www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 10 Oct. 2016, Page No. 18620-18634

Fusion Layer Topological Space Query Indexing For Uncertain Data Mining

M. Kalavathi, Dr. P. Suresh
Head, Department of Computer Science Government Arts and Science College,
Komarapalayam.- 638 183.Tamilnadu. INDIA
Email : Kalavathisakthi@gmail.com
Head, Department of Computer Science Salem Sowdeswari College,
Salem - 636 010.Tamilnadu. INDIA
Email :sur_bh0071@rediffmail.com

***Abstract* - Data uncertainty is an intrinsic property in different applications such as sensor network monitoring, object recognition, location-based services (LBS), and moving object tracking. The data mining methods are applied to the above mentionedapplications their uncertainty has to be handled to achieve the accurate query results. The several probabilistic algorithm estimates the location and control for each object but not effective in handling query processing in distributed environments. Probabilistic inverted indexing computes the lower bound and upper bound for a threshold keyword but fails to extend the technique on tackle the correlation. In order to overcome the issues in uncertain data mining, Fusion Layer Topological Space Query Indexing technique (FLTS)is introduced. Initially, the queries are articulated on any random subset of attributes in the uncertain data.The FLTS index technique answers the top-k queries competently. FLTS index correctlyshows in their dominant relationships andsignificantly reduces the number of tuple accessed through query processing by pruning redundant tuple based on two criterions such as layer point sort method and the record point sort method. Initially, layer point sort method is used in FLTS indexto sort out tuple totally on the combination of all attribute values of the tuple. Subsequently, each attributes values particularly used for rating the tuple using record point sort method. Therefore,the correlation is removed significantly. Through an analysis of the interaction of the two sorting methods, derive a fixed bound that reduces the number of tuple retrieved during query processing and obtaining the correct query results in distributed environments. An experimental result shows that the FLTS indexing technique improves the query retrieving efficiency, response time, memory consumption and scalability.**

Keywords - Fusion Layer Topological Space, Query Indexing, layer point sort method, record point sort method, uncertain data mining.

INTRODUCTION

Data mining is the method for detection and examination of large data sets to determine significant pattern and rules. In data mining, the data is mined by allowing different processes. Data preprocessing is the initial step of data retrieval to guarantee the exactness of successive process and remove the data uncertainty. Data Retrieval System offers an efficient and complicated device for preserving and organizing the records.Uncertain data managingis interest in recent years due to a several fields uses this type of data. The performances of data mining applications are concerned with the fundamental uncertainty in data. It is important to plan data mining methods that consider the uncertainty in the computation process. In essential applications like environmental surveillance by large-scale networks, uncertainty is intrinsic in data because of factors like incompleteness of data, limits of equipment and delay or loss in data transfer. The several data mining techniques were developed to show theirperformance and limitations with the help of literature.

A new U-Quadtree indexing was introduced in [1], for analyzing the uncertain objects within the multi-dimensional space. Based on, the queries can be processed effectively but it failed to handle the query processing in distributed environments. Probabilistic inverted (PI) index utilizes the probability density

function was effectively derived in [2]. PI index quickly compute the lower bound and upper bound for a threshold keyword query, by which lots of incompetent nodes are pruned and experienced nodes are returned as early as possible. PI index determines all the quasi-SLCA results meeting the threshold requirement but fails to extend the technique on tackling the correlation.

Constrained space Probabilistic range query (CSPRQ) technique was designed in [3] towards uncertain moving objects. However, the location of the uncertain data is unaddressed. The top-k similarity ofthe multi-valued objects isconsidered in [4]. Based on, the difference between the two multi-valued objects is computed using quantile based distance for obtaining the qualified instance distribution. However, it's difficult to identify different query results.

Present data inference and compression substrate over RFID streams was presented in [5] using a time-varying graph model for attaining the feasible object locations and object relationships. However, the algorithm not effective in handling query processing in distributed environments.

Uncertain Canonical Correlation Analysis (CCA) [6] for multi-view feature extraction from uncertain data streams. CCA is a method to remove frequent features from multivariate data. However, it failed to apply wide application fields.A new framework was designed in [7] for efficient uncertain RFID data processing and maintaining a number of queries to trackand trace RFID objects. However, the query processing efficiency does not improved at a required level. A novel QPPUBG method was introduced in [8] for efficient query processing in uncertain big data.

A skyline ranking method was introduced in [9] for uncertain data with confidence. The skyline work for uncertain data considers each tuple which is specified with all probabilities of confidence.A new arrangement called MRST is formed in [10] to collect the probability density function of uncertain object. However, it's difficult to support high-dimensional un-certain data.

The contribution of the research work is organized as follows. An efficient Fusion Layer Topological Space Query Indexing (FLTS) technique is developed for improving the query processing efficiency and effectively mining the uncertain data. The FLTS index answers the top k query and it also reduces the number of tuple accessed through query processing by pruning redundant tuple based on two criterions such as layer point sort method and the record point sort-out method. Based on the above said methods, the correlation is removed significantly and it improved the dominant relationship between the attributes. The two sorting methods improve thequery retrieving efficiency. This helps to improve the correct query results in distributed environments.

The organization of the research work is described as follows: Section 2 discusses a related works of different techniques to improve query processing efficiency. Section 3 describes the Fusion Layer Topological Space (FLTS) Query Indexing technique with neat diagram. Section 4 presents experimental evaluation and the performance results are described in section 5, and concluding remark is presented in Section 6.

RELATED WORKS

The ability for retrieving the features from uncertain data is highly required in the recent data mining applications. Different techniques were used for mining the frequent pattern from uncertain data and indexing structure, but it required more time for extracting specific feature.A heuristic approach was designed in [11] for clustering on uncertain objects.The data storage scheme was introduced in [12] for maintaining the multidimensional query inspired with K-D tree. An effective framework was developed in [13] for reducing the limitations of optimizing probabilistic coverage.

In [14], PH-tree based authenticated data structure (ADS) was introduced for applicability on uncertain data and probabilistic database. However, the correlation between the data is remained

unaddressed.An efficient and effective query processing optimization model was designed in [15] for query processing applications which provides optimal performance.There are two types of approaches approximate range search and approximate continuous range search in spatial databases was designed in [16] to improve the query processing efficiency.

