A Comparative Study on Real Time Path Planning using Reverse Path nearby Cluster Query (R-PNC) with Different Clustering Techniques in Spatial Networks

[1]G. Dona Rashmi, [2]Dr.V. Narayani

[1]Research Scholar, [2]Assistant Professor
[1]Bharathiar University.
[2]Department of Computer Science, st. Xavier's college, Tirunelveli.

Abstract: Planning of optimal paths in real time has been an open-ended challenge in several applications of independent systems. The challenge involving the computation of point-to-point shortest path in spatial networks is comprehensively analyzed with several techniques introduced for accelerating the computation process. Many of the current techniques simply assume that the weights (e.g., travel-time) of the network edges are invariable. The objective of this technical work is to perform an extensive and comparative analysis of the current path planning algorithms designed for Reverse Path Nearby Cluster (R-PNC) approach. This technical work examines and introduces the usage of another clustering mechanism for a new kind of spatial query known as Reverse Path Nearby Cluster (R-PNC) that searches to get the places with the most accessibility in road networks employing a suitable set of data sets. The newly introduced scheme makes use of the principle of data analytics for examining the most accessible places for a query provided and renders the comprehensive evaluation of the clustering approaches for various road networks. In order to validate the suitability of the proposed technique in optimal path planning, a set of experiments were performed. Also, few evaluations were carried out to examine in terms of the forms of the urban street of various cities in the world, specified as networks in the demographical area. The results of simulation show that a spatial analysis that depends on a series of the infrastructures of three cities and this scheme had the capability of considering the features of various kinds of clustering approaches, thus attaining a superior performance in terms of location recommendation.

***Keywords*- Path Planning, Reverse Path Nearby Cluster (R-PNC) Query, Clustering, Road Networks, Efficiency.**

I. INTRODUCTION

In day to day life, spatial networks such as road networks that link cities, rail networks that link railway stations, pipeline networks that bring water to homes are omnipresent. Spatial networks are a form of spatially enclosed graphs made by interconnecting spatial components like spatial lines and spatial points Yao (2000). In addition to their application in space based fields such as geography, cartography, Geographical Information Systems (GIS), and spatial database systems, these spatial networks are also used in transport and navigation like in GPS devices or in traffic prediction systems. Database support is required for the efficient usage of massive amount of data in any kind of spatial network Mennis and Guo (2009). However, until now, database support for spatial networks is quite uncommon and ignored. The current GIS realizations of spatial networks just make use of the database system for providing the network's elementary components such that an in-memory network can be constructed in a middleware layer.

Owing to the rapid evolution of geo-information technologies, several novel possibilities have come up. Hence, a more detailed analyzes can be carried out on spatial information. In this technical work, the probable application of spatial data mining (SDM) techniques is explored for the identification of factors, which may impact the way incident shappen Scellato et al. (2011).

These forecasts are helpful in, e.g., a presence service, rendering the location of the user available to other persons. Several other proactive applications, including early-reminder systems and traffic planning have emerged a possibility if the future position of the user is known. Spatial Data Mining refers to the procedure of finding exciting and earlier unknown but probably meaningful patterns from massive spatial datasets Hocaoglu and Sanderson (1996). Spatial Database Management (SDBS) are a form of database systems used for managing the spatial information i.e., point objects or spatially extended ones in a 2D or 3D space or in any high dimensional vector space. The process of knowledge discovery has gained increasing significance in spatial databases with rising massive quantity of data acquired from satellite images.

The ever-rising and popular online map applications and their extensive implementation in mobile devices and car-navigation systems have led to the number of user's increasingly looking out for point-to-point quickest paths and the respective travel-durations Harabor and Botea (2008).

With regard to static road networks where there are constant edge costs, this issue has been elaborately analyzed and several effective speed-up approaches have been designed for computing the quickest path in just few milliseconds. The static quickest path techniques simply assume that the travel-time for every edge of the road network is invariant (e.g., in ratio with the edge length). But, practically, the travel-time on a section of a road is hugely dependent on the traffic congestion and, hence this is a function of time i.e., time-based Ghosh et al. (2002).

The shortest route (path) query is one of the usual queries posed in a spatial network. The objective of the task is to discover a route linking two points in a network with the least distance among every route existing between those two points Haritsa (2002). These shortest route queries are utilized for finding the driving directions between physical places automatically, for example, between two cities. Sophisticated systems are mostly engineered under the network form where the nodes and edges are enclosed in space Hulgeri and Sudarshan (2002). Few examples with relevance to space and whose topology alone does not have all of the information includes transportation and mobility networks, Internet, mobile phone networks, power grids, social and contact networks, and neural networks. Therefore, defining and comprehending the structure and the development of spatial networks is critical for several other disciplines Hulgeri and Sudarshan (2002). One significant effect of space on networks is that a cost exists with respect the length of edges that in turn imposes drastic influence on the topological arrangement of these networks.

One recent proposal belongs to the Reverse Path nearby Cluster (RPNC) query, developed for the retrieval the positions with the most accessibility among a group of places. With regard to Location recommendation, the R-PNC query can be utilized for finding the optimum position in the road network Reddy and Haritsa (2005).

The R-PNC query poses a big challenge owing to three causes. At first, the association between the trajectories and data points (i.e., the minimum distance between a trajectory and a data point) is taken into consideration in this query. Forgetting the minimum distance between trajectory and data point *o*, each sample point has to be taken into account. This exudes more complexity compared to the association taken into account in the Reverse Path Nearby Cluster (R-PNC) search (i.e., the minimum distance between two diverse data points). Secondly, in the case of R-PNC, the query input is actually a list consisting of candidate location, not just one single point Hulgeri and Sudarshan (2002).The models of traditional single point query do not have efficient pruning approaches to deal with the issue(e.g., with *m* query locations given, the query must be executed *m* times to get the most accessible positions). Thirdly, the movement of the objects is restrained in road networks, in contrary to an open space.

Clustering refers to the process of classifying a set of objects in such a manner that the objects present in the same set have more similarity between themselves in some way compared to those present in other sets. It is utilized in several disciplines of research such as data mining, statistical data analysis, machine learning, pattern identification, image analysis and information retrieval.

The solution to the problem of clustering cannot be obtained with one particular algorithm but it needs different algorithms, which have considerable difference in their idea of what constitutes a cluster and the means of efficiently finding them Shang et al. (2014). Usually, the clusters consists of groups having lesser distances among the members of the cluster, dense regions of the data space, intervals or specific statistical distributions. This way, clustering can be expressed in the form of a multi-objective optimization problem. The suitable clustering algorithm and parameter settings are dependent on every single data set and the expected usage of the outcomes. Cluster analysis, by itself is an iterative procedure of carrying out knowledge discovery or interactive multi-objective optimization.

The objective of this technical work is the analysis and investigation of different popular and significant path planning clustering algorithms, such as Enhanced Density Clustering (EDC), Distribution Clustering and Trajectory Clustering and at last, the results of simulation yields the best shortest path for a certain query applying these approaches. With a trajectory data-set and a list of candidate location given by users, in case a location o refers to the Path Nearby Cluster (PNC) of k trajectories, then the influence-factor of o is specified as k and the R-PNC query retrieves the location having the greatest influence-factor. In this article, the R-PNC query combined with the clustering approach has been used in the street planning application for achieving the shortest path with efficiency.

The remaining portion of the technical work is organized as given. In section 1, the spatial data and its significance with regard to location recommendation employing Reverse path nearby cluster search and clustering approaches are studied. Section 2 provides an overview on few clustering approaches and query processing techniques used for spatial network data. In section 3, the various clustering approaches used for efficient location recommendation for a certain RPNC query is provided. Section 4 present the results of simulation and its analysis. Section 5 provides the conclusion.

II. LITERATURE REVIEW

Lu et al. (2017), introduced a new framework for efficient route planning to serve multiple requests submitted by the user. The framework comprises of two important modules, which include the planning module, in which four techniques in corporated with pruning and caching mechanisms are introduced for preliminary route planning, and the next refinement module, in which two refinement approaches are suggested for further improvement of the route quality. As far as the best of knowledge goes, this becomes the first work carried out on route planning, which takes multiple services rendered by a POI and multiple requests submitted by a user, into consideration at the same time. At last, an elaborate experimental analysis is performed using three real-world POI data sets and remarkable performance is achieved.

