Ranking Spatial Data by Quality Preferences

Yadala Sucharitha[1], P Chandra Shaker Reddy[2]
[1]*Assistant Professor of CSE Dept., CMR Institute of Technology, Hyderabad, TS, India*
[2]*Assistant Professor of CSE Dept., CMR College of Engineering & Technology, Hyderabad, TS, India*

***Abstract* - A query to a web search engine usually consists of a list of keywords, to which the search engine responds with the best or "top" k pages for the query. This top-k query model is prevalent over multimedia collections in general, but also over plain relational data for certain applications. A spatial preference query ranks objects based on the qualities of features in their spatial neighborhood. For example, using a real estate agency database of flats for lease, a customer may want to rank the flats with respect to the appropriateness of their location, defined after aggregating the qualities of other features (e.g., restaurants, cafes, hospital, market, etc.) within their spatial neighborhood. Such a neighborhood concept can be specified by the user via different functions. It can be an explicit circular region within a given distance from the flat. Another intuitive definition is to assign higher weights to the features based on their proximity to the flat. In this paper, we study how to process top- k queries efficiently in this setting, where the attributes for which users specify target values might be handled by external, autonomous sources with a variety of access interfaces. We present several algorithms for processing such queries and evaluate them thoroughly using both synthetic and real web-accessible data. Extensive evaluation of our methods on both real and synthetic data reveals that an optimized branch-and-bound solution is efficient and robust with respect to different parameters.**

***Index Terms* - Query processing, spatial databases, distance browsing, ranking, nearest neighbors.**

I.INTRODUCTION

Spatial database systems manage large collections of geographic entities, which apart from spatial attributes contain non spatial information (e.g., name, size, type, price etc.). In this paper, we study an interesting type of preference queries, which select the best spatial location with respect to the quality of facilities in its spatial neighborhood. Software Solutions is an IT solution provider for a dynamic environment where business and technology strategies converge [1]. Their approach focuses on new ways of business combining IT innovation and adoption while also leveraging an organization's current IT assets. Their work with large global corporations and new products or services and to implement prudent business and technology strategies in today's environment. Spatial database systems manage large collections of geographic entities, which apart from spatial attributes contain non spatial information (e.g., name, size, type, price etc.). In this paper, we study an interesting type of preference queries, which select the best spatial location with respect to the quality of facilities in its spatial neighborhood [2].

Given a set D of interesting objects (e.g., candidate locations), a top-k spatial preference query retrieves the k objects in D with the highest scores. The score of an object is defined by the quality of features (e.g., facilities or services) in its spatial neighborhood. As a motivating example, consider a real estate agency office that holds a database with available flats for lease. Here "feature" refers to a class of objects in a spatial map such as specific facilities or services [3]. A customer may want to rank the contents of this database with respect to the quality of their locations, quantified by aggregating non-spatial characteristics of other features (e.g., restaurants, cafes, hospital, market, etc.) in the spatial neighborhood of the flat (defined by a spatial range around it). Quality may be subjective and query parametric. For example, a user may define quality with respect to non-spatial attributes of restaurants around it (e.g., whether they serve seafood, price range, etc.) [4].

1.1 Background

The artifacts arising from many imaging devices are quite different from the images that they contaminate, and this difference allows humans to "see past" the artifacts to the underlying image. The goal of image restoration is to relieve human observers from this task (and perhaps even to improve upon their abilities) by reconstructing a plausible estimate of the original image from the distorted or noisy observation. A prior probability model for both the noise and for uncorrupted images is of central importance for this application [5]. Modeling the statistics of natural images is a challenging task, partly because of the high dimensionality of the signal. Two basic assumptions are commonly made in order to reduce dimensionality. The first is that the probability structure may be defined locally. Typically, one makes a Markov assumption, that the probability density of a pixel, when conditioned on a set of neighbors, is independent of the pixels beyond the neighborhood. The second is an assumption of spatial homogeneity: the distribution of values in a neighborhood is the same for all such neighborhoods, regardless of absolute spatial position. Although the most common model arising from these two assumptions is a Gaussian Markov random field, the restriction to second-order processes is not required, and is problematic for image modeling, where the complexity of local structures is not well described by Gaussian densities [6]. A useful framework for capturing higher order statistics comes from augmenting a simple parametric model for local dependencies (e.g., Gaussian) with a set of "hidden" random variables that govern the parameters (e.g., variance). Such hidden Markov models have become widely used, for example, in speech processing. Adaptive Kernel-Based Image De-noising Employing Semi-Parametric Regularization is a simple tool for Adaptive Kernel-Based Image De-noising Employing Semi-Parametric Regularization, which includes different filters and tools to analyze images available in the framework. It's easy to develop your own filters and to integrate them with the code or use the tools in our own application. In Figs. 1a, 1b and 1c, feature points and existing sites are shown as black and gray points, respectively [7].