An efficient stream mining algorithms was designed in [17] to mine frequent patterns effectively and efficiently from such streaming data.In [18], a tree-based mining algorithm was developed that used to mine frequent patterns from dynamic uncertain data stream based on time-fading and landmark models.An efficient Skyline query algorithm was designed in [19] of uncertain moving stream data but the performance is failed to use index structures such as grid, R tree and so on. A novel algorithm was effectively designed in [20] to mine the uncertain group patterns that improve the mining efficiency but it not capable to handle more complex patterns.

The above mentioned challenges related with different data mining techniques difficult to tackle the data uncertainty problem effectively. Hence, a novel approach called Fusion Layer Topological Space Query (FLTS) Indexing technique is introduced to solve the uncertain data mining problem on processing the user requested query. The brief explanation about the FLTS Indexing technique is presented in next section.

FUSION LAYER TOPOLOGICAL SPACE QUERY INDEXING TECHNIQUE

In the development ofdifferent database applications, data uncertainty is animportant area due to the extensive survival of uncertainty. Hence, the plan of query processing methods is significant research area in data mining. In general, the uncertainty causes a various difficulties in the data base applications comprises data randomness and incompleteness, control of measuring equipment, delay or loss of data updates and privacy preservation. Hence the new approach called as Fusion Layer Topological Space (FLTS)index technique is introduced.

A. Problem description

In this section, problem of top-k queries are structured when the tuples are rated by a random subset of attributes. A target relation R of number of tuples N and has 'n' attributes$Attr_1, Attr_2, Attr_3 \ldots Attr_n$. Each attributes consists of the value is ranges between [0.0, 1.0], henceevery tuple in R is assumed as a point in the d-dimensional space [0.0, 1.0] d. After that, the space [0.0, 1.0] d is identified, refer to a tuple T in R as an object T in the space, and uses the tuple and the object interchangeably as is appropriate.

A scoring function$f(T)$maps an each data tuplesto a real value in [-1.0, 1.0]. Followed by, a top-k query is to determine the k objects in R that contains the minimum or maximum score under f (T). From that, let us assumed and are looking for the minimum scored objects. Therefore, the aim is to retrieve a sequence of tuples [T_1, $T_2 \ldots T_k$] that satisfies the condition as, f (T_1)≤ f (T_2) ≤…..≤f(T_k) ≤ F (T_l), k+1≤ l ≤N. Here, T_j represents thejth rated tuplesand it arranged in the ascending order of their score value, where the j varies from 1 to N (1≤j≤N). The scoring function for top-k queries is normallyconsidered to be either linear [4,6,8,9,30] or monotone [5,3,6,8,9,21,29,30,33].A linear scoring function is a function which is formularized as follows,

$$f_w(T) = \sum_{i=1}^{d} W_t[i]T\,[i] \quad (1)$$

From (1), T[i] is denoted as the ith attribute value of T and $W_t[i]$ is the "weight" of the ith attribute. The vector of $W_t[i]$is a preference vector with thevalue ranges between [−1.0, 1.0] then the normalized function is denotes as,

$$\sum_{i=1}^{d} |W_t[i]| = 1 \quad (2)$$

DOI: 10.18535/ijecs/v5i10.48

From (2), a linear function $f_w(T)$ is monotone if and only if its $W_t[i]$ values are all non-negative. Based on the sign of the $W_t[i]$values, a linear function is a chance of non-monotone. However, the Fusion Layer Topological Space indexis designed to handlelinear scoring functions which are monotone or non-monotone.Based on above said description, FLTP index is effectively constructed and handles the subspace top-k queries.

B. Fusion Layer Topological Space Index

In Fusion layer topological space index, query processing is an important problem in the data mining process. Indexing is an effective technique to the query processing especially to a set of Fusion layer topological space. This scheme answer the user requested queries and produces the better search result on the uncertain data. In this section, an effective fusion layer topological space query indexing techniques is developed to mine the uncertain data. Initially, the user requested queries are articulated on several random subsets of attributes in the uncertain data. The FLTS index technique is used to answer the top k queries for avoiding the uncertainty. In addition, FLTS index technique uses integration of two types of sort method provides better results for query processing.

The main objective of FLTS index technique is to enable both layer point sort method and record point sort out method.The layer point sort method prunes anumber of redundanttuples through the overall combination of its entire attribute values similar to layer based approach. The record point sort out method prunes every attribute used for ranking the tuple with nonzero weights.In addition, the number of tuples retrieved is reduced during query processing. Therefore, the correct query results is obtained in distributed environments. The structural diagram of the proposed FLTS-QI technique is illustrated in figure 1.