Chen et al. (2010), introduced a scheme, which first identifies the important locations where an individual may leave from or go to with the help of a clustering-based algorithm known as FBM (Forward–Backward Matching), then extracts the trajectories on the basis of a space partitioning technique, and at last, obtains the patterns of movement from the trajectories abstracted employing an improved CRPM (Continuous Route Pattern Mining) algorithm.

The movement patterns obtained is arranged in terms of the origin–destination couples. The prediction is done on the basis of on a pattern tree constructed from these patterns of movement. The results indicate that this technique can be of much help in achieving nearly 80% and 60% accuracy with regard to destination prediction and 1-step prediction, correspondingly, and provides an average deviance of nearly 60 m.

Kang and Yong (2010), designed a novel scheme that makes use of the frequent trajectory patterns for location prediction. With the help of line simplification and clustering, the newly introduced technique makes the trajectories simple and then clusters them into spatio-temporally useful areas. Once the discretization of the actual trajectories are performed into the sequences utilizing regions, trajectory patterns are extracted from discretized sequences employing a prefix-based projection technique. After this, a tree-structured prediction model is constructed with the help of these patterns, allowing a meaningful indexing of the patterns extracted in order to get the best match. Through Experimental analysis, it has been shown that the approach is quite efficient in finding the trajectory patterns, making the prediction of a future position with accuracy even if the query time is much distant in the future.

Choudhury et al. (2016), introduced a relevant query for spatial-textual databases, which gets an optimal position, and a group of keywords maximizes the size of bichromatic reverse spatial textual k closest neighbors (MaxBRSTkNN). Several practical applications are possible with this query involving social media advertisements where a restricted number of intended advertisements are shown to every user. Here, the challenge is to discover the position and the textual contents that need to be added in an advertisement such that it will be shown to the maximum possible number of users. The rising accessibility towards spatial-textual sets helps in resolving these queries related to both spatial neighborhood and textual similarity.

Zhao et al. (2017), recommended support for a novel kind of query, which is the Reverse Top-k Geo-Social Keyword (RkGSK) query. This query considers both the spatial, textual, and social data, and gets potential customers for geo-labelled objects. Also a hybrid index, referred as the GIM-tree is proposed, which performs the indexing of the positions, keywords, and social information about geo-labelled users and objects, and then, making use of the GIM-tree, effective RkGSK query processing algorithms, which make the best use of various pruning approaches are presented. The efficiency of RkGSK retrieval is defined through a case analysis, and elaborate experiments employing a ctual datasets provide perspective about the efficacy of the novel index and algorithms.

Shang et al. (2012) introduced a Path Nearest Neighbor based on Compressed Trajectories (PNN-CT query). This kind of query is executed on compressed trajectories and the objective is the retrieval of the PNN with the greatest probability (lossy compression results in unsureness) that, in turn, can get considerable advantages to users in several common applications like trip planning. In this, a two-phase solution for compressed trajectories (PNN-CT) query will be utilized with extreme efficiency and care. At first, the meta-data and sample points are used for specifying a stringent range of search.

The primary aspect of this stage is that the number of data objects/trajectory sections that need to get processed or decompressed has be maintained as less as possible. Secondly, a reconstruction algorithm that depends on probabilistic models is proposed in order to take the uncertainty into account during the decompression of the trajectory sections present in the candidate set. Elaborate experiments carried out on original and manmade trajectory data in road networks prove the efficacy of the newly introduced PNN-CT query processing.

Safar (2008) suggested a solution to the problem of Group K-Nearest Neighbors (GKNN) queries in spatial network databases, and a new technique that depends on actual network distances is recommended. This technique primarily utilizes the network Voronoi diagram characteristics along with an advanced incremental network extension to decide the inside network distances, which are required for getting the GKNN queries.

Wang et al. (2018) introduced a new problem of finding the accessible positions in spatial networks employing region-dependent geo-social data. With a set Q consisting of query regions, the top-k Accessible Location Discovery Query (kALDQ) discovers k positions, with the greatest spatial-density associations with Q.

Due consideration is given to both of the spatial distances between the positions and areas and the POI (point of interest) density within the areas. There are three problems faced in kALDQ: (1) the way of modelling the spatial-density correlation in real life, (2) the way to prune the search space with efficiency, and (3) the means of scheduling the searches from several regions of query. At last, the a new three-phase solution is proposed with a pair of upper and lower threshold limits of the spatial-density association and a heuristic scheduling mechanism for scheduling the regions of multiple query.

Gotoh and Okubo (2016) suggested reverse k-nearest neighbor (RkNN) searching techniques, which make use of the opposite direction along with the position between the query and targeted objects. Then a searching technique is evaluated for a bichromatic reverse k-nearest neighbor (BRkNN), which contains objects and queries in spatial networks. This proposed technique searches for the BRkNN of the query employing an influence zone for every object using a Network Voronoi Diagram.

Tianyang et al. (2019) demonstrated a new algorithm for direction-sensitive KNN (DAKNN) queries for objects moving in a road network. As a first step, in this technique, R-tree and simple grid are utilized in the form of the base index structure, where the R-tree is utilized to index the static road network and the simple grid is utilized for the indexing of the mobile objects. After this, the idea of “azimuth” is presented for representing the direction of the objects’ movement in a road network, and suggests a new local network extension technique to rapidly predict the direction of the objects movement.

By accounting whether an object is moving farther away from or coming nearer to a query point, the object, not certainly in the KNN result set is removed ultimately. This way, the communication cost is reduced, and at the same time, helps in simplifies the computation involved in shifting the direction between the objects in movement and the query point. Elaborate experiments are carried out and the results indicate that this algorithm can help in achieving real-time and effective queries for retrieving the objects travelling towards the query point in a road network.

Rocha-Junior et al. (2012) designed a top-k spatial keyword queries on road networks in which the shortest path is actually the distance between the query location and the spatial object. The novel type of query is formalized, and new indexing structures and algorithms, which have the capability of process these queries with efficiency are presented. At last, an experimental assessment, which demonstrates the efficacy of this technique is performed.

De Felipe et al. (2008)suggested an effective technique for top-k spatial keyword queries. In this research work, an indexing structure known as IR2-Tree (Information Retrieval R-Tree) is introduced, combining an R-Tree and superimposed text signatures.

Also, algorithms, which build and maintain an IR2-Tree are presented, and it is used for getting an answer to the top-k spatial keyword queries. A comparative analysis of these newly introduced algorithms and the current techniques are presented and it demonstrates the superiority of performance and remarkable scalability.

Tao and Sheng (2013) designed a novel access technique known as the spatial inverted index, which improves the classical inverted index to deal with multidimensional data, and includes algorithms, which can provide answer to theclosest neighbor queries with keywords just in real time. The SI-index is quite economical in terms of space, and also has the capability ofcarrying out keyword-supported closest neighbor search in a time span at the order of couple of milli-seconds. Through experiments it has been proven that the proposed approaches perform better than the IR2-tree in query response time considerably, frequently by a factor of few orders of magnitude.

Gao et al. (2014) presented few novel kind of queries, called as, Reverse top-K Boolean Spatial Keyword (RkBSK) retrieval, which makes the assumption that the objects are on the road network and takes both spatial and textual data into consideration. With a data set P given on a road network and a query point q with a group of keywords, an RkBSK query finds the points in P, which have q to be one of the points as an answer for their top-k Boolean spatial keyword queries.

The RkBSK query is formalize and then filter-and-refinement framework based algorithms is proposed for providing an answer to the RkBSK search with random k and no prior computation involved. In order to speed up the query process, various new pruning heuristics, which use both spatial and textual data are used for reducing the search space with efficiency. Moreover, a novel data structure known as count tree has been designed for improving the query performance further. An experimental analysis employing both actual and artificial data sets show the efficiency of the novel pruning heuristics and the performance of the newly introduced algorithms.