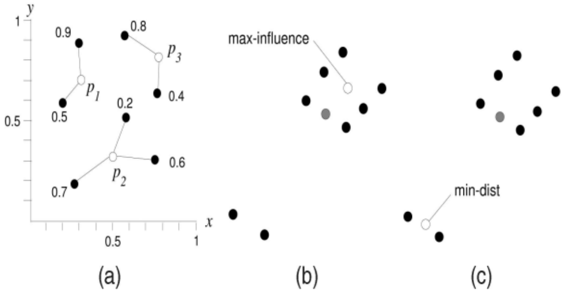

Fig. 1. Influential sites and optimal location queries. (a) Top-k influential. (b) Max-influence. (c) Min-distance

1.2 Problem statement

In the existing works there is no solution for processing the top-k spatial preference query and there are no alternative techniques for minimizing the I/O accesses to the object and feature data sets. In the existing studies there are two basic ways for ranking objects, 1) Spatial ranking, which orders the objects according to their distance from a reference point. 2) Non spatial ranking, which orders the objects by an aggregate function on their non-spatial values [8].

The main objectives of the proposed research are,

- In this paper, we studied top-k spatial preference queries, which provide a novel type of ranking for spatial objects based on qualities of features in their neighborhood.
- The neighborhood of an object p is captured by the scoring function. 1) The range score restricts the neighborhood to a crisp region centered at p, whereas 2) the influence score relaxes the neighborhood to the whole space and assigns higher weights to locations closer to p.
- We presented five algorithms for processing top-k spatial preference queries.
- The baseline algorithm SP computes the scores of every object by querying on feature data sets. The algorithm GP is a variant of SP that reduces I/O cost by computing scores of objects in the same leaf node concurrently.
- The algorithm BB derives upper bound scores for non-leaf entries in the object tree, and prunes those that cannot lead to better results.
- The algorithm BB* is a variant of BB that utilizes an optimized method for computing the scores of objects (and upper bound scores of non-leaf entries).
- The algorithm FJ performs a multi way join on feature trees to obtain qualified combinations of feature points and then search for their relevant objects in the object tree.
- The challenge is to develop alternative methods for computing the upper bound scores for a group of points on a road network.

The rest of this paper is structured as follows: Section 2 discussed about related works and in Section 3 presented the proposed methodology. In Section 4, our query algorithms are experimentally evaluated with real and synthetic data. Finally, Section 5 concludes the paper with future research directions.

II. RELATED WOKS

The problem of noise removal from a digitized image is one of the most important ones in digital image processing [9]. So far, various techniques have been proposed to deal with it. Among the most popular methodologies are, for example, the wavelet-based image de-noising methods (which dominate the research in recent years, see for example), the image de-noising methods based on Partial Differential Equations, neighborhood filters, some methods or impulse detection see, methods based on fractal theory and, more recently, methods of nonlinear modeling using kernel regression and/or local expansion approximation techniques [10]. In many cases, the de-noising techniques are focused on a particular noise model (gaussian, impulse, etc.). Thus, they cannot treat effectively more complex models, which are often met in practical applications. In this paper, we propose a different approach. Our only assumption is that the image is corrupted by zero mean additive noise, without any additional information with respect to the noise pdf [11]. To remove the noise, we employ the well-known (especially in pattern analysis) theory of kernels. In kernel methodology, the notion of the Reproducing Kernel Hilbert Space (RKHS) plays a crucial role. A RKHS, introduced in, is a rich construct (roughly, a smooth space with an inner product), which has been proven to be a very powerful tool. Kernel based methods are utilized in an increasingly large number of scientific areas, especially where nonlinear models are required. For example, in pattern analysis, a classification task of a set is usually reformed by mapping the data into a higher dimensional space (possibly of infinite dimension), which is a RKHS [12]. The advantage of such a mapping is to make the task more tractable, by employing a linear classifier in the feature space, exploiting Cover’s theorem. This is equivalent with solving a nonlinear problem in the original space. Similar approaches have been used in principal components analysis, in Fisher’s linear discriminate analysis, in clustering, regression and in many other sub-disciplines. Recently, processing in RKHS is gaining in popularity within the Signal Processing community in the context of adaptive filtering and beam forming [13].