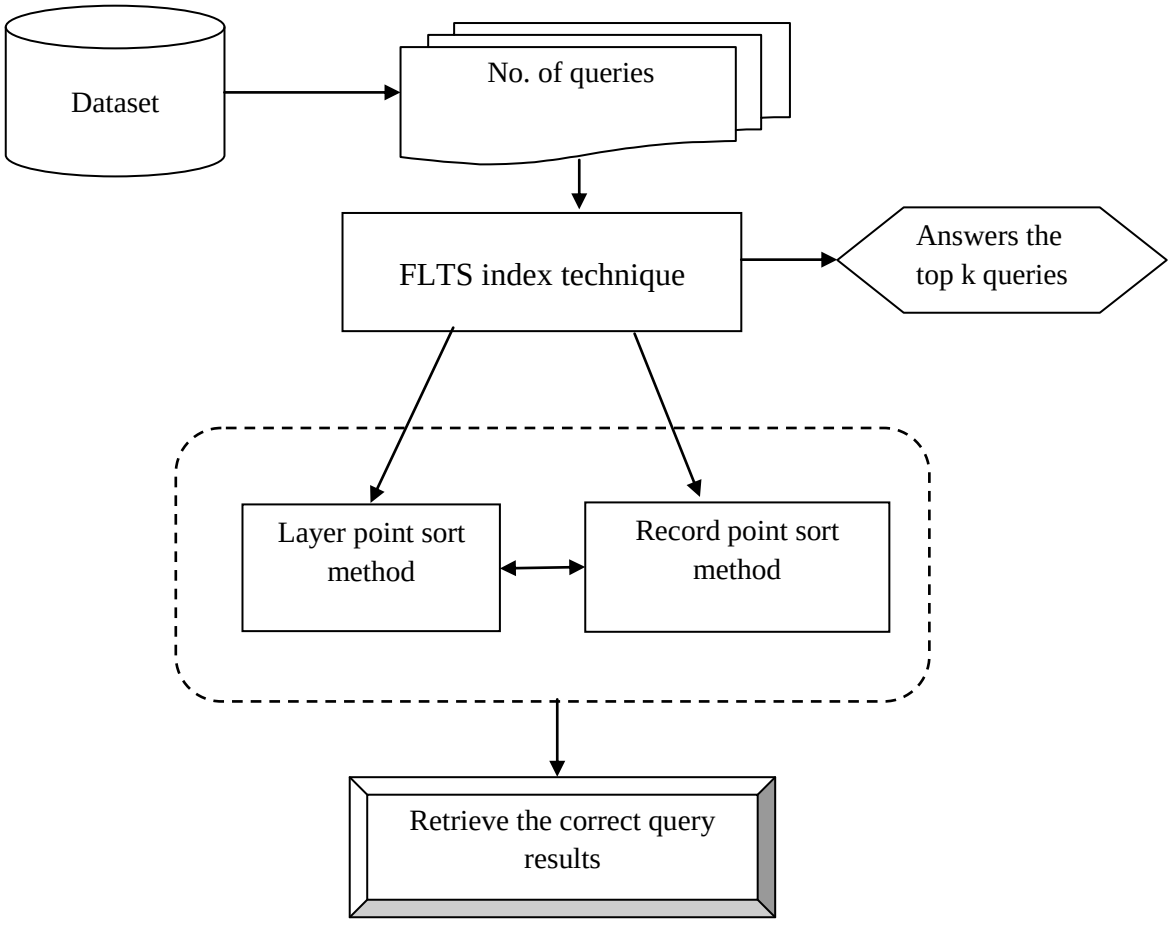

Figure 1. Structural diagram of Fusion Layer Topological Space Query Indexing

Figure 1 illustrates the design of Fusion Layer Topological Space Query Indexing technique for obtaining the accurate query result in distributed environment. The above processing step shows that a given dataset, the uncertain set of data are analyzed and processed. At first, the FLTS index technique answers the top k queries among the number of user requested queries in uncertain data. After that, FLTS query indexing uses the collective result of layer point sort method and the record point sort method. Based on the sorting method, the FLTS index formally proves in their dominant relationships. This helps to sorting the redundant tuple in query processing and also removes the correlation between them. The query processing techniques is explained in forthcoming subsection.

B.1. Construction of Fusion Layer Topological Space Index

Initially, a number of queries are expressed on any random subset of attributes in the uncertain data.The proposed indexing techniques answer the top k query in Fusion Layer Topological Space for improving the query processing. This indexing scheme answer the user queries and produces better search result rate on uncertain data based on the two types of sorting methods. The FLTS index constructed with two types of sorting method layer point sort method and record point sort method.

B.1.1. Layer point sort method

The layer point sort out method constructs a global index based on the group of all attribute values of each tuple. The tuples are divided into several layers in theindex, where the ith layer holds the tuples that is the top-i answer.Hence, the top-k answers are measured throughconsidering at most k layers.In layer point sort out method,tuples in the relation R are divided into a disjoint set of layers which is expressed as $\{L_1, L_2, L_3 \ldots L_n\}$, where L_i denotes the i^{th} layer. Each tuplesarrived with one and only one layer. Once the tuples are divided into layers, the top-k tuplesis attained from at most the first k layers.The number of tuples presents in the other layers can be removedusing layer point sorting method.

B.1.2. Record point sort method

The Record point sort approach constructs a set of records by sorting all tuples depends on their values in each attribute. After that, it determines the top-k tuples by combining the number of records are required. Under the Record point sort approach, each record is independent of one another. The top-k tuples are measured by accessing only those records similar to the attributes used in the query. Specifically, it also used to sortthe redundant tuples by individual consideration of these attribute values. Based on, FLTS index technique improving the relationship between the attributes when creating the sorted records, its performance gets increased as the number of attributes mentioned in the query also increased.

TABLE 1. TARGET RELATIONAL TABLE

ID	**Attributes**	
	Attribute 1	***Attribute 2***
T_1	0.2	1.0
T_2	0.1	0.3
T_3	0.3	0.4
T_4	0.65	0.1
T_5	0.5	0.6
T_6	0.75	0.45
T_7	1.0	0.5
T_8	0.4	0.8
T_9	0.8	0.9

DOI: 10.18535/ijecs/v5i10.48

In this method, for each layer,sorted record is expressed as$\{R_{i1}, R_{i2,}R_{i3} \ldots . R_{id}\}$where R_{ij} denotes the records of tuples in sorted with the ascending order of their j[th] attribute values.Let us consider that the relation R has nine tuples which is represents as $T_1, T_{2,} \ldots T_9$ and two attributes, Attribute 1 and Attribute 2. From that consideration, the ID represents the identifier of the tuples. The convex hull and tuple position at the vertices of this convex hullas shown in the figure 2. The construction of the layer record and the FLTS index in the two-dimensional space is shown in below figure.The target relation R is considered nine different tuples values which are represented as ID.

Table 1 shows the target relational table with two different attributes. The attributes contains the different values for constructing the FLTS index. Based on the attribute values, the convex hull are sketched and find the position of the tuples present in the relation table.

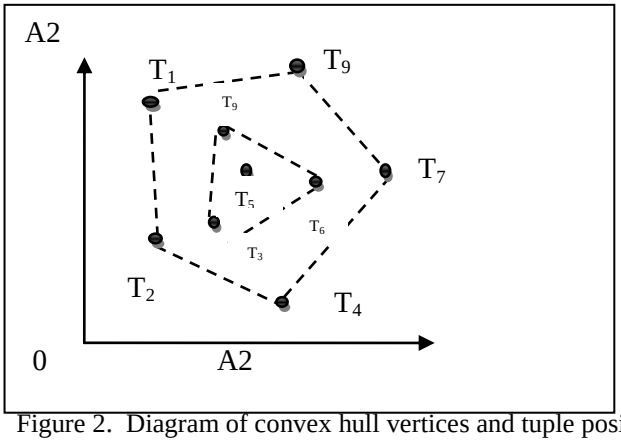

Figure 2. Diagram of convex hull vertices and tuple position

Figure 2 illustrates the outline diagram of the two attributes values which helps for constructing the FLTS index based on layer point sort method and record point sort method. Based on the above said attributes values, three records are maintained and it is arranged in the ascending order which is shown below.