III. ANALYSIS OF CLUSTERING TECHNIQUES ALONG WITH THE REVERSE PATH NEARBY CLUSTER (R-PNC) SEARCH

It is quite a challenge to discover the regions of interest in big cities. In this research work, a new query known as the Reverse Path Nearby Cluster (RPNC) query is analyzed and investigated and it discovers the regions of prospective attention (e.g., places for sightseeing and commercial districts) with regard to a user-chosen travel route.

At first, this R-PNC query is formulated by the clustering approach and after this, the RPNC gets reversed and ordered based on their greatest density distributions and later scanned from the maximum down to the minimum. In this research work, three kinds of clustering approaches such as, Enhanced Density Clustering (EDC), Distribution Clustering and Trajectory Clustering are explored for assessing the performance of clustering techniques. For a certain data set P on a road network and a query point q having a group of keywords, a Reverse Path Nearby Cluster (R-PNC) query restores the points in P, which have q to be one of the points as the answer for their top k -RPNC queries. This newly introduced work provides highly effective clustering model for carrying out path planning in spatial networks.

3.1. Network Modeling

A spatial network model is formulated in the form of a connected and undirected graph G(VE, ED, FU, WE, CON), where VErefers to a vertex set and $ED \subseteq VE \times VE$ indicates an edge set. A vertex $v_i \in VE$denotes a road bifurcation or the end point of a road Rocha-Junior and Nørvåg (2012). An edge $e_k = (v_i, v_j) \in ED$is specified by two vertices and indicates a section of a road, which helps in travelling between vertices$v_i$and v_j. Function $FU : VE \cup ED \rightarrow Geometries$contains the records of the geometrical data of the spatial network G. Especially, it performs the mapping of a vertex and an edge to the point location of the respective road bifurcation and to a polyline indicating the associated road segment, correspondingly.

Function $WE: ED \rightarrow R$indicates a function, which designates a real-valued weight to every edge. The weight $W(e)$ of an edge e indicatesthelength of therespective road segment or anyother relevant characteristic like its travel time or fuel consumption, which may be acquiredthrough the mining ofpast traffic information. With two vertices $p_a$and p_b in a spatialnetwork, the network shortest path between them (i.e., a sequence of edges connecting$p_a$and $p_b$where the accumulatedweight is very less) is represented by $SP(p_a, p_b)$ and its length is expressed by $Sd(p_a, p_b)$.

- **Problem Definition 1**

With a query qe(So, key), a parameter k, and a dataset 'P' with every POI $p \in P$ in the form of (So, key), let $P_{qe.key}$refer to the set of POIs in P containing qe.key, i.e., $P_{qe.key}$= {p ∈ P |qe.key⊆p.key}. A Top k PN query (on the road network) submitted at q, represented as TkPN(qe), retrieves the k POIs in $P_{qe.key}$with the minimal network distances to q, given in formal expression as, TkPN(qe)={S ⊆ $P_{qe.key}$||S|= k ∧ ∀ s ∈ S, ∀p∈($P_{qe.key}$-S), ||qe,s||≤||qe,p||}. For any data point present in TkPN(qe), that it is one among the Boolean spatial keyword closest neighbors of 'qe'.S refer to the set of POIs for user provided query.

- **Problem Definition 2**

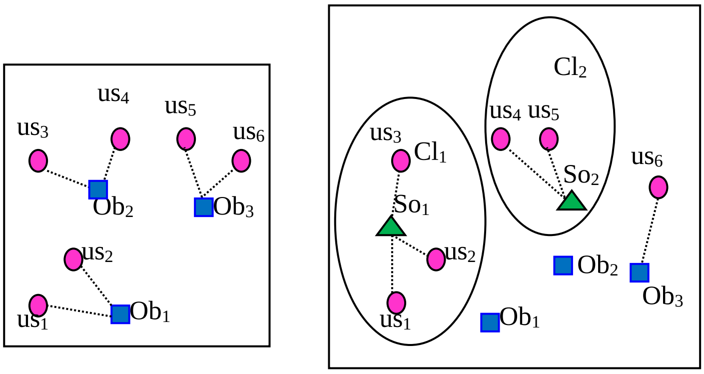
The top-k Reverse Path Nearby query is reversed on the road network. With a query qe(So, key) given, a parameter k, and a dataset P, an TkRPN query (on the road network) submitted at qe, represented as TkRPN(qe), provides all the POIs in P whose top-k spatial keyword queries consists ofqe, given in formal notation as, TkRPN (qe) = {p ∈ P | q ∈ TkPN (p)}. A reverse query route is basically a set of vertices $(p_1, p_2, \dots, p_n)$, where p_i and p_{i+1}form theneighboring vertices in G, i = (1, 2, . . . ,n – 1). With a reverse query route 'qe' and a vertex 'p' in a spatial network, the distance d(p,qe) between them is given in equation (2).

3.2. Problem Specification

The predominant type of nearby query is the reverse of nearby queries, which concentrates on the inverse relationship among the points. A Reverse Path Nearby(RPN) query$'qe'$ is aimed at finding all the objects for which qe is their closestto user query. The formal definition of a RPN query is givenas follows.

Provided a set of objects $'\mathrm{Ob}'$ and a query object qe, a Reverse Path Nearby (RPN) query is aimed at finding a group of objects RPN such that for any object ob ∈ Ob and r ∈ RPN, dist(r,qe)≤ dist(r,ob). The RPN set of a query qe might be either empty or may include one or multiple objects.

For instance, a RPN query may request the group of clients impacted by the inauguration of a new store outlet location so that the relevant clients can be informed. This query can also be utilized for identifying the place that increases the number of prospective client. Suppose another example, an RNN query may be submitted to retrieve the store outlets, which get affected by the opening of a new store outlet at some particular place. It is to be noted that in the first example, two unique sets (stores and customers) interested in RNN query exists while in second example, just one set (stores) exists. A bichromatic query (the first example) is aimed at finding the RPN where the base data set comprises of two different sorts of elements. A monochramtic RPN query (the second example) is aimed at finding the RPN where the road side network has just one kind of data objects. In this research work, the bichromatic query based RPN is focused on, and itretrieves the optimal area to keep an element'Ob' such that the cardinality of the BRPN query results provided by 'Ob' in 'SO' (Spatial object) is increased.



(a) (b)

Identifier	Keywords
us_1	Pizza, tea
us_2	Pizza
us_3	Pizza, Coffee
us_4	Coffee
us_5	Coffee, tea
us_6	tea
Ob_1	Pizza
Ob_2	Coffee
Ob_3	Tea
So_1	Second Raja Street
So_2	Fourth Raja Street
So_3	Sixth Raja Street

Figure 1: Example of a Restaurant of Reverse Path Nearby Cluster (RPNC) Query

Consider a restaurant application, which retrieves the optimal place for starting a new restaurant, and the items that are to be included in the menu so that it will become a top-k restaurant for a maximum number of clients. In Figure 1a, the users us_1,...,us_6 are represented by circles, and ob_1 ,ob_2 and ob_3 indicate three restaurants. Table 1 illustrates the respective descriptions associated with the items that are given in menu. In this, the top first spatial-textual relevant restaurant for every customer is depicted with a dotted line that connects. Assume a service provider desires to open a new restaurant, Ob_x in one of the spatial objects So_1,So_2,So_3represented with triangles as shown in Figure 1b, and for reasons related to cost, the number of menu items, which can be exhibited is ‘1’. The items of choice provided in the menu items include {‘Pizza’, ‘Coffee’, and ‘Tea’}. If Ob_x is kept in So_1, and the menu is ‘Pizza’’, Ob_x goes on to become a top-1 relevant restaurant for us_1, us_2, and us_3, as illustrated in Figure 1b. As for the provided preferences of locations and keywords, and the other competing restaurants, Ob_x can become the top-1 relevant restaurant for a maximum of three users. Therefore in this example, the optimal position and menu item are Ob_1 and ‘Pizza’, correspondingly.

3.3. Clustering Techniques

In this technical work, the R-PNC query optimization given has been carried out with the help of three diverse Clustering approaches such as, Enhanced Density Clustering (EDC), Distribution Clustering and Trajectory Clustering. All these approaches will be discussed in the section as follows.

