall rate and precision rate are computed. Based on the sub classifier weight, compute the F-measure. Describe sub classifier j's weight.

Step 4. Performance of the model is evaluated using input set.

Step 5. Unclassified samples are given as input. Based on the F-measure, the samples are classified. Each sub classifier's weighted vote of classification defines the H. Describe sub classifier j's classification result and weight result. Figure 5 shows the basic ensemble model.

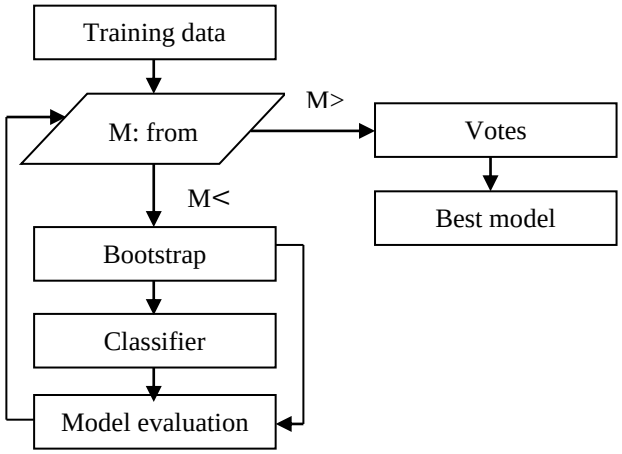


Figure 5. Basic Ensemble Learning model

The decision of a final classification is expressed as,

$$W_j = F_j = \frac{2TP_j}{2TP_j+FP_j+FN_j} \quad (24)$$

$$H(x) = \underset{Y}{\arg\max} \sum_{j=1}^{K} W_j I(h_j(x) = Y) \quad (25)$$

In eq. 25, combined classification model formed by weighted RF algorithm is represented by H(x), output variables are represented by Y and indicator function is represented by I.

3.4. Improved weight based Grey Wolf Optimization Algorithm (IGWO).

Depends on grey wolves nature, Grey wolf optimization algorithm (GWOA) is proposed. The behavior of grey wolves like hunting and social behavior are used by this algorithm. There are four classes of grey wolves namely, alpha (α), beta (β), delta (δ), and omega (ω) wolves. Among this leading wolves is α. During hunting process, important decisions are taken by them. It also maintains the social equality by tracking other wolves in group [29].The β wolves are the second dominant wolves and they are the consultant of α wolves. Guidance under various situations are given by them. The β wolves are upgraded as α wolves, when they get older or die. The ω wolves are controlled by δ wolves and share the information with α and β wolves. The ω wolf is at the lowest level and they are the children of the group. There three main steps involved in hunting process. They are, prey tracking and searching, harassing and encircling of prey by grey wolves and prey attacking. The mathematical modeling of GWOA is given by [1,10],

1. Community Hierarchy: The α wolf is considered as fittest solution. The β defines second best solution and δ wolves define third best solutions. ω wolves used to define balance candidate solution. The α wolves, β wolves and δ wolves are used to guide the optimization process and they are followed by ω wolves.

2. Encircling the prey: The prey is enriched by grey wolves. Following equations expresses this process,

$$X(t+1) = X_p(t) - A.\,Dist \quad (26)$$

Here

$$Dist = \left|C.X_p(t) - X(t)\right| \quad (27)$$

Where, distance between a grey wolf (X). and position of prey (XP) are represented by Distand current iteration is denoted by t. Position vector of prey is represented by $X_p(t)$. Control coefficients are given by A and C are , they are expressed as,

$$A = 2av.\,rand_1 - av \quad (28)$$

$$C = 2.\,rand_2 \quad (29)$$

During the iterations, the av value is decreased from 2 to 0 linearly. Random variables are represented by 1 rand and 2 rand and they range from 0 to 1.

3. Hunting the prey: The awareness about the position of prey are contained by best candidate solutions. Based on the searched best agents, they updates their positions. The following equations describes the behavior of hunting,

$$D_{alpha} = \left|C_1.X_{alpha} - X\right| \quad (30)$$

$$D_{beta} = \left|C_2.X_{beta} - X\right| \quad (31)$$

$$D_{delta} = \left|C_3.X_{delta} - X\right| \quad (32)$$

$$X_1 = X_{alpha} - A_1.D_{alpha} \quad (33)$$

$$X_2 = X_{beta} - A_2.D_{beta} \quad (34)$$

$$X_3 = X_{delta} - A_3.D_{delta} \quad (35)$$

$$X_3 = \frac{X_1+X_2+X_3}{3} \quad (36)$$

4. Attacking the prey: The exploitation power of algorithm is represented by this step. The value of av is decreased from 2 to 0 linearly to perform this. The wolves are made to attack on prey by keeping the value of A is less than 1.

