Hindawi
Mobile Information Systems
Volume 2022, Article ID 2887395, 9 pages
https://doi.org/10.1155/2022/2887395

Research Article

Application of the R-Tree Clustering Model in Medical Information Retrieval

Yun Dai and Hao Liu

Department of Statistical Information,
Liyuan Hospital of Tongji Medical College of Huazhong University of Science and Technology, Wuhan 430077, China

Correspondence should be addressed to Yun Dai; 2015ly0915@hust.edu.cn

Received 8 June 2022; Revised 3 July 2022; Accepted 13 July 2022; Published 11 August 2022

Academic Editor: Muhammad Zakarya

Hospitals produce a large amount of medical information every day. In the face of medical big data, the existing data processing methods cannot meet expectations and need to be continuously optimized. In the database system, when the stored objects are very large, and then the efficiency of data retrieval is a major bottleneck, therefore restricting the application of medical information. For that reason and to improve the efficiency of information retrieval, it is necessary to add an index to the information object and filter the dataset participating in the connection retrieval through the index. In this paper, an information retrieval technique grounded on the R-tree clustering model index is proposed for massive hospital information. The R-tree clustering model is constructed in massive hospital information by using the dynamic determination clustering center (DCC) algorithm. Finally, the superiority of the method is proved by simulations. The experiments and empirical evaluation show that the proposed R-tree clustering model index significantly improves data retrieval efficiency.

1. Introduction

With the prompt growth of medical diagnosis technologies such as medical and health information systems, Internet of things, big data, and high-throughput sequencing, the medical and health field is gradually entering the "big data era" [1]. Using big data and Internet of things can improve medical quality and solve current problems, but its unique characteristics (large quantity, fast growth, various types, and difficult to determine accuracy) also pose challenges to technology and management. Since many hospitals have digitized their administrative and treatment procedures, the generation speed and quantity of data exceed the limitations of traditional data processing software [2]. Complex data forms greatly increase the difficulty of storing, mining, and analyzing data. In the database system, when the stored objects are very large, the efficiency of data retrieval is an important bottleneck that limits the application of medical information [3, 4]. Therefore, improving the information retrieval ability in the context of medical big data is of prodigious importance to develop and enhance the level of medical services and encourage the construction of medical informatization. The medical big data includes basic data such as electronic medical records, residents' behavioral health, detection reports, diagnosis and treatment data, medical images, economic data, and medical management. In fact, it is characterized by large scale, fast growth, diverse structure, and high application value.

In order to acquire the required statistics from the massive medical data, retrieval technology must be used as a support [5]. At present, there are many methods for database indexing, such as fixed grid method [6], quadtree method [7], and R tree (and its variants) [8] index. Because the R-tree clustering model index has the advantages of dynamic, great efficiency, and great aggregation, therefore, it has become one of the maximum extensively used and well-established index technologies in the literature [9, 10]. According to the characteristics of various types of medical data and large amount of data, combined with the R-tree clustering model index, this paper proposes a method to quickly retrieve and process medical data. The proposed method uses the R-tree clustering model to retrieve medical data through the R-tree

clustering model index, so as to increase the efficiency of the information retrieval system.

In the face of medical big data, the existing data processing methods cannot meet expectations and need to be continuously optimized. In the database system, when the stored objects are very large, the efficiency of data retrieval is a major bottleneck which, in fact, restricts the application of medical information. Therefore, it is necessary to add an index to the information object (within the large database) and filter the dataset participating in the connection retrieval through the index to increase the effectiveness of the information retrieval system. The major contributions of this research are as follows:

(i) A retrieval method based on the R-tree clustering model index is proposed for massive hospital information

(ii) The R-tree clustering model is constructed in massive hospital information by using the dynamic determination clustering center (DCC) procedure

(iii) The model of the R-tree clustering index improves data retrieval efficiency

The remaining of the manuscript is organized in the following manner. The establishment of an R-tree clustering model is discussed in section 2. Moreover, a dynamic algorithm is demonstrated that can determine the centers of the clusters. In section 3, the proposed algorithm is tested in terms of its application simulation in medical information retrieval. Finally, section 4 summarizes the paper and gives some insights into future research.