3.3.1. Enhanced Density Clustering

In this research work, Enhanced Density Clustering (EDC) can automatically decide on a suitable close partition value. At first, it is begun with a random value of partition. In case it is not able to identify a cluster, it increments the nearby value to 0.5. Also, if during an iteration, more than 10% of data objects that are similar has been detected, it is taken that a cluster has been found Dona Rashmi and Narayani (2019). Then that particular cluster has been maintained separately and removed from the primary dataset. The algorithm, then again increments the adjacent cluster spatial data object and also the partition values for detecting the next subsequent cluster. In this manner, once 95% data has been done with, the algorithm presumes that every cluster has been successfully identified. After this, the rest of the spatial data points will be displayed as noise points, and the clusters that are saved will be plotted.

In this, ‘cl’ refers to a cluster with center at vertex p, and the scanning of p has been done by network extensions running from p_i and p_j. When a vertex p gets scanned by the network extensions running from both p_i and p_j, the upper and lower bounds of its distance to the query route ‘qe’ is computed. After this, the spatial-object density for the clusters with center at vertex dp is estimate. The number of spatial objects enclosed by the circular area (p,thr.r) is computed in prior, represented by p.max. In case the value of p.max is lesser than the size threshold of the cluster thr.s, the clusters with center at p can be pruned effectively. Else, the density for clusters with center at p is computed as given

$$cl_h.\rho = \max_{cl.m=p \wedge cl.r<thr.r \wedge cl.s>thr.s} \{cl.\rho\} \geq \frac{p.max}{\sum_{\forall ed \in (p,thr.r)} ed.we} \&con \tag{1}$$

Here, cl refers to a cluster with center at p (i.e., cl.m = p), and ‘cl_h’ refers to the cluster having the highest density among all the qualifying clusters with center at p. the upper bound of the density-evaluation score is estimated as below

$$E_{dis}(cl_h) < \frac{2}{1+e^{\left(p\cdot\frac{n}{\sum ed.we}\right)}} - 1 = E_{dis}(cl_h).ub.con \tag{2}$$

By merging the upper bounds of the distance-evaluation score E_s(cl, qe).ub and the density-evaluation score $E_s(cl_h)$.ub, the upper bound of the distance-and-density evaluation score E_{sdis}(cl, qe). ub is estimated as given

$$E_{sdis}(cl_h, qe) = \lambda.E_s(cl, qe).ub + (1-\lambda).E_{dis}(cl_h).ub.\text{con} \tag{3}$$

Where c l_h refers to the cluster having the highest density with centre at p. Among all vertices, whose scanning has been done by the network extensions from both p_i and p_j, a global upper bound UB is defined as below,

$$UB = min\{E_{sdis}(cl_h, qe).ub.con\} \tag{4}$$

where the center of cluster cl_h forms a vertex, whose scanning is done the network extensions running from both p_i and p_j. Just like LB, UB varies dynamically during the processing of RPNC query Dona Rashmi and Narayani (2019). At last, the cluster having the highest distance-and-density evaluation score is discovered.

Step 1. Dijkstra’s algorithm is utilized for network extension for computing the network distance dis(p,qe)between a vertex p and a query route qe. Dijkstra’s algorithm always chooses the vertex having the least weight for extension, therefore the first vertex v ∈qethat is scanned by the network extension from p is the vertex nearest to p, and dis(p, qe)=sdis(p, ve).

Step 2. To get the cluster with the greatest density among all the clusters with center at p, it is extended from p as per Dijkstra's algorithm. The sub graph in the scanned area is extended one step at a time, and at every step, its current density is computed and recorded as

$$cl.\rho = \frac{\sum_{p\in cl.VE} p.g}{\sum_{ed\in cl.ED} ed.we} \tag{5}$$

Where cl.r refers to the density of the area that has been scanned till now, p refers to a vertex in the scanned area, and $\Sigma p.g$indicates the number of spatial objects that are enclosed by the scanned area. When the radius of the scanned region goes beyond the cluster-radius threshold thr.r, then network expansion stops, and the cluster having the highest density is retrieved.

3.3.2 Distribution Clustering

A popular presumption is that scalable query processing can only be attained in a distributed environment. In this research work, the probability distribution of the closest neighbor distances of a cluster is analyzed. This analysis depends on the presumption that the points present inside of a cluster are distributed evenly, i.e. the points of a cluster are distributed in the form of a homogeneous Poisson point process confined to a specific portion of the data space Dona Rashmi and Narayani (2019). A major precondition for holding this presumption is that the global query optimization can happen. Hence, control on the nodes that participate is required.

To decide the probability distribution of the distances of the closest neighbor. Suppose the *N* points be evenly distributed over a data space R with a volume *Vol(R).*Suppose that these *N* points "pass" independently into the data space *R*, so that the probability of one of these *N* points getting into a subspace *S* with volume *Vol(S)*is equivalent to *Vol(S)/Vol(R).* The probability that the closest neighbor distance *D* from any query point *q* to its closest neighbor in the space *R* is higher than some *x* is therefore equivalent to the probability that none of the *N* points is positioned within a hypersphere around *q* with radius *x*, represented by *SP(q,x):*

$$P(D > x) = (1 - Vol(SP(q,x))/VOL(R))^N \tag{6}$$

As a result, the probability that D is not higher than x is:

$$P(D \leq x) = 1 - P(D > x) \tag{7}$$

$$= 1 - (1 - Vol(SP(q,x))/VOL(R))^N \tag{8}$$

In 2-dimensional space, the distribution function is henceforth:

$$F(x) = P(D \leq x) \tag{9}$$

$$= 1 - (1 - \pi x^2/Vol(R))^N \tag{10}$$

Here, the distribution function consists of two parameters *N* and *Vol(R).*When it is pretty straight to decide *N*. For concluding the discussion above, it is stated that the specification of a cluster depending on the distribution of the closest neighbor distance set is as given:

Definition 3 (cluster) Let *DB* refer to a set of points. A *cluster C* is a non-empty subset of *DB* having the following characteristics:

(1) *NND ist Set(C)* demonstrates the expected distribution with a necessary confidence level.

(2) *C* is *maximal*, i.e. every expansion of C by neighboring points does not satisfy condition (1). (maximality).

(3) *C* is *connected*, i.e. for every pair of points (a,b) of the cluster there is a path consisting of occupied grid cells that connect a and b (connectivity).

- **General process for Distribution Clustering**

Distribution Clustering is an iterative algorithm, i.e. the allocation of a point to a cluster is only dependent on the points that are processed till now without taking the entire cluster or even the entire database into consideration. Distribution Clustering iteratively improves an initial cluster by its neighboring points until the closest neighbor distance set of the resultant cluster still suits the anticipated distance distribution Dona Rashmi and Narayani (2019). A candidate is a point that not yet belongs to the present cluster that needs to be checked for probable membership in this cluster. The process of candidates' generation and testing is explained as follows.

- **Generating Candidates**

The group of candidates in a cluster is built with the help of region queries that can be efficiently aided employing spatial access techniques. A region query retrieves all the objects in the database that intersect the particular query area, e.g. a circle. For every new member *p* of the present cluster *C*, the new candidates are retrieved with a circle query having a desirable radius *m*. This radius is selected in such a way that for none of the points in the cluster a much greater distance to the closest neighbor is to be anticipated. A higher *m* would result in more number of candidates for the c2-test and the efficiency of the algorithm is reduced. Again, the computation of *m*, again, is dependent on the model of evenly distributed points within a cluster *C*. Let *A* refer to the area of *C* and *N* indicate the number of its objects.

A condition required for *m* is:

$$N \times P(NNdist_C(p) > m) < 1 \tag{11}$$

And using this formula for the equation (14) as,

$$(1 - \pi m^2/A)^N < 1/N \tag{12}$$

Subsequently, it is needed

$$m > \sqrt{A/\pi.(1 - 1/N^{1/N})} \tag{13}$$

During the insertion of a new point *p* into cluster *C*, a circle query with center *p* and radius *m* is carried out and the resultant points are taken as new candidates.