5. Searching for the prey: The exploitation power of algorithm is represented by this step. In order search best prey, grey wolves are diverges from each other. The value of A is assigned with the value greater than 1, to avoid the wolves to keep local optima instead searching for better prey. In step 2, two control parameters are introduced namely, C and A. The exploitation and exploration power of GWA are balanced using parameter A. Exploration process is influenced by parameter C. In distance computation, the weight of the prey is indicated by the observed value of C [29]. They prey's weight is increased if C value is greater than 1 else it is reduced. Using this condition, parameter C is modified. The exploitation of GWOA can be improved by using this method. The number of iterations defines the value of C and it is given by,

$$C = 2.\,rand_2 - \left({}^{av}/_{2}\right) \quad (37)$$

Basic steps in GWOA are used in the enhanced weight based grey wolf optimization algorithm (IWGWOA). Step 2 introduces the proposed equation. Equation 37 is used to replace equation 27. Steps involved in IWGWOA are shown in figure 6.

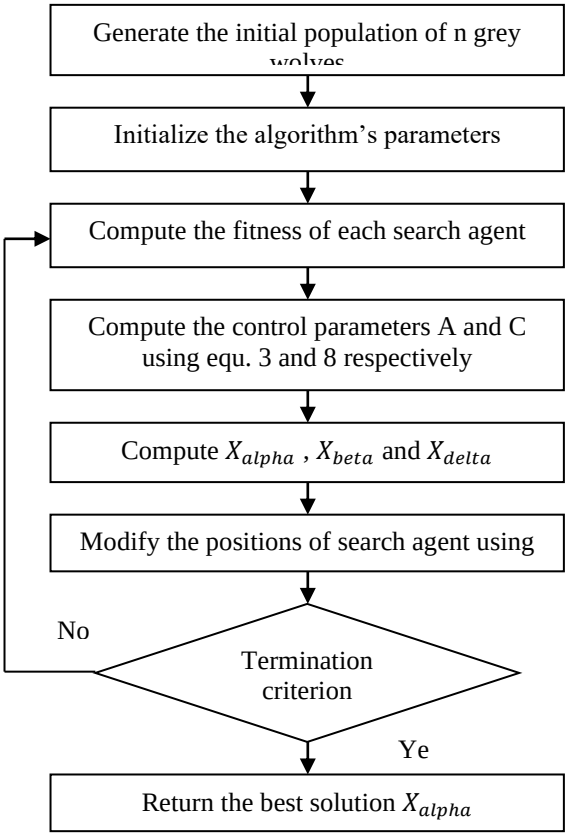


Figure 6. General steps for improved weight based Grey Wolf Optimization Algorithm

3.5. Top K Extension

Top-k PNC query is supported by extending the proposed technique. All the clusters are scanned in spatial as well as density domain. For top-k PNC query, global upper bound UBk is defined as,

$$UB_k = \max_{c \epsilon C_k} \{E_{sd}(c,q).ub.con\},$$

(38)

Where, set of clusters are represented by C_f and they are used in spatial as well as time domain. $C_k = \{c_1, c_2, \dots\dots, c_k\}$,so that

$$\forall c \epsilon C_k (\forall c' \epsilon C_f \backslash C_k (E_{sd}|(c,q).ub \leq E_{sd}(c',q).ub), con)$$

(39)

The UB_k value is dynamic in nature and it is continuously updated. Top-k PNC query searched. The search process is terminated when LB is greater than UB_k. Cluster candidates are refined and top-k clusters with minimum evaluation score in spatial density are computed.

IV. RESULTS AND DISCUSSION

In this section, performance of proposed RPNC-TCEL is computed by conducting experiments with spatial data sets [30]. City of San Francisco's road network is used in this experimentation. It has edges and nodes, so that location prediction can be done effectively. Proposed and existing algorithms are evaluated using performance metrics. The performance metrics includes Query time, candidate ratio and Rand Index (RI). High performances are exhibited by the proposed technique when compared to existing techniques.

i) Rand Index

The similarity among two data clusters can be measured by a parameter called Rand Measure or Index [31]. The adjustments made to group the elements may also be termed as Rand Index and it ranges from 0 to 1. Accuracy is related to rand index in mathematical point of view and it does not require any class labels. Rand index is given by,

$$Rand\ Index = \frac{a+d}{a+b+c+d}$$

(40)

ii) Candidate Ratio

Generate data points uniformly with density on the network. Low value of data point density corresponds to high range requirement for searching [32]. Increase in density decreases the candidate ratio.

$$Candidate\ Ratio = \frac{CS.size}{O.num}$$

(41)

$$Density\ of\ data\ points = \frac{O.num}{V.num}$$

(42)

Where, data point set is represented by O , vertex set in the network are represented by V and data point candidate set is represented by CS and it is specified by RPNC query. High pruning effect can be obtained with low value of candidate ratio.