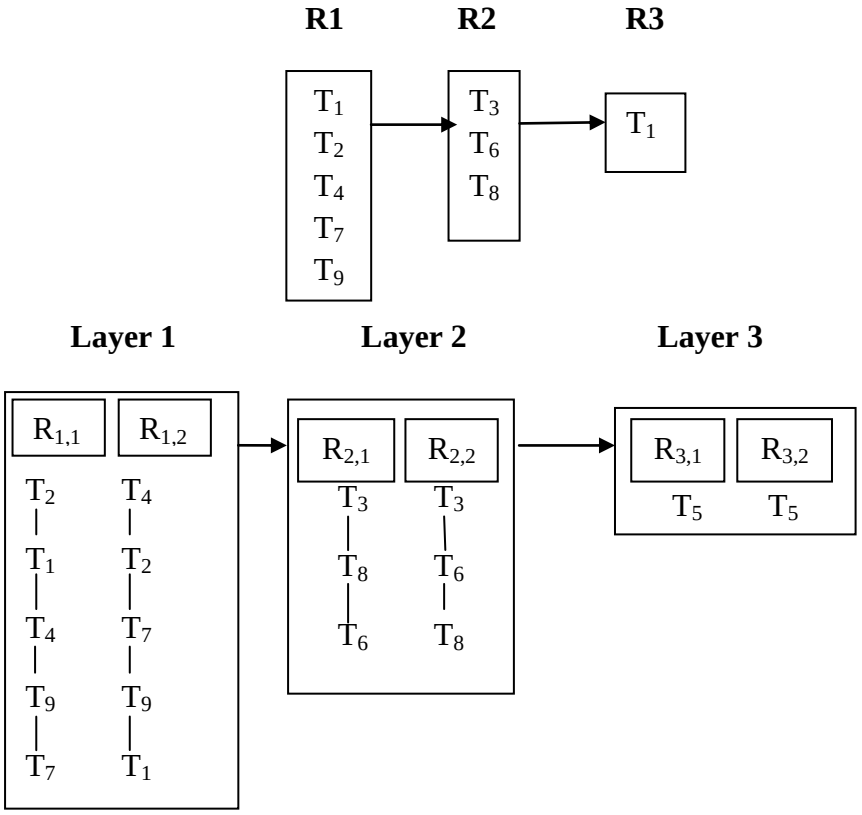

Figure 3. Construction of fusion layer topological indexes

From the figure 3, the fusion layer topological index is constructed using layer and record based approach. From the figure, the three different layersare considered.The relation table *R* is separated then two sorted records, $R_{1,1}$and $R_{1,2}$, for the five tuples in R_1. In particular, $R_{1,1}$ and $R_{1,2}$ records the tuples IDs in R_1that are arranged in ascending order of their attributes 1 and 2 respectively. Form the figure 3, the first tuples in $R_{1,1}$ is T_2 because it is the tuples with the leastAttribute 1 value in R_1. Repeat the process until the input set relational table R becomes empty.

In FLTS index,$R = \{L1 = R_{1,1}, R_{1,2,}R_{1,3} \ldots . R_{1,d}\}$ where each sorted record$R_{i,j}$contains only tuple identifiers (ID), but not thefull attributes values of the tuples. Then the full attributevalues are stored in a separate place. As a result, once atupleID is obtained from the FLTS index, then full valuesof its attributes is retrieved separately in order to compute the tuples scorein a given scoring function. The algorithmic description of the layer based record sort method is described for constructing the FLTS index.

Input : R relation table which contains a set of the tuples
Output : Constructing FLTS index
Step 1: Begin
Step 2: Layer number i is initialized with zero (i=0)
Step 3: I =i+1
Step 4: R : tuples at the vertices of the convex hull
Step 5: For each attributes
Step6: Sort the tuples with the ascending order of their j-th attribute values and store their identifiers as the record $R_{i, j}$
Step 7: Obtain the set of d sorted records
Step 8: This process is repeated until the input set R becomes empty
Step 9: End for
Step 10: End

Algorithm 1. Construction of FLTS index

The above algorithmic description shows that the construction of FLTS index for using layer point sort method and record point sort method. At first, the numbers of layer are considered and the relation table contains the attributes. For each attribute, the tuple sorting is performed then it is stored in the record. Followed by, the sorted records are identified. The process is repeated till the relation table is emptied. Based on, the redundant tuples are reduced. After the construction of FLTS index, the query processing is performed through the integration of the layer point sort method and the record point sort method.

B.1. Query processing using Integration of Layer point sort method and record point sort method

The FLTS query indexing uses the synergic result of integrates the layer point sort method and the record point sort method. FLTS index formally proves in their dominant relationships.In the FLTS indexconstruction algorithm, the input tuples in R are separated into three different layers by thefrequent extraction of the convex hull vertices.The top-k tuples are ensured in the initialk layers L_1throughout L_k. Therefore, during the top-k answers computation, all tuples in the layer L_{k+1} and beyond the layers is eliminated usingLayer point sort method.