- **Testing Candidates**

The iterative scheme of distribution clustering involves an intrinsic dependency of the clustering discovered from the order of generation and testing of candidates. When the distance fixed of the entire cluster might suit the anticipated distance distribution, this is not necessarily true for all of subsets present in this cluster. Therefore, the sequence of testing the candidates is critical.

Candidates that are not allowed by the test when taken for the first time are known as unsuccessful candidates. In order to reduce the dependency on the sequence of testing, the distribution clustering includes two significant features:

(1) Unsuccessful candidates are not removed but are attempted again at a later point of time.

(2) Points that are already allocated to some cluster may move to another cluster at a later point of time.

Unsuccessful candidates are not removed but they are stored. Once every candidate of the present cluster has been processed, the unsuccessful candidates of that particular cluster are taken into account once more. In several scenarios, they will now suit the distance distribution of the improved cluster. For many of these candidates, a c2value is acquired that is considerably greater than the threshold value indicating that this candidate is not allocated to the cluster. However, when carrying out the c2 test for the distance set of the whole cluster, a c2 value considerably lesser compared to the threshold is acquired implying that the distance set of the cluster is indeed a fit for the anticipated distance distribution.

Even when none of the unsuccessful candidates pass the test separately, it may be probable that some bigger subset of the group of unsuccessful candidates suits the distance distribution of the present cluster. It is also probable that some point may be moved to another cluster many times prior to its assignment to its last cluster. However, on the average case, the number of reassignments made for certain point is found to be considerably less. The algorithm stops due to the following characteristics. First, each point of the database is chosen at most once as a starting point for the generation of a new cluster. Secondly, generating a single cluster stops since if there are no more candidates, the unsuccessful candidates are not taken into consideration once more, if none of them is a fit for the present cluster. While the candidates are generated, the distribution clustering does not verify if a candidate already has been allocated to any other cluster or not.

3.3.3. Trajectory Clustering

Trajectory clustering refers to the extraction procedure of the similarity, anomaly and valuable patterns out of the trajectory data. The aim of trajectory segmentation is to segment the entire trajectory into sub-trajectories in which the targets move with the similar movement feature. Therefore, the key factor to the segmentation process is to get the point where the target's motion feature varies quickly Dona Rashmi and Narayani (2019). The angle between two track sections can depict the movement style. This technique includes two steps for efficient detection of a location employing a Reverse Path Nearby Cluster Query.

Micro-clustering is the first step. As an infinite amount of data source exists, it is not possible to have all the preprocessed input data stored and the clusters computed from them as per request. In order to resolve this problem, this newly introduced technical work presents the principle of trajectory micro clusters. The term "micro" stands for the extreme closeness of the clusters. The concept is to just create very fine granular clusters. Therefore, the number of micro-clusters is much higher compared to that of the final trajectory clusters. The second step involves macro-clustering that will be explained in the section below. In comparison with the micro-clustering step that are updated regularly as and when new data is obtained, the macro-clustering step is just initiated after the receipt of the user's request on trajectory clusters. Then this step will make use of the micro-clusters in the form of input.

- **Trajectory Micro-Clustering**

Since the newly obtained trajectories will just impact the result of local clustering, trajectory micro clusters (or just micro-clusters) are presented here to keep up a fine-granular clustering. Micro-clusters exhibit much more restriction compared to the last clusters in the aspect that every micro-cluster is supposed to just maintain and summarize the information about local partitioned trajectories Dona Rashmi and Narayani (2019). Micro-clustering will facilitate more effective computation of final clusters in comparison with computation from the actual line segments.

Micro-Cluster Definitions: Every trajectory micro-cluster will maintain and summarize a group of divided trajectories that are primarily line segments.

- **Definition 1 (Micro-Cluster)**

A trajectory micro-cluster (or micro-cluster) for a group of directed line segments $L_1, L_2, \ldots, L_N$is indicates as the tuple: (N, LS_{center}, LS_θ, LS_{length}, SS_{center}, SS_θ, SS_{length}), where N refers to the number of line segments present in the micro-cluster, LS_{center}, LS_θ, and LS_{length}form the linear sums of the line segments' center points, angles and lengths correspondingly, SS_{center}, SS_θ, and SS_{length}refer to the squared sums of the line segments' center points, angles and lengths correspondingly.

The definition on trajectory micro-cluster extends the cluster feature vector. The linear sum LS indicates the fundamental summarized information of line segments(i.e., center point, angle and length). The square sum SS will be utilized for calculating the proximity of micro-cluster. The additive characteristic of the definition renders it the addition of new line segments into the micro-cluster to be easy and combine the two micro-clusters. At the same time, the definition is developed with consistency towards the distance measurement of line segments. In addition, each trajectory micro-cluster will include a characteristic line segment. Like it is suggested by the name, this line segment forms the characteristic line segment of the cluster. It is an "average" of some kind.

Definition 2 (Representative Line Segment). The representative line segment of a micro-cluster is denoted by the starting point s and ending point e. s and e can be calcualted from the micro-cluster features.

$$s = \left(center_x - \frac{cos\theta}{2} len, center_y - \frac{sin\theta}{2} len\right) \quad (14)$$

$$e = \left(center_x + \frac{cos\theta}{2} len, center_y + \frac{sin\theta}{2} len\right) \quad (15)$$

Where

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

Four line segments exists in the micro-cluster that are traced as thin lines. The characteristic line segment of the micro-cluster is traced as a thick line.

- **Creating and Updating Micro-Clusters**

On the receipt of a new line segment L_i, the first job is to get the nearest micro-cluster MC_k, which can intake L_i. In case the distance between L_i and MC_k is lesser compared to the distance threshold d_{max}, L_i is then included to MC_k and MC_k is updated in accordance; if not, a new micro cluster is generated. This section will explain the way in which these steps are carried out.

Just after this, the distance definition between a line segment and a micro-cluster is provided. As a micro-cluster includes its representative line segment, the distance is actually specified between two line segments, which consists of three elements: the center point distance (d_{center}), the angle distance (d_θ) and the parallel distance (dk). The distance is obtained from a similarity measure utilized in the field of pattern recognition, an improved line segment Hausdorff distance. The identical distance measure isalso brought into use. On the contrary, component d_{center} is used in place of $d_\parallel$. The reason behind choosing d_{center} is that it is more balanced between d_θ and $d_k$and it is convenient to use the concept of extent. Let s_i and e_i refer to the beginning and ending points of L_i; likewise for s_j and e_j with L_j .With no generality loss, the longer line segment is allocated to L_i, and the shorter one is assigned to L_j.

Definition 3. The distance function is defined to be the sum of three elements:

$$dist(L_i, L_j) = d_{centre}\,(L_i, L_j) + d_\theta\,(L_i, L_j) + d_\parallel(L_i, L_j) \quad (16)$$

The centre distance:

$$d_{centre}\,(L_i, L_j) = \|center_i - center_j\| \quad (17)$$

Where $\|center_i - center_j\|$refers to the Euclidean distance between center points of L_i and L_j

The angle distance:

$$d_\theta(L_i, L_j) = \begin{cases} \|L_j\| \times \sin(\theta), & 0^\circ \le \theta < 90^\circ \\ \|L_j\|, & 90^\circ \le \theta \le 180^\circ \end{cases} \quad (18)$$

where$\|L_j\|$represents length of L_j, $\theta(0^\circ \le \theta \le 180^\circ)$ stands for the smaller intersecting angle between $L_i$and L_j. It is to be noted that the range of $\theta$is not $[0^o, 360^o)$ asθrefers to the value of smaller intersecting angle without the direction considered.

The parallel distance:

$$d_\parallel(L_i, L_j) = min(l_{\parallel 1}, l_{\parallel 2}), \quad (19)$$

where$l_{\parallel 1}$refers to the Euclidean distances of $p_s$to s_i and $l_{\parallel 2}$is that of $p_e$to e_i. p_s and p_e stands for the projection points of the points s_j and e_j onto L_i correspondingly. Once the nearest micro-cluster MC_k are found, and in ase the distance from L_i is lesser compared d_{max}, L_i is added into it, and the linear and square sums in MC_k are updated in accordance. Since they are only sums, the additive property is applicable and the update is effective. In case the distance between the closest micro-cluster and L_i is greater than d_{max}, anew micro-cluster will be generated for L_i. The first level measures in the new micro-cluster is just obtained from line segment L_i (i.e., center point, theta, and length).