2. Establishment of the R-Tree Clustering Model

2.1. A Dynamic Algorithm for Determining Cluster Centers. In order to accomplish an effective contact to large-scale data, this paper uses the clustering model based on the R-tree clustering model index to retrieve data from the medical information system. Setting the clustering center in advance will cause the final clustering result to differ from reality when building the R-tree clustering model if the data distribution rule is unknown. Therefore, affecting the efficiency of the index of the constructed R-tree clustering model [6, 11]. This study introduces the DCC approach to build an R-tree clustering model, efficiently determining the clustering center [12].

Definition 3.1. Set the distance index of measuring adjacent objects as r [13], expressed as the following formula:

$$R = \frac{1}{\sqrt{(m/D)}}. \tag{1}$$

In formula (1), m is the quantity of spatial data, D is the given spatial area range, and d_i denotes the space from data to i. In case $d_i \le R$, then mark i as the contiguous entity of data. Similarly, if $d_i > R$, then mark i as a noncontiguous entity of data [14, 15].

Given that $r_1, r_2, \ldots, r_m$ is a set of R^d spatial data of m, and c_l is the center of the cluster l, at that point the distance function, i.e., distance among r_i and c_l can be expressed as expressed by the following formula:

$$d(r_i, c_l) = \sqrt{\left(r_i^1 - c_l^1\right)^2 + \left(r_i^2 - c_l^2\right)^2 + \cdots + \left(r_i^d - c_l^d\right)^2}. \tag{2}$$

Let the sample of class l be expressed as the following formula:

$$S_l = \{c_{l1}, c_{l2}, \ldots, c_{\ln}\}. \tag{3}$$

If formula (3) contains n data, the average point of this category is expressed as the following formula:

$$c_l = \left(c_l^1, \ldots, c_l^k, \ldots, c_l^d\right). \tag{4}$$

In formula (4), c_l^k is the k attribute of c_l, which is expressed as the following formula :

$$c_l^k = \frac{r_{l1}^k + r_{l2}^k + \cdots + r_{\ln}^k}{n}. \tag{5}$$

When choosing a cluster center, initially acquire the value point (mean), denoted by c_l, of the data in the class [16, 17]. In the next stage, calculate the distance between the mean value point and other data, and obtain the adjacent objects of c_l while agreeing to the distance index R. In the third stage, compute the value point (mean), denoted by c_l', of the adjacent objects. Finally, choose the spatial data which is the nearby to the c_l' as the cluster center. The final stage is expressed mathematically as given by the following formula:

$$r = \arg\min\left(c_l', r\right), \quad r\{r_{l1}, r_{l2}, \ldots, r_{\ln}\}. \tag{6}$$

In algorithm 1, the first line is the process of calculating the mean point, and c_l is the cluster center. Lines 2–7 get all neighboring objects. Line 3 calculates the distance between each spatial data and c_l. If it is a neighboring object, it is put into the set M (lines 4–6). Line 8 calculates the mean point c_l' of adjacent objects in the set M. In line 9, find the data closest to c_l' in the spatial data as the cluster center r_c. If r_c is not unique, the cluster measure function is used to compare it, and the number with the smallest convergence value is selected as the cluster center.

2.2. Constructing the R-Tree Clustering Model. By means of the dynamic R-tree clustering model and the proposed algorithm, the sensible leaf nodes are interleaved into the destination object. Furthermore, the above dynamic determination clustering center algorithm is used to build, so as to realize the dynamic optimization of the R-tree clustering model at a large-scale [18, 19]. For the generation of the R-tree clustering model for any spatial data set, the main process is as follows: first, the minimum boundary rectangle is established for all spatial objects. Then, the base rectangle is grouped according to the DCC algorithm. For example, in Figure 1(a), we first select R12 which, in fact, is nearby to the average point as the preliminary clustering center, and then $k = 1$. Then, we select R19 which is the farthest from R12, and R8 which is the farthest from R19, as clustering centers and start clustering [20].