III. PROPOSED METHODOLOGY

We assume that the object data set D is indexed by an R-tree and each feature data set Fc is indexed by an MAX a R-tree, where each non-leaf entry augments the maximum quality (of features) in its sub-tree. Nevertheless, our solutions are directly applicable to data sets that are indexed by other hierarchical spatial indexes (e.g., point quad-trees) [14].

The rationale of indexing different feature data sets by separate R-trees is that:

1. A user queries for only few features (e.g., restaurants and cafes) out of all possible features (e.g., restaurants, cafes, hospital, market, etc.), and
2. Different users may consider different subsets of features.

3.1 Probing Algorithms

We first introduce a brute-force solution that computes the score of every point p 2 D in order to obtain the query results. Then, we propose a group evaluation technique that computes the scores of multiple points concurrently [15].

3.2 Optimized Branch-and-Bound Algorithm

GP is still expensive as it examines all objects in D and computes their component scores. We now propose an algorithm that can significantly reduce the number of objects to be examined. The key idea is to compute, for non-leaf entries e in the object tree D, an upper bound T a of the score for any point p in the sub-tree of e. If T a, then we need not access the sub-tree of e, thus we can save numerous score computations [16]. In Fig. 2, v1 and v2 are non-leaf entries in the object tree D respectively.

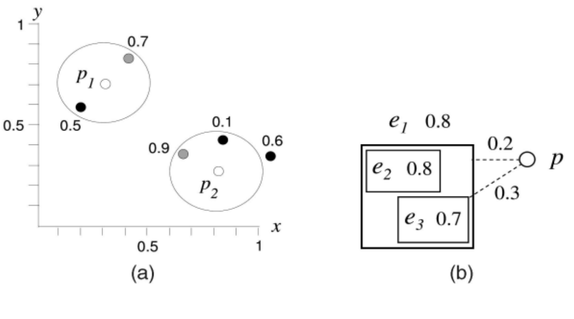

Fig. 2. Examples of deriving scores. (a) Upper bound scores. (b) Optimized computation.

As we can see the system architecture depicts the heart of the system, the architecture contains administrator, user, system, server and database. Our system interacts with two subsystem or block i.e admin and user. Admin will initially handle the database he will be the first to interact with database. The functions which admin will perform are Admin Login i.e he will login with authorization, the next function performed by admin is to Add Flats Details. Now after Admin role comes the Users interaction with the system. The user will register himself. The user needs to properly login to view flats details. As the user would like to view the flats accordingly to his requirement the flats will be searched according to spatial and non-spatial information stored in database [17, 18]. As per our system he can request flat with various parameters that can be with respect to café, restaurant, market, hospital. This spatial information is ranked objects based on qualities of features in spatial neighborhood. These features were spatial, but in our system, we have implemented interesting type of preference queries which apart from spatial attributes also contain non-spatial information (eg name, size, type, price etc).so the user will be able to view flats with high quality features also after viewing the flat he can vote for the flat on basis of quality. When user request to view the flat the system (browser) will retrieve information from database and give him and after viewing flat he again interacts with the system by voting and vote will make changes in database by server dynamically. Now this is all about user, admin and how system works [19, 20]. Now about database, database here in our system is the spatial database which manage large collection of geographical entities which apart from spatial attributes contain non-spatial information where spatial attributes are café, restaurant, hospital, market, and non-spatial attributes are name, size, type, price and so on calculated the potential."