- **Merging Micro-Clusters**

If the total space used by micro-clusters exceeds a given space constraint, some micro-clusters have to be merged to satisfy the space constraint. Mean while, if the number of micro-clusters keeps increasing, it will affect the efficiency of algorithm because the most time-consuming part is finding the nearest micro-cluster. And what is most important, it may be unnecessary to keep all the micro-clusters since some of the micro-clusters may become closer after several rounds of updates. Therefore, the algorithm demands merging close micro-clusters when necessary to speed up efficiency and save storage. Obviously, pairs of micro-clusters that contain similar line segments are better candidates for merging because the merge results in less information loss. One way to compute the similarity between two micro-clusters is to calculate the distance between the representative line segments of the micro-clusters. Though intuitive, this method fails to consider the tightness of the micro-clusters.

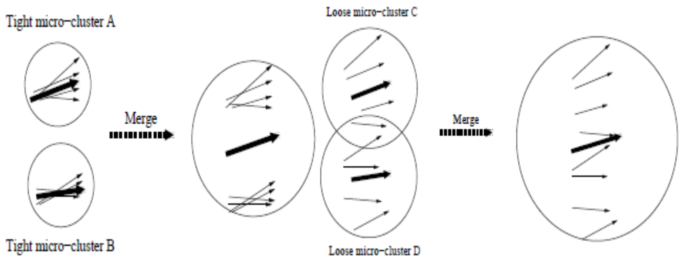


Figure 1(a): Merging Tight Micro-Clusters Figure 1(b): Merging Loose Micro-Clusters

Figure 1: Merging Micro-Clusters

Figure 1 shows an example that how tightness might effect distance between two micro-clusters. Figure 1(a) shows two tight micro-clusters and the micro-cluster after merging them. Figure 1(b) shows the case for two comparatively loose micro-clusters. See that micro-cluster A and micro-cluster C have same representative line segments, and so domicro-clusters B and D. Thus the distance between micro-cluster A and B should be the same as that between micro-clusters C and D if measure the distance only using representative line segments. In this case, the chance to merge micro-clusters A and Bis equal to that of merging micro-clusters C and D. However, actually prefer merging micro-clusters C and D. There are two reasons: on one hand, if both micro-clusters are very tight, they may not be good candidates for merging because it would break that tightness after the merge. On the other hand, if they are both loose, it may not do much harm to merge them even if their representative line segments are somewhat far apart. Hence, a better approach would be to consider the *extent* of the micro-clusters and use that information in computing the distance between micro-cluster.

- **Micro-Cluster Extent**

The extent of a micro-cluster is an indication of its tightness. Recall that micro-clusters are represented by tuples of the form: (N, LS_{center}, LS_θ,LS_{length}, SS_{center}, SS_θ, SS_{length}), which maintain linear and square sums of center,angle and length. The extent of the micro-cluster also includes three part $extent_{center}$,$extent_\theta$ and $extent_{length}$ to measure the tightness of three basic facts of a trajectorymicro-cluster. The extents are the standard deviation that calculated from its corresponding LS and SS.

IV. Results and Discussion

In this section, conduct extensive experiments on real and synthetic spatial data sets to study the performance of the developed algorithms https://www.cs.utah.edu. In this work the graphs extracted from three spatial networks, namely the San Francisco, North America and City of Oldenburg Road Network which contains nodes and edges to perform the effective prediction of location. The performance metrics are evaluated by using different road works with the help of three different clustering techniques effectively. The performance metrics are such as Rand Index (RI), Candidate Ratio and Query time is evaluated by using proposed clustering techniques. From the experimental result, conclude that the RPNC-TCEL system is more efficient than the existing system.

i) Rand Index

The similarity among two data clusters can be measured by a parameter called Rand Measure or Index. 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,

$$Rand\ Index = \frac{a+d}{a+b+c+d} \tag{20}$$

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. Increase in density decreases the candidate ratio.

$$Candidate\ Ratio = \frac{CS.size}{O.num} \tag{21}$$

$$Density\ of\ data\ points = \frac{O.num}{V.num} \tag{22}$$

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.

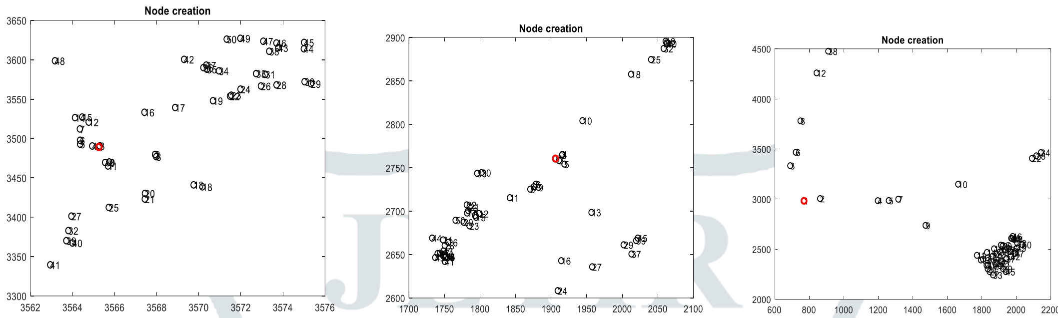


Figure 2(a): North America Figure 2(b): San Francisco Figure 2(c): Oldenburg

Figure 2: Node Creation of the Different Clustering Technique for Three Types of Cities of Road Network

The figure 2. Shows the Node creation of the clustering technique for three types of cities of road network for initializing process of the Reverse Path Nearby query based search. This is the starting process of RPNC query.

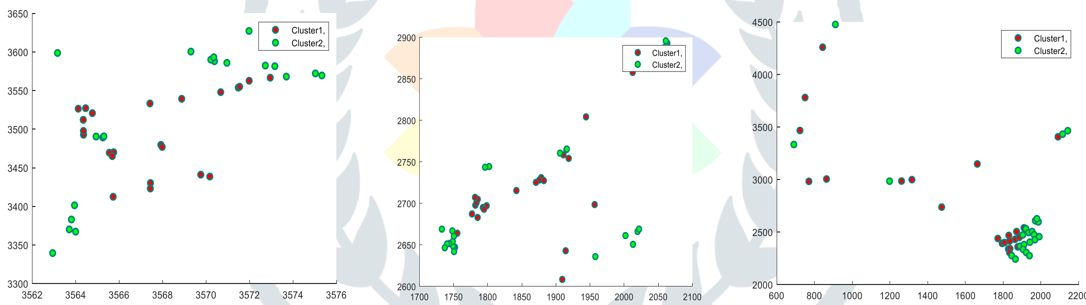


Figure 3(a): North America Figure 3(b): San Francisco Figure 3(c): Oldenburg

Figure 3: Formation of Clustering Process of the Proposed Three Techniquesfor Three Types of Cities

The figure 3. illustrates the formation of clustering process of the three techniques for Reverse Path Nearby query based search.

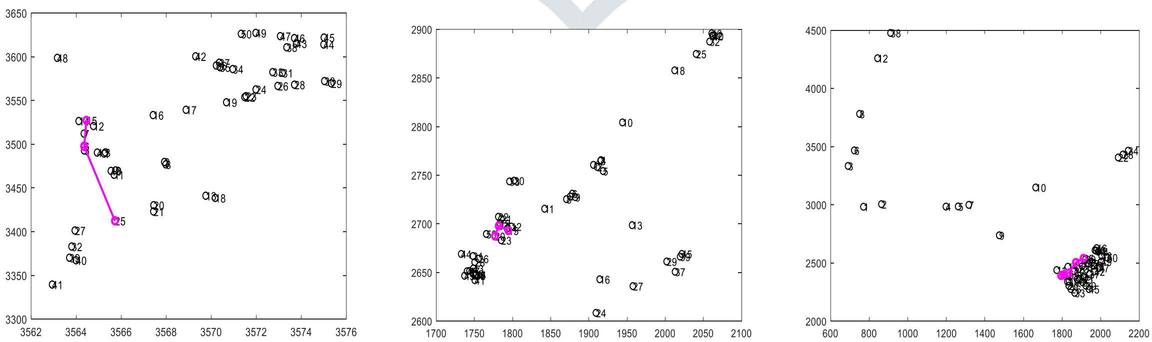


Figure 4(a): North America Figure 4(b): San Francisco Figure 4(c): Oldenburg

Figure 4: Pathnearby Cluster Detection of The Three Clustering Techniques for Three Types of Cities

The Figure 4. Shows the path Nearby cluster detection of the three clustering techniques. This 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.