Input: mR^d space data $S = \{r_1, r_2, \ldots, r_n\}$
Output: cluster center r_c
(1) $c_l = (c_l^1, \ldots, c_l^k, \ldots, c_l^d)$;
(2) for $i = l$ to m do
(3) Computing $d(r_i, c_l)$;
(4) if $d(r_i, c_l) < R$;
(5) $r_i \in M, M = \{r_b | d(r_b, c_l) \leq R, b \leq i\}$;
(6) end if
(7) end for
(8) Computing c_l';
(9) $r_c = \arg\min(d(r_c, c_l')), r_c \in S$;
(10) Output cluster center r_c

Algorithm 1: Build algorithm 1: DCC.

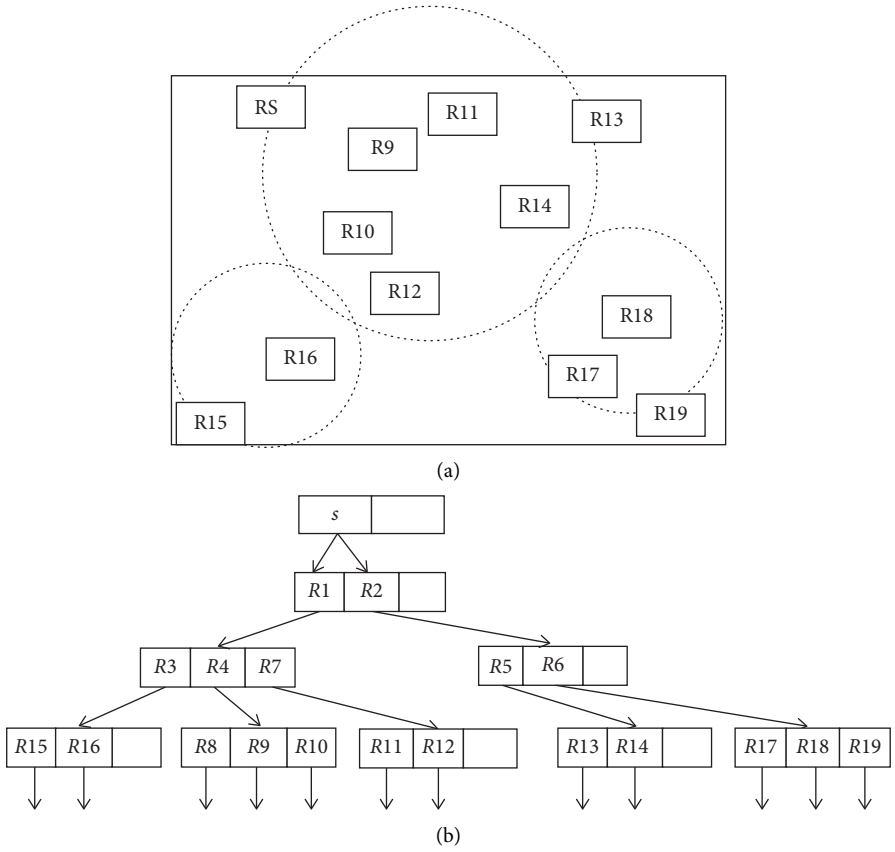

Figure 1: Hierarchy and its R-tree index. (a) Three cluster centers. (b) Record in table.