3.1 Optimized Branch and Bound Algorithm

```
Algorithm: Enhanced Branch and Bound
Wk: = new min-heap of size k (initially empty);
?: =0;
// k-th score in Wk
1: Call search algorithm
// Take input as search result E from search algorithm
2: V: {E| E e N}; //V denotes set in which points are to be stored
3: If N is non-leaf then
4: for c: =1 to m do
5: compute T (E) for all E e V concurrently.
6: remove entries E in V such that T+ (E) <=?;
7: for each entry E e v such that T (E) >? do
8: read the child node N pointed by E;
9: continue step 2;
10: else
11: for c: =1 to m do Ranking Spatial Data by Quality Preferences 72
12: compute T (E) for all E e V concurrently;
13: remove entries e in V such that T+ (E) <=V;
14: Sort entries E e V in descending order of T (E);
15: Update Wk (and?) by entries in v;
```

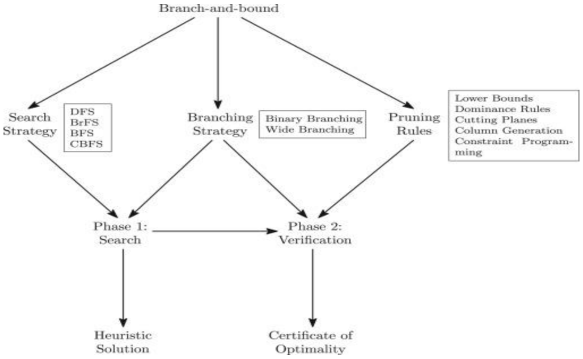

Fig. 3. A diagram of the three main B&B components.

In branch and bound algorithm, changes have been made in getting input values and also about sorting the entries, resulted with enhanced branch and bound algorithm. The input values of enhanced BB are the output of searching algorithm [21, 22]. Instead of performing sorting individually on each node among its child nodes, entire tree node has been sorted after this process is over. This will reduce the time effectively and improve the performance. Conceptual diagram of the BB is displayed in Fig. 3 and execution process is shown in Fig. 4.

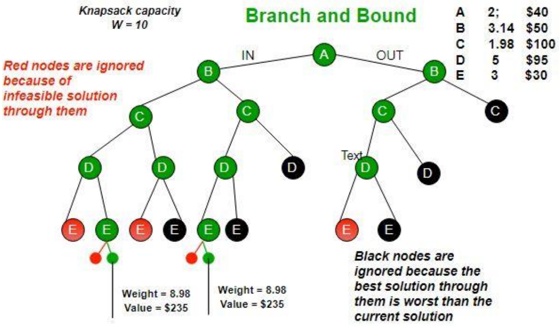
Knapsack capacity
W = 10
Branch and Bound
A 2; $40
B 3.14 $50
C 1.98 $100
D 5 $95
E 3 $30
IN
OUT
A
B
B
C
C
C
C
D
D
D
D
D
D
E
E
E
E
E
E
E
E
Text
Red nodes are ignored because of infeasible solution through them
Weight = 8.98
Value = $235
Weight = 8.98
Value = $235
Black nodes are ignored because the best solution through them is worst than the current solution

Fig. 4. Implementation of the proposed method (BB)

Spatial preference query ranks objects based on quality of features in their spatial neighborhood. So, the ranking is done by assigning higher weights to features based on proximity of flat. So, we have formally defined spatial preference queries and proposed appropriate searching algorithms. Here we have evaluated of method on both real and synthetic data using branch and bound solution [23, 24]. In branch and bound what will happen, the objects will be examined but by significantly reducing number of observation whereas earlier in GP (group probing) the procedure was expensive as it examined all objects in D and computed their component scores. As per branch and bound the key idea is to compute for non-leaf entries e in object tree D, an upper bound T(e) of score r(p) for any point p in sub-tree of e, thus we can save numerous score computations .BB is called with N being the root of D. If N is non-leaf the scores will be computed. T(e) for non-leaf entries e concurrently. The equation will be evaluated for component scores Tc(e) known so far, we can derive T+(e) an upper bound of T(e) and if T+(e)<=γ then sub-tree of e cannot contain better result than and it is removed from V. In order to obtain points with high scores early, we sort entries in descending order of T(e) before invoking the above procedure recursively on child entries in V. The branch and bound algorithm were responsible for reducing number of computations. After that, the feature join algorithm is used for evaluating top-k spatial preference query by a multi-way spatial join on various features whose quality is to be judged. F1, F2…. Fm are those features, also the feature join is used to obtain combination of feature points which can be in neighborhood of some object from D. Hence the object with top-k highest score is retrieved by feature join.