Let us consider, $L = \{L_1, L_2, L_3 \ldots L_n\}$ is a set of layer constructed through the recursiveextraction of convex hull vertices. Then the minimum ratedtuple is represented as $T_{min}\ (L)$. Then, there is no other tuples presented in the layer $L = \{L_{i+1}, L_2, L_3 \ldots L_n\}$and it have the score value is less than $T_{min}\ (L)$. Initially, layer sort method is used to evaluate score value of the layer Li and evaluates their scores. Once the tuples scores are measured, it identifies the minimum-scored (i.e Rated) tuples $T_{min}\ (L_i)$ and then computes threshold factor H(i). In the designing of FLTS indexalgorithm, only subsets of tuples are retrieved from each layer.The minimum-rating tuples in the set Sand H (i) is measured. From the above discussion, $S_{i,j}$ (n) is

referred as set of the first n tuples at the top of the record $R_{i,j}$.TheThreshold factor H (i) is measured as follows,

$$H(i) = \{f_w(T) \leq f_w\ (T_{min}(L_i)\} \quad (3)$$

From (3), $f_w(T)$ is the linear scoring function of tuple and $T_{min}\ (L)$ minimum-scored tuples $T_{min}\ (L_i)$ from the layer. The algorithmic representation of the query processing is shown as follows,

Input : set of layer $L = \{L_1, L_2, L_3 \ldots L_n\}$, set of the tuples
Output : Obtain the top k answers
Step 1: For each layer
Step 2: Evaluate score value of all tuples$f_w\ (T)$ in layer
Step 3: Find the minimum ratedtuples$f_w\ (T_{min}\ (L_i))$ from the layer
Step 4: Calculate H (i) using (3)
Step 5: If $H(i) \geq k$ then
Step 6: Return the k tuples with the lowest score
Step 7: else
Step 8: Go to the next layer and repeats the process using step 2
Step 9: End if
Step 10: End for
Step 11: End

Algorithm 2. Query processing using Integration of Layer point sort method and record point sort method

Based on the above algorithmic description, query processing is done using the layer sort method and record point sort method. From that, the proposed FLTS index considerably reduces the number of tuples retrieved during query processing while guaranteeingthe correct query results.The individual records in the FLTS indextoperform record point sort out method by retrieving only a subset of tuples in each layer and reduces the redundant tuples based on the scored value.For any linear scoring function, the proposed correctly FLTS indexfinds top k query with minimum scores. This helps to decrease the more retrieved tuples present in the layer.Through the analysis of the integration of the two sorting methods, derive a fixed bound that reduces the number of tuple retrieved during query processing while guaranteeing the correct query results in distributed environments. Therefore, the query retrieving efficiency is increased and also the processing effectiveness is increased.

EXPERIMENTAL EVALUATION

In this section, an effective Fusion Layer Topological Space Query Indexing technique (FLTS)is evaluated using MATLABwith census income datasetand census income KDD data set extracted from UCI repository. Thecensus income datasets consists of 14 attributes with categorical and integer characteristics. In census income datasets,the prediction of uncertain data is measured whether it exceeds '50 $dollar/year$'. The extraction of census income dataset was madethrough Barry Becker from the 1994 Census database. A set of logically clean records was extracted based on the conditions ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0)).Prediction task is used in that dataset to find a person creates over 50K per year. The attributes in the datasets are age, work class, fnlwgt, education, education-num, marital status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week and native-country.

The census income KDD data set includes 40 attributes and ithave weighted census data extracted from the year of 1994 and 1995 present population investigation performed through the U.S. Census Bureau.The instance weight signifies the more people in the population that each record represents due to stratified sampling.This dataset consist of 199523 instances in the data file whereas 99762 instances used in the test file. The different attributes are used such as age AAGE, class of worker ACLSWKR,industry code

ADTIND, adjusted gross income AGI, sex ASEX, and wage per hour AHRSPAY,total person income PTOTVAL and so on. In order to conduct the experiments, FLTS index technique used six important attributes like, age, work-class, education, and occupation, hours-per-week and native country.

RESULT ANALYSIS

To perform the efficiency of the proposed Fusion Layer Topological Space Query Indexing technique (FLTS)against the U Quadtree indexing was introduced in [1] andProbabilistic inverted (PI) index [2]. The experimental evaluation is carried out with the different parameter such as queryretrieval efficiency, response time, memory consumption for query processingand system scalability. Performance is evaluated along with the following metrics with help of tables and graph values.

A. Impact of query retrieval efficiency

Query Retrieval Efficiency (QRE) of the FLTS index is defined as the rate at which the required information is retrieved based on the users' requestedqueries. The QRE is measured in terms of percentage (%). The following expression is used to calculate the efficiency.

$$QRE = \frac{Correctly\ retrieved\ user\ qeries}{No.of\ user\ queries} * 100 \quad (4)$$

From (4), Query Retrieving Efficiency (QRE) is measured based on the number of user queries used in the experimental evaluation.

TABLE 2. TABULATION FOR QUERY RETRIEVAL EFFICIENCY

No. of queries (Q)	**Query retrieval efficiency(%)**		
	FLTS index	***U Quadtree indexing***	***PI index***
5	83.10	75.20	66.35
10	84.35	76.35	68.23
15	86.35	77.10	69.10
20	88.20	78.36	71.36
25	91.13	79.10	72.12
30	92.32	80.31	73.84
35	93.41	81.20	74.65

Table 2 illustrates the query retrieval efficiency is measured based on the number of queries used in the experimental evaluation. From the table value, the proposed FLTS index is compared with the existing indexing techniques such as U Quad tree indexing [1] and PI index [2]. The results reported here it confirms that the FLTS index provides the improved results than the other existing methods.