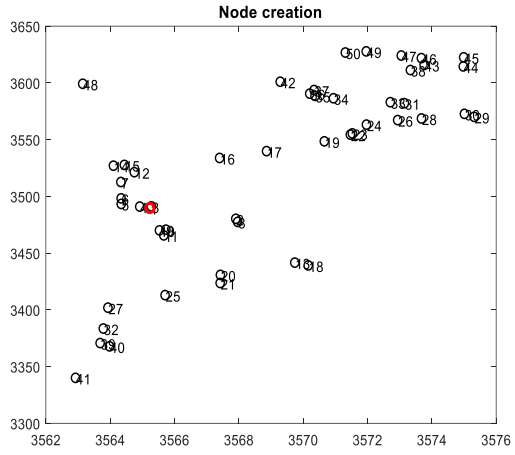


Figure 7. Node creation of the proposed RPNC-TCEL technique

The figure 7. Shows the Node creation of the proposed RPNC-TCEL search for initializing process of the Reverse Path Nearby query based search. This is the starting process of RPNC query.

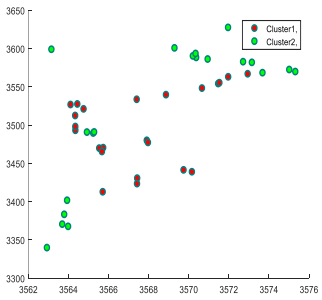


Figure 8. Formation of clustering using the proposed trajectory clustering with ensemble learning

The figure 8. Illustrates the formation of clustering using the proposed trajectory clustering with ensemble learning (RPNC-TCEL) for Reverse Path Nearby query based search.

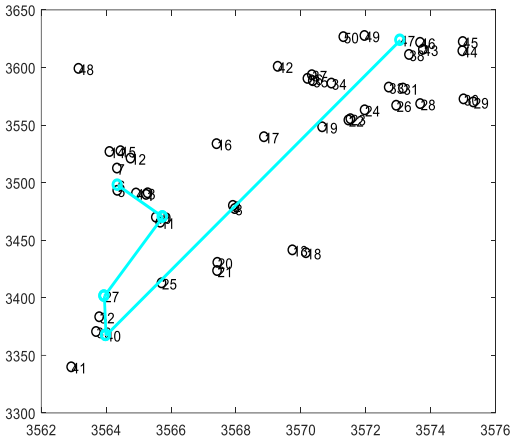


Figure 9. Reverse path detection of the proposed RPNC-TCEL

The Figure 9. Shows the Reverse path detection of the proposed RPNC-TCEL. This proposed Reverse Path Nearby query based search exactly determines the reverse path route by user query. The simulation result verifies that the proposed method effectively detects the route for the user.

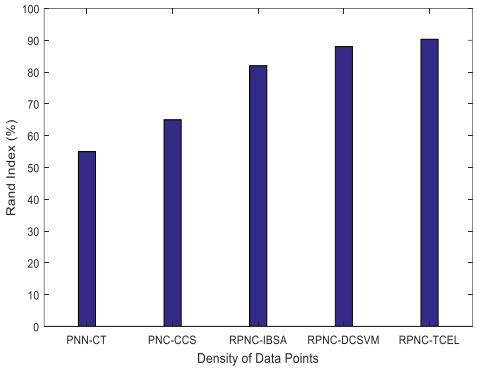


Figure 10. Accuracy comparison of existing and proposed clustering algorithms

Figure 10. Shows the accuracy comparison of proposed and existing algorithms of clustering. From the above results it is concluded that high accuracy is produced by the proposed RPNC-TCEL algorithm.

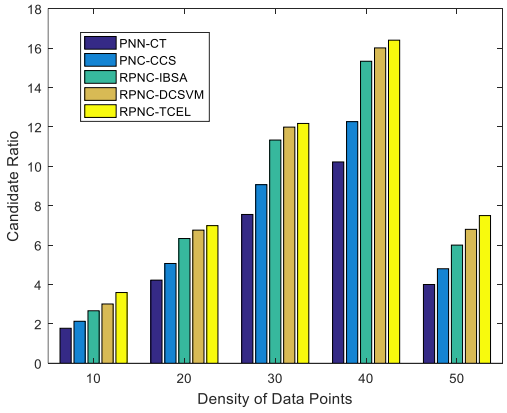


Figure .11. Candidate ratio analysis of existing and proposed clustering algorithms

Candidate ratio analysis of proposed and existing algorithms of clustering is shown by Figure 11. It shows that proposed RPNC-TCEL algorithm increases the density of data points and decreases the candidate ratio when compared to existing algorithms of clustering.