As per the flow of our system first comes the homepage of Ranking Spatial System. After visiting homepage, the person can choose to login, so the person who wants to login can be admin or user. Now we will see these two main components of system one by one. First the admin, if the person is admin, he will enter into admin login page and he will enter id and password. If authorized person then he can add flat details and if unauthorized person the flow will end so the admin can add flat details, save the details i.e. store in database, manipulations in database i.e. maintain flat details. All this interaction with database is of admin and its work is done. this side can also be said as admin rating. Now on user side if the user is not authorized the flow will stop otherwise if the user is authorized, he can search flats then for him the flats will be retrieved from database and viewed to him. After his use of flat details, he can also rate for it and logout. This is how the flow of our system goes [25, 26].

As per our Implementation, spatial preference query ranks the objects based on their spatial neighborhood including the qualities of features. We used appropriate indexing techniques and search algorithms for calculating the accurate output. As per Figure4 we can see that proper calculating of attributes are fetched from our system, as per Figure4 user can access single attributes considering all Spatial data such as Location, area(sq.), distance and price And Figure5 describes with multiple attributes selection for the user benefit. Combination of both Spatial and non-spatial information result to accurate and perfect output and gives us the ranked data as per the requirement of the user. Figure6 shows the admin rating provided as per the survey of the spatial neighborhood.

IV. RESULTS ANALYSIS

Performance is evaluated on basis of factor like ease of development, availability of hardware and re-usable code availability. The feasibility of running software is tested to be of minimum risk, these were selected as a platform for development. Performance is to estimate whether it is possible to develop the proposed system with the available hardware, software, and network resources. Since all proposed hardware, software and network requirement are easily available; the development of application became feasible.

In this section, we conduct experiments on real object and feature data sets in order to demonstrate the application of top-k spatial preference queries. We obtained three real spatial data sets from a travel portal website, http://www.allstays.com/. Locations in these data sets correspond to (longitude and latitude) co-ordinates in US. We cleaned the data sets by discarding records without longitude and latitude. Each remaining location is normalized to a point in the 2D space ½0; 10; 000_2. One data set is used as the object data set and the other two are used as feature data sets. The object data set D contains 11,399 camping locations. The feature data set F1 contains 30,921 hotel records, each with a room price (quality) and a location. The feature data set F2 has 3,848 records of Wal-Mart stores, each with a gas online availability (quality) and a location. The domain of each quality attribute (e.g., room price and gasoline availability) is normalized to the unit interval ½0; 1_. Intuitively, a camping location is considered as good if it is close to a Wal-Mart store with high gasoline availability (i.e., convenient supply) and a hotel with high room price (which indirectly reflects the quality of nearby outdoor environment).

Fig. 5 plots the cost of the algorithms with respect to _, for queries with range scores. At a very small _ value, most of the objects have the zero score as they have no feature points within their neighborhood. This forces BB, BB*, and FJ to access a larger number of objects (or feature combinations) before finding an object with nonzero score, which can then be used for pruning other unqualified objects. Fig. 6 compares the cost of the algorithms with respect to _, for queries with influence scores. In general, the cost follows the trend in Fig. 16. BB* outperforms BB at low_ value whereas BB incurs a slightly lower cost than BB* at a high _ value. Observe that the cost of BB and BB* is close to that of FJ when _ is sufficiently high. In summary, the relative performance between the algorithms in all experiments is consistent to the results on synthetic data.