DOI: 10.18535/ijecs/v5i10.48

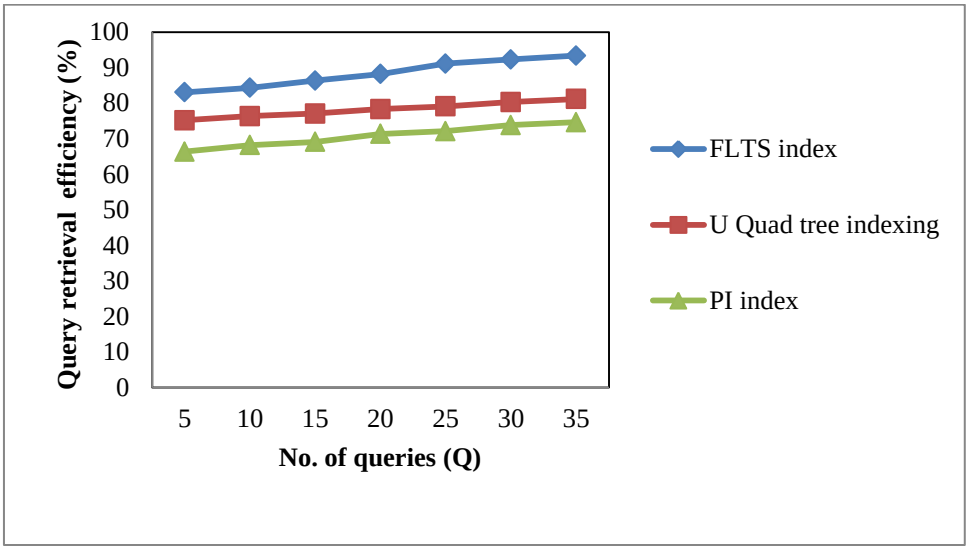

Figure 4. Measure of Query retrieval efficiency

Figure 4 describes the query retrieval efficiency which is measured based on the number of queries used. The queries taken for the experimental consideration is varied from 5 to 35. From the figure it is clearly evident that the proposed FLTS index technique improves the query retrieval efficiency than the state –of-the-art methods [1][2]. Due to, the FLTS index technique uses integrating of the layer point and record point sort methods. These methods are used to sort the redundant tuples by individual consideration of these attribute valuesfor queries on uncertain data. Based on, FLTS index improving the relationship among the attributes when creating the sorted lists.The performance of query retrieval efficiency is increased by 11% compared to existing U Quad tree indexing [1]. In addition, the integration of the two sorting methods is used to obtain a fixed bound that reduces the number of redundant tuple retrieved during query processing. This helps toensure the correct query results in distributed environments. Hence the retrieval rate is 20% high in the proposed FLTS index technique than the existing PI index [2].

B. Impact of response time

Response time is the total amount of timethe FLTS index takes to respond a request for analyzing query. The response time is measured in terms of milliseconds (ms). The following mathematical expression is measured as follows,

$$Response\ time = Number\ of\ query * Time\ (respond\ the\ user\ requested\ query) \quad (4)$$

TABLE 3. TABULATION FOR RESPONSE TIME

No. of queries (Q)	**Response time (ms)**		
	FLTS index	***U Quadtree indexing***	***PI index***
5	18	25	29
10	20	27	31
15	22	30	35
20	25	32	36
25	28	36	38
30	29	38	41
35	30	40	47

DOI: 10.18535/ijecs/v5i10.48

Table 3 illustrates the response time based on the requested queries. The proposed FLTS system uses minimum time for respond the user requested query than the other two methods namely Quad tree indexing [1] and PI index [2]. From the tabulated value, the graph is plotted and shows their performance.

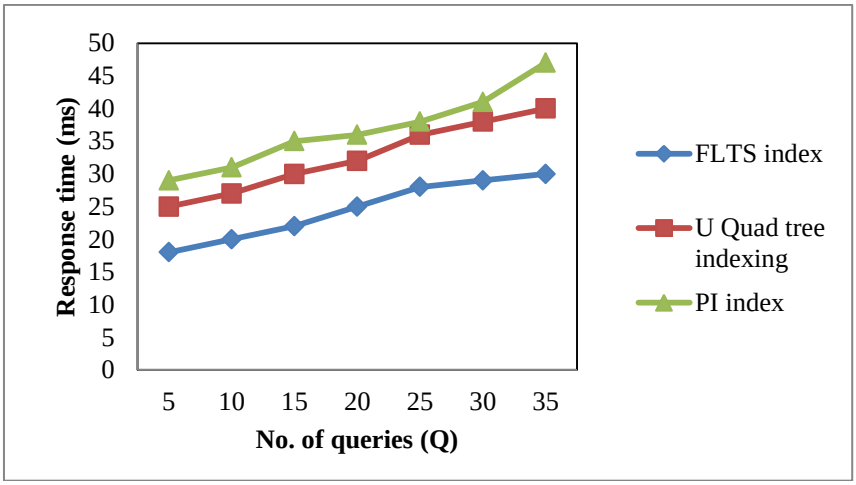

Figure 5. Measure of response time

Figure 5 illustrates the targets on measuring the response time based on number of queries as input. The figure shows, while increasing the number of user queries, the response time also gets increased in all the methods. But comparatively, it is reduced in the proposed FLTS than the other methods [1] [2]. This is because, in FLTS index technique, the queries are expressed on any separation of attributes in the uncertain data.The FLTS index technique answers the top-k queries effectively. The proposed FLTS index determines the topk query with minimum scores for any linear function. This helps to improve the query retrieval efficiency in the layer. Therefore, the proposed techniques answer the accurate query results with minimum response time. Through the analysis of the sorting methods, query processing is ensured to obtain thecorrect query results. The response time is considerably reduced by 33% and 50% using FLTS index technique compared to U Quad tree indexing [1] and PI index [2] respectively.

C. Impact of Memoryconsumption for query processing

Memory consumption is defined as the amount of storage space utilized by the processer (CPU) for processing the given query. It is difference between the total memory of processor and the unused memory. It is measured in terms of (MB).

$$Memory\ consumption = Total\ memory\ space - unused\ memory\ space \quad (5)$$

TABLE 4. TABULATION FOR MEMORY CONSUMPTION FOR QUERY PROCESSING

Number of queries (Q)	**Memory consumption for query processing (MB)**		
	FLTS index	***U Quadtree indexing***	***PI index***
5	96	152	160
10	108	158	168
15	120	160	167
20	132	163	170
25	141	168	200
30	146	170	235
35	153	171	240

DOI: 10.18535/ijecs/v5i10.48

Table 4 describes the Memory consumption during the query processing based on the number of query consider for experimental evaluation. While increasing the query, the memory consumption gets increased in all the three methods. But, the FLTS indexing technique utilizes minimum memory space for storage processing compared toU Quad tree indexing [1] and PI index [2].