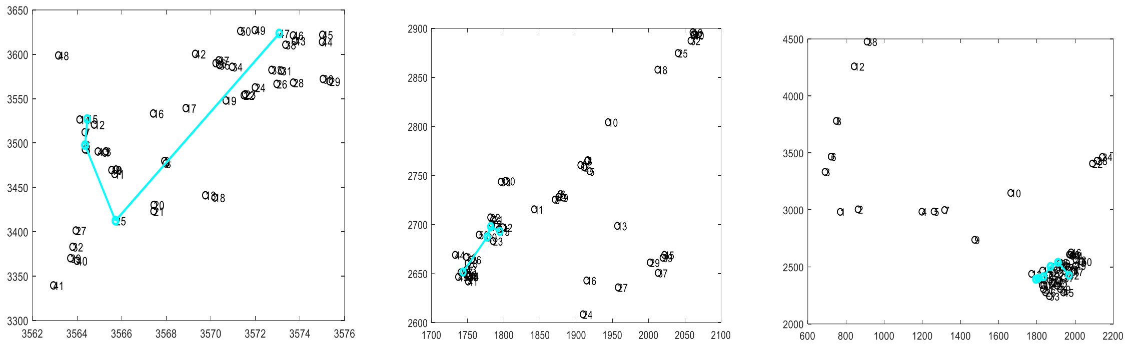


Figure 5(a): North America Figure 5(b): San Francisco Figure 5(c): Oldenburg

Figure 5.: Reverse Path Detection of The Three Clustering Techniques for Three Types Of Cities

The Figure 5. Shows the Reverse path detection of the three clustering techniques. This 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.

Methods	**Rand Index (RI)**		
	North America	**San Francisco**	**Oldenburg**
PNC-CCS	41.66	41.66	27.77
RPNC-IBSA	50.00	50.00	33.33
RPNC-DCSVM	62.5	62.5	41.66
RPNC-TCEL	78.12	78.12	52.08

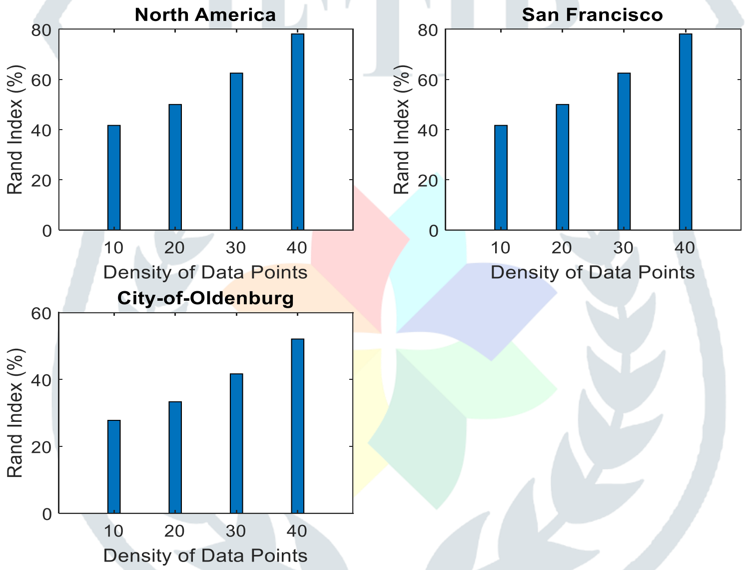


Figure 6: Accuracy Comparison of the Proposed Three Types of Clustering Algorithms for Three Types of Cities

Figure 6. Shows the accuracy comparison of proposed three types of clustering algorithms. From the above results it is concluded that high accuracy is produced by the proposed RPNC-TCEL algorithm for three given city of road networks.

Density of points / Methods	**Candidate ratio**											
	North America				**San Francisco**				**Oldenburg**			
	PNC-CCS	**RPNC-IBSA**	**RPNC-DCSVM**	**RPNC-TCEL**	**PNC-CCS**	**RPNC-IBSA**	**RPNC-DCSVM**	**RPNC-TCEL**	**PNC-CCS**	**RPNC-IBSA**	**RPNC-DCSVM**	**RPNC-TCEL**
10	10.71	4.285	17.85	33.57	10.71	12.14	14.28	33.57	10.71	9.28	11.42	12.85
20	12.00	4.800	20.00	37.60	12.00	13.60	16.00	37.60	12.00	10.40	12.80	14.40
30	15.00	6.000	25.00	47.00	15.00	17.00	20.00	47.00	15.00	13.00	16.00	18.00
40	20.25	8.100	33.75	63.45	20.25	22.95	27.00	63.45	20.25	17.55	21.60	24.30

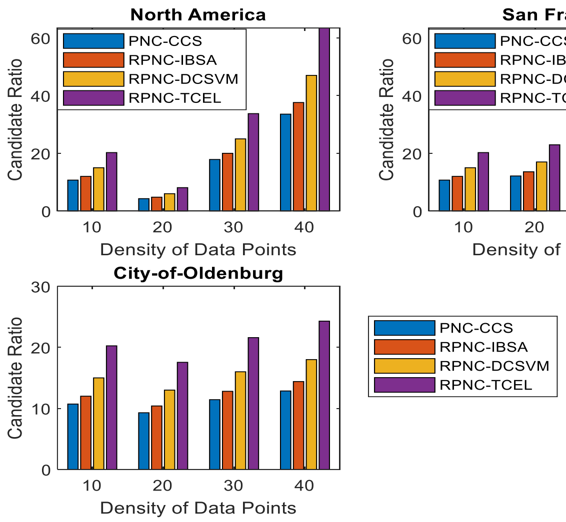


Figure 7: Candidate Ratio Analysis of Proposed Three Types of Clustering Algorithms for Three Types of Cities

Candidate ratio analysis of proposed and existing algorithms of clustering is shown by Figure 7. It shows that the proposed RPNC-TCEL algorithm increases the density of data points and decreases the candidate ratio when compared to other two algorithms of clustering.

Density of points / Methods	**Query Time**											
	North America				**San Francisco**				**Oldenburg**			
	PNC-CCS	**RPNC-IBSA**	**RPNC-DCSVM**	**RPNC-TCEL**	**PNC-CCS**	**RPNC-IBSA**	**RPNC-DCSVM**	**RPNC-TCEL**	**PNC-CCS**	**RPNC-IBSA**	**RPNC-DCSVM**	**RPNC-TCEL**
10	7.29	5.83	4.86	4.16	5.69	4.55	3.79	3.25	5.15	4.122	3.43	2.94
20	5.00	4.00	3.33	2.85	4.45	3.56	2.97	2.54	4.11	3.29	2.74	2.35
30	3.67	2.93	2.44	2.09	5.32	4.25	3.54	3.04	4.73	3.78	3.15	2.70
40	3.82	3.05	2.54	2.18	3.66	2.93	2.44	2.09	6.35	5.08	4.23	3.62

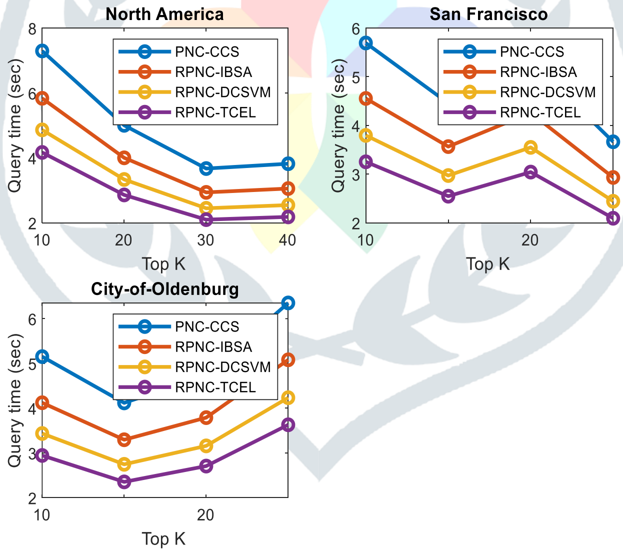


Figure 8: Query Time Comparison of Proposed Three Types of Clustering Algorithms for Three Types of Cities

Comparison of query time of proposed and existing algorithms of clustering is shown by Figure 8. It shows that query time required by the proposed RPNC-TCEL approach is low when comparing with other two clustering algorithms.