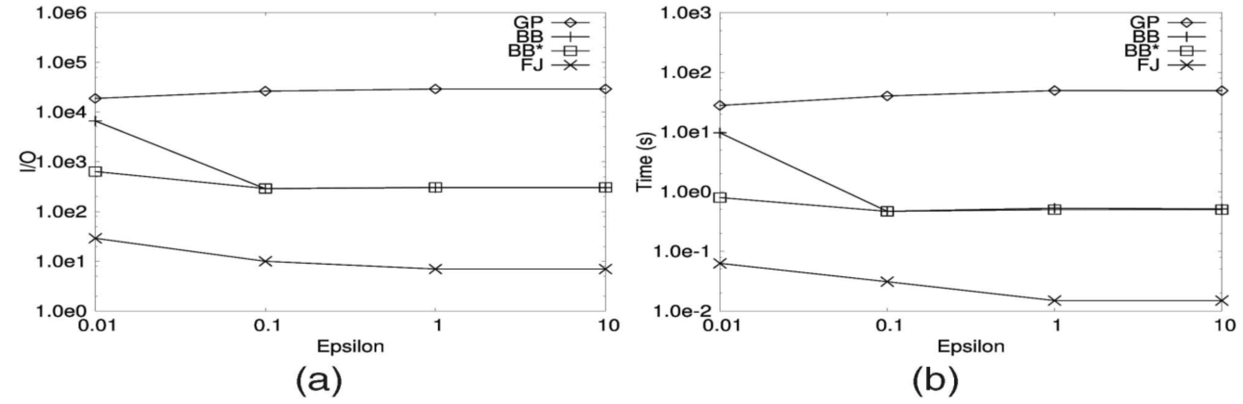

Fig. 5. Effect of _, range scores, real data. (a) I/O. (b) Time.

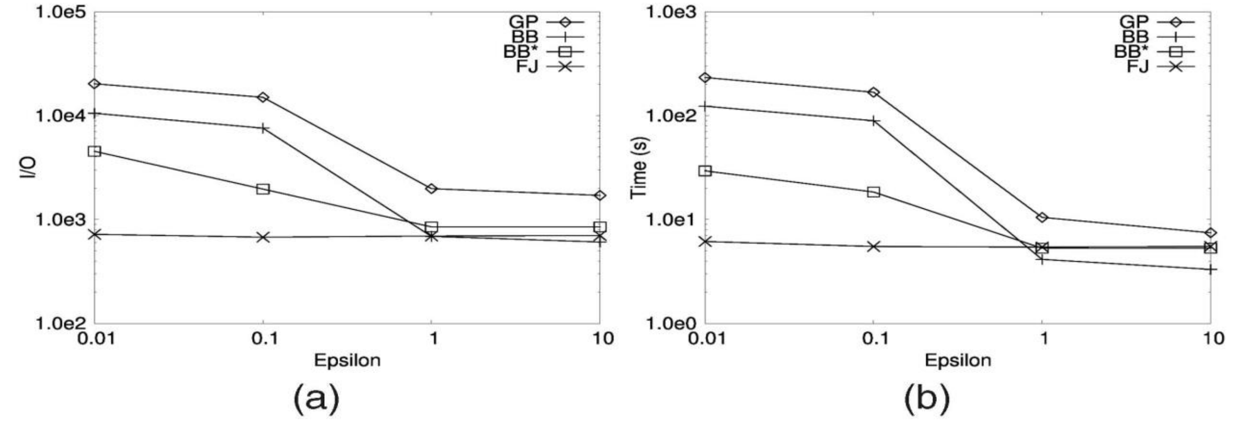

Fig. 6. Effect of _, influence scores, real data. (a) I/O. (b) Time.

V. CONCLUSION

In this paper, we implemented a top-k spatial preference query, which provides a new type of ranking for spatial objects based on qualities of features in their neighborhood. The neighborhood of an object p is captured by the scoring function: (i) the range score restricts the neighborhood to a crisp region centered at p, whereas (ii) the influence score relaxes the neighborhood to the whole space and assigns higher weights to locations closer to p. The algorithm BB derives upper bound scores for non-leaf entries in the object tree, and prunes those that cannot lead to better results. The algorithm BB utilizes an optimized method for computing the scores of objects (and upper bound scores of non-leaf entries). The algorithm FJ performs a multi-way join on feature trees to obtain qualified combinations of feature points and then search for their relevant objects in the object tree. BB is scalable to large datasets and it is the strongest algorithm with respect to various parameters. However, FJ is the best algorithm in cases where the number m of feature datasets is low, and each feature dataset is small. In the future, we will study the top-k spatial preference query on road network, in which the distance between two points is defined by their shortest path. The challenge is to develop alternative methods for computing the upper bound scores for a group of points on a road network.

REFERENCES

[1] M.L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k Spatial Preference Queries," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2007.

[2] N. Bruno, L. Gravano, and A. Marian, "Evaluating Top-k Queries over Web-Accessible Databases," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2002.

[3] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD, 1984.

[4] Sucharitha Y, Vijayalata Y, Prasad VK. Analysis of Early Detection of Emerging Patterns from Social Media Networks: A Data Mining Techniques Perspective. InSoft Computing and Signal Processing 2019 (pp. 15-25). Springer, Singapore.