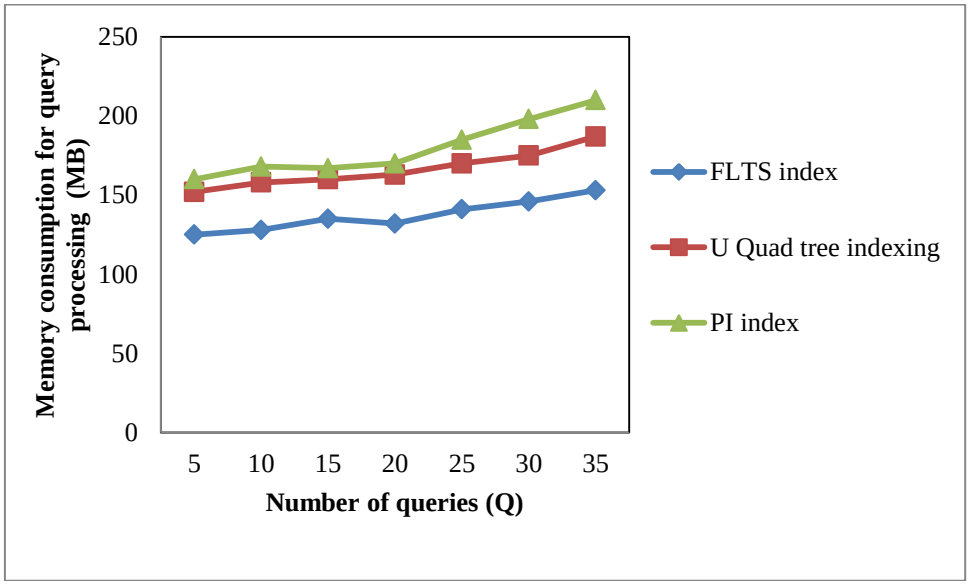

Figure 6 Measure of Memory consumption

The memory consumption with the different number of user requested queries is shown in figure 6. From that, it is clearly illustrates that the proposed FLTS index technique utilizes the less amount of memory than the other Methods [1] [2]. The storage space required for two census income dataset and 35 different queries during the experimental evaluationwith different time intervals. As shown in the figure with the increase in the number of queries, the storage space gets increased. But comparatively, the memory consumption is reduced. Due to, the proposed FLTS index construction algorithm requires less memory utilization for improving the query processing and mining uncertain data. The memory consumption is reduced by 21% and 31% using FLTS index compared to existing U Quad tree indexing [1] and PI index [2] respectively.

D. Impact of Scalability

The main objective of theFLTS index is to develop techniques for scalable query processing in databases.Scalability of the systemdescribes its abilityof the proposed methods and increases its level of performance by larger operational demands.

TABLE 5. TABULATION FOR SCALABILITY

Methods	Scalability (%)
FLTS index	89.35
U Quad tree indexing technique	74.20
PI index	68.35

Table 5 describes the performance of scalability based on the three techniques FLTS index, U Quad tree indexing technique [1] and PI index [2]. The scalable query processing is obtained in the propped FLTS index compared to existing methods.

DOI: 10.18535/ijecs/v5i10.48

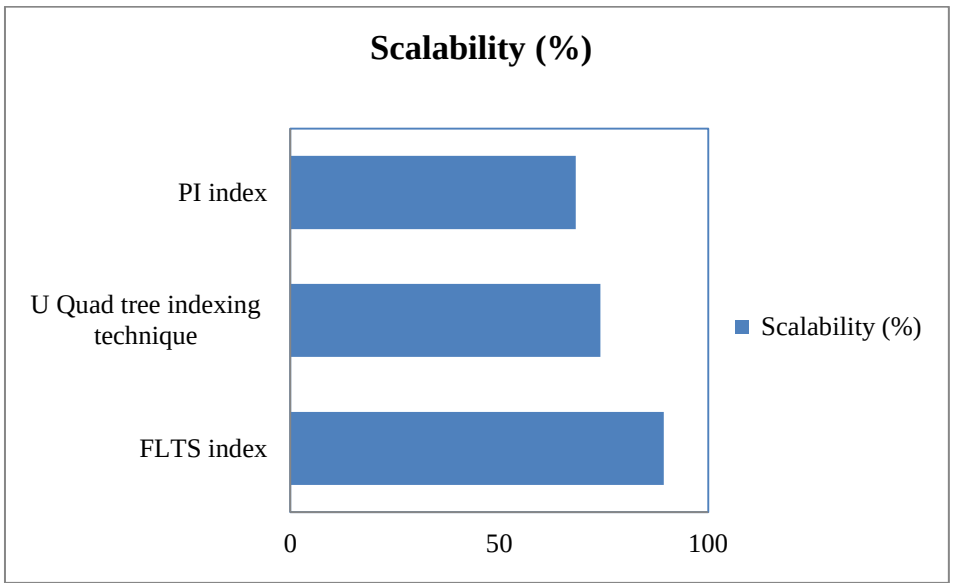

Figure 7 Measure of scalability

From the figure 7, the scalabilityperformance during the query processing is obtained with the different indexing techniques. The proposed FLTS index technique improves the scalability level during the query processing. The proposed technique providesbetter solutionunder uncertain data sets with the higher efficiency and scalability using the layer point sort method. The FLTS indexanswers the top k queries between the numbers of user requested queries in uncertain data. It uses the integrating result of layer point sort method and the record point sort method. Based on the sorting method, the FLTS index proves in their dominant relationships and it reduces the redundant tuple in query processing and also removes the correlation between them. By applying linear functions, it resulted in efficient scalability based on CPU and I/O costs. The scalability of the proposed FLTS index improves the performance results of 17% and 24% compared to U Quad tree indexing technique [1] and PI index [2] respectively.

CONCLUSION

In this paper, a novel technique called as Fusion Layer Topological Space Query Index(FLTS) is developed. At first, the queries are expressed on any random separation of attributes in the uncertain data. TheFLTS index answers top k queries among the number of user requested queries in uncertain data. Then the FLTS query indexing uses the integrative result of layer point sort method and record point sort method. In the construction of FLTS index, the layer point sort method is used to sort the tuple totally on the combination of all attribute values of the tuple. The record point sort method is used for scoring the tuple to obtain the accurate query results. After that, the query processing is done with the help of the two sort methods. This helps to retrieve only a subset of tuples in each layer and reduces the redundant tuples based on the scored value. Therefore, the query retrieving efficiency is increased and obtaining the correct query results in distributed environments.Experimental evaluation is carried out to measure the query retrieving efficiency, response time, memory consumption and system scalability. The performance results shows that the proposedFLTS index technique considerably increases the query retrieving efficiency and system scalability with minimum query response time. The memory utilization also reduced using the construction of FLTS index than the state-of-the-art methods.