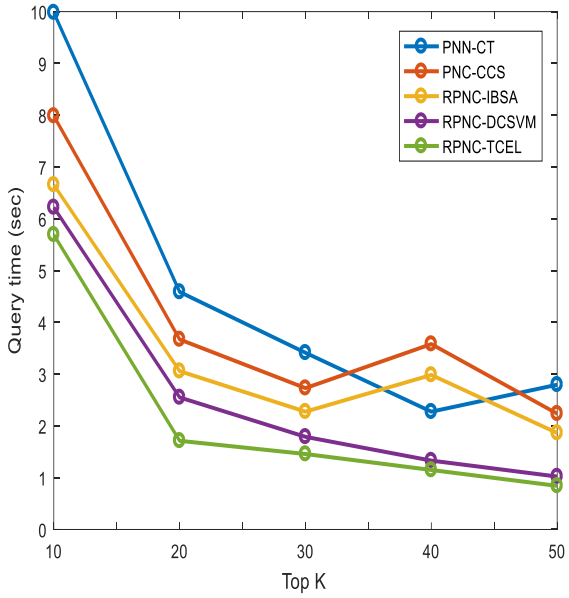


Figure .12. Query time Comparison of existing and proposed clustering algorithms

Comparison of query time of proposed and existing algorithms of clustering is shown by Figure 12. It shows that query time required by the proposed RPNC-TCEL approach is low when comparing with existing algorithms.

V. CONCLUSION

In this work, trajectory data are clustered using, Trajectory Clustering with Ensemble learning for Reverse Path nearby Cluster (RPNC). Datasets are processed in two-steps. Micro-clusters are formed and based on based on input data they continuously updated. Infinite data sources are compressed into a finite data set using Micro-clusters. More trajectory information can be stored using this finite dataset. In the second stage, ensemble learning is used to convert these micro-clusters into macro-clusters of trajectories. Reverse the top k –RPNC. Based on their highest distribution they are stored and scanned to minimum from maximum. Improved Weight based Grey Wolf Optimization Algorithm (IWGWOA) is used to compute these values. All the input data are not required at the same time in TCEL method. It makes the difference and advantage when compared to other methods. Trajectory data summary is given by these micro-clusters. New information is updated by this data trajectory. Analysis of location prediction can be done easily with this. Space and time dimensions cannot be combined fully. This is the major drawback of trajectory clustering algorithms and they considered time as an additional dimension of trajectory.