[5] Reddy PC, Babu AS. Survey on weather prediction using big data analystics. In2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) 2017 Feb 22 (pp. 1-6). IEEE.

[6] G.R. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases," ACM Trans. Database Systems, vol. 24, no. 2, pp. 265- 318, 1999.

[7] R. Weber, H.-J. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High- Dimensional Spaces," Proc. Int'l Conf. Very Large Data Bases (VLDB), 1998.

[8] Reddy PC, Sureshbabu A. An applied time series forecasting model for yield prediction of agricultural crop. InInternational Conference on Soft Computing and Signal Processing 2019 Jun 21 (pp. 177-187). Springer, Singapore.

[9] K.S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'Nearest Neighbor' Meaningful?" Proc. Seventh Int'l Conf. Database Theory (ICDT), 1999.

[10] R. Fagin, A. Lotem, and M. Naor, "Optimal Aggregation Algorithms for Middleware," Proc. Int'l Symp. Principles of Database Systems (PODS), 2001.

[11] Reddy PC, Babu AS. A novel approach to analysis district level long scale seasonal forecasting of monsoon rainfall in Andhra Pradesh and Telangana. International Journal of Advanced Research in Computer Science. 2017 Nov 1;8(9).

[12] I.F. Ilyas, W.G. Aref, and A. Elmagarmid, "Supporting Top-k Join Queries in Relational Databases," Proc. 29th Int'l Conf. Very Large Data Bases (VLDB), 2003.

[13] N. Mamoulis, M.L. Yiu, K.H. Cheng, and D.W. Cheung, "Efficient Top-k Aggregation of Ranked Inputs," ACM Trans. Database Systems, vol. 32, no. 3, p. 19, 2007.

[14] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP Operations in Spatial Data Warehouses," Proc. Int'l Symp. Spatial and Temporal Databases (SSTD), 2001.

[15] S. Hong, B. Moon, and S. Lee, "Efficient Execution of Range Top-k Queries in Aggregate R-Trees," IEICE Trans. Information and Systems, vol. 88-D, no. 11, pp. 2544-2554, 2005.

[16] Reddy PC, Sureshbabu A. An Adaptive Model for Forecasting Seasonal Rainfall Using Predictive Analytics. InInternational Journal of Intelligent Engineering and Systems 2019 (pp. 22-32).

[17] T. Xia, D. Zhang, E. Kanoulas, and Y. Du, "On Computing Top-t Most Influential Spatial Sites," Proc. 31st Int'l Conf. Very Large Data Bases (VLDB), 2005.

[18] Y. Du, D. Zhang, and T. Xia, "The Optimal-Location Query," Proc. Int'l Symp. Spatial and Temporal Databases (SSTD), 2005.

[19] Sucharitha Y, Prasad VK, Vijayalatha Y. Emergent Events Identification in Micro-Blogging Networks Using Location Sensitivity.

[20] D. Zhang, Y. Du, T. Xia, and Y. Tao, "Progessive Computation of The Min-Dist Optimal-Location Query," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB), 2006.

[21] Y. Chen and J.M. Patel, "Efficient Evaluation of All-Nearest- Neighbor Queries," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2007.

[22] P.G.Y. Kumar and R. Janardan, "Efficient Algorithms for Reverse Proximity Query Problems," Proc. 16th ACM Int'l Conf. Advances in Geographic Information Systems (GIS), 2008.

[23] M.L. Yiu, P. Karras, and N. Mamoulis, "Ring-Constrained Join: Deriving Fair Middleman Locations from Pointsets via a Geometric Constraint," Proc. 11th Int'l Conf. Extending Database Technology (EDBT), 2008.

[24] M.L. Yiu, N. Mamoulis, and P. Karras, "Common Influence Join: A Natural Join Operation for Spatial Pointsets," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2008.

[25] Y.-Y. Chen, T. Suel, and A. Markowetz, "Efficient Query Processing in Geographic Web Search Engines," Proc. ACM SIGMOD, 2006.

[26] V.S. Sengar, T. Joshi, J. Joy, S. Prakash, and K. Toyama, "Robust Location Search from Text Queries," Proc. 15th Ann. ACM Int'l Symp. Advances in Geographic Information Systems (GIS), 2007.