REFERENCES

[1] Ying Zhang, Wenjie Zhang, Qianlu Lin, Xuemin Lin and Heng Tao Shen, “Effectively Indexing the Multi-dimensional Uncertain Objects”, IEEE Transactions on Knowledge and Data Engineering, Volume 26, Issue 3, March 2014, Pages 608 – 622

[2] Jianxin Li., Chengfei Liu., Rui Zhou., and Jeffrey Xu Yu., "Quasi-SLCA based Keyword Query Processing over Probabilistic XML Data", IEEE Transaction on Knowledge and Data Engineering, Volume 26, Issue 4, 2014, Pages 957 – 969

[3] Zhi-Jie Wang, Dong-Hua Wang, Bin Yao, and Minyi Guo, "Probabilistic Range Query over Uncertain Moving Objects in Constrained Two-Dimensional Space", IEEE Transactions on Knowledge and Data Engineering, Volume 27, Issue 3, March 2015, Pages 866-879

[4] Wenjie Zhang, Liming Zhan, Ying Zhang, Muhammad Aamir Cheema, Xuemin Lin, "Efficient top-k similarity join processing over multi-valued objects", World wide Web, Springer, Volume 17, Issue 3, May 2014, Pages 285–309

[5] Yanming Nie., Richard Cocci., Zhao Cao., Yanlei Diao., and Prashant Shenoy., "SPIRE: Efficient Data Inference and Compression over RFID Streams," IEEE Transactions on Knowledge and Data Engineering, Volume 24, Issue 1, January 2012

[6] Wen-Ping Li, Jing Yang, Jian-Pei Zhang, "Uncertain canonical correlation analysis for multi-view feature extraction from uncertain data streams", Neuro computing, Elsevier, Volume 149, 2015, Pages1337–1347

[7] Dong Xie, Jie Xiao, Guangjun Guo, and Tong Jiang, "Processing Uncertain RFID Data in Traceability Supply Chains",The Scientific World Journal, Hindawi Publishing Corporation Volume 2014, March 2014, Pages 1-22

[8] Zhenhua Huang, Jiawen Zhang and Qiang Fang, "Efficient Query Processing Platform for Uncertain Big Data", International Journal of Database Theory and Application, Volume 8, Issue 5, 2015, Pages 149-160

[9] Hyountaek Yong, Jongwuk Lee, Jinha Kim, Seung-won Hwang, "Skyline ranking for uncertain databases", Information Sciences, Elsevier, Volume 273, 2014, Pages 247–262

[10] Rui Zhu, Bin Wang, Guoren Wang, "Indexing Uncertain Data for Supporting Range Queries", Information Sciences, Elsevier, Volume 273, 2014, Pages 247–262

[11] Lei Xu, Qinghua Hu, Edward Hung, Chi-Cheong Szeto, "A heuristic approach to effective and efficient clustering on uncertain objects", Knowledge-Based Systems, Elsevier, Volume 66, August 2014, Pages 112–125

[12] Keji Mao, Xiaomin Zhao, Qike Shao,Wenxiu He, Yanqiang Ou, and Qingzhang Chen, "Data Storage Scheme Supporting for Multidimensional Query", Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks, Volume 2013, April 2013, Pages 1-8

[13] Jie Xu, Dmitri V. Kalashnikov, and Sharad Mehrotra, "Efficient Summarization Framework for Multi-Attribute Uncertain Data", ACM, SIGMOD International Conference on Management of Data, June 2014, Pages 421-432

[14] Levent Unver Tafan I. Gundem, "Authentication of uncertain data based on k-means clustering", Turkish Journal of Electrical Engineering & Computer Sciences, Volume 24, Issue 4, January 2016, Pages 2910-2928

[15] Linlin Ding, Yu Liu, Baoyan Song, and Junchang Xin, "Efficient ELM-Based Two Stages Query Processing Optimization for Big Data", Mathematical Problems in Engineering,Hindawi Publishing Corporation, Volume 2015, Pages 1-12, November 2014

[16] Haidar AL-Khalidi, David Taniar, Maytham Safar, "Approximate algorithms for static and continuous range queries in mobile navigation", Computing, Springer , Volume 95, Issue 10, Pages 949–976, October 2013

[17] Peter Braun, Juan J. Cameron, Alfredo Cuzzocrea, Fan Jiang and Carson K. Leung, "Effectively and Efficiently Mining Frequent Patterns from Dense Graph Streams on Disk", Procedia Computer Science, Elsevier, Volume 35, 2014, Pages 338-347

[18] C.K. Leung, A. Cuzzocrea, F. Jiang, "Discovering frequent patterns from uncertain data streams with time-fading and landmark models", LNCS Transactions on Large-Scale Data- and Knowledge-Centered Systems, Volume 8, 2013, Pages 174–196

[19] Wang Hanning, Xu Weixiang, Jiulin Yang, Lili Wei, and Jia Chaolong, "Efficient Processing of Continuous Skyline Query over Smarter Traffic Data Stream for Cloud Computing", Hindawi Publishing Corporation, Discrete Dynamics in Nature and Society, Volume 2013, November 2013, Pages1-10

[20] S. Wang, L. Wu, F. Zhou, C. Zheng, H. Wang, "Group Pattern Mining Algorithm of Moving Objects' Uncertain Trajectories", International Journal of Computers Communications & Control (IJCCC), Volume 10, Issue 3, June, 2015, Pages 428-440

M. Kalavathi is a Head, Department of Computer Science, Govt. Arts & Science College, Komarapalayam. She received the M.Sc., Degree from Bharathiar University in 2004, M.Phil Degree from Annamalai University in 2006, respectively in Computer Science. Her research interest includes Data Mining and Digital Image Processing.

Dr. P. Suresh is a Head, Department of Computer Science, Salem Sowdeswari College [Govt. Aided], Salem. He received the M.Sc., Degree from Bharathidasan University in 1995, M.Phil Degree from Manonmaniam Sundaranar University in 2003, M.S (By Research) Degree from Anna University, Chennai in 2008, PGDHE Diploma in Higher Education and Ph.D., Degree from Vinayaka Missions University in 2010 and 2011 respectively in Computer Science. He is an Editorial Advisory Board Member of Elixir Journal. His research interest includes Data Mining and Natural Language Processing. He is a member of Computer Science Teachers Association, New York.