V. Conclusion

In this technical work, a new challenge involving the Reverse Path Nearby Cluster Query (R-PNC) is studied and examined along with the three diverse clustering approaches such as, Enhanced Density Clustering (EDC), Distribution Clustering and Trajectory Clustering for location recommendation process in spatial networks. This R-PNC query is developed for discovering the regions of prospective attention, and it is believed that it is quite in conditions including trip planning and location recommendation. In order to validate the presented technique, a sequence of experiments was carried out to show its use in location recommendation. In this research work, three kinds of road networks (San Francisco, North America and City of Oldenburg) were utilized for the analysis of the performance of clustering approaches. The results revealed that the proposed technique was capable of considering the various kinds of R-PNC query along with the Trajectory clustering techniques with much location forecasting efficiency compared to the other clustering approaches. At last, this comparative evaluation has shown that the provided R-PNC query with Trajectory clustering (RPNC-TCEL) yields a much better rate of location recommendation with any kind of road networks found in spatial networks.

REFERENCES

[1] Yao, X. 200. Research Issues in Spatio-Temporal Data Mining. Workshop on Geospatial Visualization and Knowledge Discovery. University Consortium for Geographic Information Science, Virginia, pp. 1-6.

[2] Mennis, J. and Guo, D. 2009. Spatial Data Mining and Geographic Knowledge Discovery—an Introduction. Computers, Environment and Urban Systems, 33(6): 403-408.

[3] Scellato, S. Noulas, A. and Mascolo, C. 2011. Exploiting Place Features in Link Prediction on Location-Based Social Networks. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1046-1054.

[4] Hocaoglu, C. and Sanderson, A.C. 1996. Planning Multi-Paths Using Speciation in Genetic Algorithms. Proceedings of IEEE International Conference on Evolutionary Computation, pp. 378-383.

[5] Harabor, D. and Botea, A. 2008. Hierarchical Path Planning for Multi-Size Agents in Heterogeneous Environments. IEEE Symposium on Computational Intelligence and Games, pp. 258-265.

[6] Ghosh, A. Parikh, J. Sengar, V. S. and Haritsa, J. R. 2002. Plan Selection based on Query Clustering. Proceedings of the International Conference on Very Large Databases, pp. 179-190.

[7] Haritsa, V. S. S. J. R. 2002. Plan Selection based on Query Clustering. Proceedings 2002 VLDB Conference: 28th International Conference on Very Large Databases (VLDB), pp.1-2.

[8] Hulgeri, A. and Sudarshan, S. 2002. Parametric Query Optimization for Linear and Piecewise Linear Cost Functions. Proceedings of the International Conference on Very Large Databases, pp. 167-178.

[9] Reddy, N. and Haritsa, J. R. 2005. Analyzing Plan Diagrams of Database Query Optimizers. In Proceedings of the 31st International Conference on Very Large Data Bases, pp. 1228-1239.

[10] Hulgeri, A. and Sudarshan, S. 2002. Parametric Query Optimization for Linear and Piecewise Linear Cost Functions. Proceedings of the International Conference on Very Large Databases, pp. 167-178.

[11] Shang, S. Zheng, K. Jensen, C. S. Yang, B. Kalnis, P. Li, G. and Wen, J. R. 2014. Discovery of Path nearby Clusters in Spatial Networks. IEEE Transactions on Knowledge and Data Engineering, 27(6):1505-1518.

[12] Lu, E. H. C. Chen, H. S. and Tseng, V. S. 2017. An Efficient Framework for Multi Request Route Planning in Urban Environments. IEEE Transactions on Intelligent Transportation Systems, 18(4): 869-879.

[13] Chen, L. Lv, M. and Chen, G. 2010. A System for Destination and Future Route Prediction based on Trajectory Mining. Pervasive and Mobile Computing, 6(6): 657-676.

[14] Kang, J. and Yong, H. S. 2010. A Frequent Pattern Based Prediction Model for Moving Objects. IJCSNS International Journal of Computer Science and Network Security, 10(3): 200-205.

[15] Choudhury, F. M. Culpepper, J. S. Sellis, T. and Cao, X. 2016. Maximizing Bichromatic Reverse Spatial and Textual k Nearest Neighbor Queries. Proceedings of the VLDB Endowment, 9(6): 456-467.

[16] Zhao, J. Gao, Y. Chen, G. Jensen, C. S. Chen, R. and Cai, D. 2017. Reverse Top-k Geo-Social Keyword Queries in Road Networks. IEEE International Conference on Data Engineering (ICDE), pp. 387-398.

[17] Shang, S. Yuan, B. Deng, K. Xie, K. Zheng, K. and Zhou, X. 2012. PNN Query Processing on Compressed Trajectories. GeoInformatica, 16(3): 467-496.

[18] Safar, M. 2008. Group k-Nearest Neighbors Queries in Spatial Network Databases. Journal of Geographical Systems, 10(4): 407-416.

[19] Wang, Y. Li, J. Zhong, Y. Zhu, S. Guo, D. and Shang, S. 2018. Discovery of Accessible Locations Using Region-Based Geo-Social Data. World Wide Web, pp. 1-19.

[20] Gotoh, Y. and Okubo, C. 2016. A Searching Method for Bichromatic Reverse k-Nearest Neighbor with Network Voronoi Diagram. ACM Proceedings of the International Conference on Advances in Mobile Computing and Multi Media, pp. 71-78.

[21] Tianyang, D. Lulu, Y. Qiang, C. Bin, C. and Jing, F. 2019. Direction-aware KNN queries for moving objects in a road network. World Wide Web, pp. 1-33.

[22] Rocha-Junior, J. B. and Nørvåg, K. 2012. Top-k Spatial Keyword Queries on Road Networks. ACM Proceedings of the International Conference on Extending Database Technology, pp. 168-179.

[23] De Felipe, I. Hristidis, V. and Rishe, N. 2008. Keyword Search on Spatial Databases. IEEE International Conference on Data Engineering, pp. 656-665.

[24] Tao, Y. and Sheng, C. 2013. Fast Nearest Neighbor Search With Keywords. IEEE Transactions on Knowledge and Data Engineering, 26(4): 878-888.

[25] Gao, Y. Qin, X. Zheng, B. and 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.

[26] Dona Rashmi G. and Narayani, V. 2019. Enhanced Density Clustering and Improved Bat Search Algorithm (IBSA) with Reverse Path Nearby Clusters (RPNC) in Spatial Networks. Jour of Adv Research in Dynamical & Control Systems, 11(04): 1754-1767.

[27] Dona Rashmi G. and Narayani, V. 2019.Hybrid Distribution Clustering With Support Vector Machine (Dcsvm) For Reverse Path Nearby Cluster (Rpnc) Query In Spatial Networks. Journal of Web Engineering, 18(4): 5096-5121.

[28] Dona Rashmi G. and NarayaniV. 2019. Reverse Path Nearby Cluster (Rpnc) Query Optimization using Trajectory Clustering With Ensemble Learning (Tcel) In Spatial Networks. International Journal of Engineering and Advanced Technology (IJEAT), 9(1): 7562-7572.

[29] https://www.cs.utah.edu/~lifeifei/SpatialDataset.htm.