REFERENCES

1. Kanjilal, V., & Schneider, M. (2010). Modeling and Querying Spatial Networks in Databases. JMPT, 1(3), 142-159.
2. Shang, S., Yuan, B., Deng, K., Xie, K., & Zhou, X. (2011, November). Finding the most accessible locations: reverse path nearest neighbor query in road networks. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 181-190). ACM.
3. Du, Y., Wang, C., Qiao, Y., Zhao, D., &Guo, W. (2018). A geographical location prediction method based on continuous time series Markov model. PloS one, 13(11), e0207063.
4. Song, L. J., Meng, F. R., & Yuan, G. (2016). Moving object location prediction algorithm based on Markov model and trajectory similarity. Journal of Computer Applications, 36(1), 39-43.
5. Xia, L., Huang, Q., & Wu, D. (2018). Decision Tree-Based Contextual Location Prediction from Mobile Device Logs. Mobile Information Systems, 2018.
6. Scellato, S., Noulas, A., Lambiotte, R., &Mascolo, C. (2011, July). Socio-spatial properties of online location-based social networks. In Fifth international AAAI conference on weblogs and social media.
7. Ghosh, A., Parikh, J., Sengar, V. S., &Haritsa, J. R. (2002, January). Plan selection based on query clustering. In VLDB'02: Proceedings of the 28th International Conference on Very Large Databases (pp. 179-190). Morgan Kaufmann.
8. Haritsa, V. S. S. J. R. (2002, December). Plan Selection based on Query Clustering. In Proceedings 2002 VLDB Conference: 28th International Conference on Very Large Databases (VLDB) (p. 179). Elsevier.
9. Gariel, M., Srivastava, A. N., &Feron, E. (2011). Trajectory clustering and an application to airspace monitoring. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1511-1524.
10. Morris, B., & Trivedi, M. (2009, June). Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 312-319). IEEE.
11. Li, Z., Lee, J. G., Li, X., & Han, J. (2010, April). Incremental clustering for trajectories. In International Conference on Database Systems for Advanced Applications (pp. 32-46). Springer, Berlin, Heidelberg.
12. Lu, J., Lu, Y., & Cong, G. (2011, June). Reverse spatial and textual k nearest neighbor search. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (pp. 349-360). ACM.
13. Taniar, D., &Rahayu, W. (2013). A taxonomy for nearest neighbour queries in spatial databases. Journal of Computer and System Sciences, 79(7), 1017-1039.
14. Cao, X., Chen, L., Cong, G., Jensen, C. S., Qu, Q., Skovsgaard, A., ...&Yiu, M. L. (2012, October). Spatial keyword querying. In International Conference on Conceptual Modeling (pp. 16-29). Springer, Berlin, Heidelberg.
15. Chen, L., Cong, G., Jensen, C. S., & Wu, D. (2013, January). Spatial keyword query processing: an experimental evaluation. In Proceedings of the VLDB Endowment (Vol. 6, No. 3, pp. 217-228). VLDB Endowment.
16. Rocha-Junior, J. B., &Nørvåg, K. (2012, March). Top-k spatial keyword queries on road networks. In Proceedings of the 15th international conference on extending database technology (pp. 168-179). ACM.
17. Wu, D., Yiu, M. L., Jensen, C. S., & Cong, G. (2011, April). Efficient continuously moving top-k spatial keyword query processing. In 2011 IEEE 27th International Conference on Data Engineering (pp. 541-552). IEEE.
18. De Felipe, I., Hristidis, V., &Rishe, N. (2008, April). Keyword search on spatial databases. In 2008 IEEE 24th International Conference on Data Engineering (pp. 656-665). IEEE.
19. Tao, Y., & Sheng, C. (2013). Fast nearest neighbor search with keywords. IEEE transactions on knowledge and data engineering, 26(4), 878-888.
20. Huang, W., Li, G., Tan, K. L., & Feng, J. (2012, October). Efficient safe-region construction for moving top-k spatial keyword queries. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 932-941). ACM.
21. Gao, Y., Qin, X., Zheng, B., & Chen, G. (2014). Efficient reverse top-k boolean spatial keyword queries on road networks. IEEE Transactions on Knowledge and Data Engineering, 27(5), 1205-1218.
22. Zhao, J., Gao, Y., Chen, G., Jensen, C. S., Chen, R., &Cai, D. (2017, April). Reverse top-k geo-social keyword queries in road networks. In 2017 IEEE 33rd International Conference on Data Engineering (ICDE) (pp. 387-398). IEEE.
23. Shang, S., Zheng, K., Jensen, C. S., Yang, B., Kalnis, P., Li, G., & Wen, J. R. (2014). Discovery of path nearby clusters in spatial networks. IEEE Transactions on Knowledge and Data Engineering, 27(6), 1505-1518.
24. Li, Z., Lee, J. G., Li, X., & Han, J. (2010, April). Incremental clustering for trajectories. In International Conference on Database Systems for Advanced Applications (pp. 32-46). Springer, Berlin, Heidelberg.
25. Chung, K. M., Kao, W. C., Sun, C. L., Wang, L. L., & Lin, C. J. (2003). Radius margin bounds for support vector machines with the RBF kernel. Neural computation, 15(11), 2643-2681.
26. Son, Y., & Lee, Y. (2017). Offloading Method for Efficient Use of Local Computational Resources in Mobile Location-Based Services Using Clouds. Mobile Information Systems, 2017.
27. Nir, M. (2015). Scalable resource augmentation for mobile devices (Doctoral dissertation, Carleton University).
28. Gao, X., Wen, J., & Zhang, C. (2019). An Improved Random Forest Algorithm for Predicting Employee Turnover. Mathematical Problems in Engineering, 2019.
29. Long, W., & Xu, S. (2016, October). A novel grey wolf optimizer for global optimization problems. In 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (pp. 1266-1270). IEEE.
30. https://www.cs.utah.edu/~lifeifei/SpatialDataset.htm.
31. Yeung, K. Y., &Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics, 17(9), 763-774.
32. Shang, S., Yuan, B., Deng, K., Xie, K., & Zhou, X. (2011, November). Finding the most accessible locations: reverse path nearest neighbor query in road networks. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 181-190). ACM.

AUTHORS PROFILE

G. Dona Rashmi, Research Scholar, Bharathiar University. E-mail:donamca08@gmail.com .

Dr.V. Narayani, Assistant Professor, Department of Computer Science, st. Xavier's college, Tirunelveli.