The R13, R14, R17, and R18 are divided into R19, and R9, R10, R11, R12, R15, and R16 are divided into R8. At this time, two clusters ($k = 2$) are formed, and the cluster center along with the cluster measurement function are computed. Subsequently, we select the cluster having the biggest radius and its cluster center R12 from the two clusters. Then, we select R15 which is the farthest from R12, R11, and R18 which are the farthest from R15 as the cluster center for reclustering. After that, we then calculate its cluster center and cluster measurement function (at this time, $k = 3$). In a cycle, the value of k upsurges as far as the converge of the clustering function occurs. Finally, a rectangle comprising entirely spatial objects in the whole area is shaped to obtain the R-tree clustering model. The entire process is shown in Figure 1(b).

Algorithm 2 is the process of constructing the R-tree clustering model through the DCC algorithm. The first line clusters the data, and the second to fifth lines construct the subtree layer-by-layer from the root.

2.3. Information Retrieval. A whole index establishment, retrieval, and node deletion mechanism is built into the R-tree clustering model itself. The R-tree clustering model

```
Inputs: S = {r_1, r_2, ..., r_n}
Outputs: root
(1)  M = DCC(S);
(2)  for i = 1 to j do
(3)     root ⟶ child(i) = M_i;
(4)     Construction_R(root ⟶ child(i), M_i);
(5)  end for
```

Algorithm 2: The build algorithm 2: Construction_R.

```
Inputs: N denotes the type of the R-tree
   W is the rectangle of request.
Outputs: return a suitable rectangle for the input W
(1) if level of N = 0 then
(2)     return 0;
(3) else
(4)     for j = 1, 2, ..., n do
(5)         if W intersets with the N.MBR then
(6)             R_Search (W, N.p_j);
(7)         end if
(8)     end loop
(9)  end if
(10) return the data rectangles for W.
```

Algorithm 3: The build algorithm 3: R_Search.

index is used to obtain the geometric data from the database [21]. The R-tree clustering model index can be created in the database to significantly increase the speed of multi-user data retrieval.

Algorithm 3 is used to discover all the data rectangles overlapping WN in the R-tree clustering model along with the root node N.

3. Application Simulation in Medical Information Retrieval

3.1. Simulation Parameters. In this paper, the R-tree clustering model is built in the medical record management system for simulation to verify the system performance after the integration of R-tree. Compared with the hash index [22, 23], the multidimensional analysis of the system performance in various cases is carried out. IoT devices are the infrastructure for hospitals to collect data. Health information gathered by medical wristbands is one example of a group of devices that each have their own management server for. The volume of data affects the size of R-tree. Set T as the data volume and W is the network bandwidth resource [24]. N represents the number of searches and Q represents the search complexity. The simulation variable α represents the available bandwidth and β represents the workload. Relevant parameter settings of simulation are shown in Table 1.

3.2. Simulation Results. In order to study the influence of network status and the task size on the index of the R-tree clustering model, the simulation parameters set in this paper are: ① $\alpha = 0, \beta = 0$ (that is, the network is idle and the task is small), and the results are shown in Figure 2. ② $\alpha = 1, \beta = 0$ (that is, the network is busy and the number of tasks is small), and the results are shown in Figure 3. ③ $\alpha = 0, \beta = 1$ (that is, the network is busy and the number of tasks is small), and the results are shown in Figure 4. ④ $\alpha = 1, \beta = 1$ (that is, the network is busy with a large number of tasks), and the results are shown in Figure 5. The ordinates are all simulation times.

Table 1: The experimental parameters and their values.

Parameter	Numerical value
W	100 MB
T_1, T_2	0.47 MB, 4.48 MB
N_1, N_2	5,549, 12,068
Q_1, Q_2	1, 4
$\alpha = 0$	100 MB
$\alpha = 1$	50 MB
$\beta = 0$	$T_1 + N_1 + Q_1$
$\beta = 1$	$T_2 + N_2 + Q_2$

3.3. Discussion. The retrieval time of the system using the R-tree clustering model was observed, initially, slower than the hash index, i.e., in particular, when the task volume is modest, but over time, the system speed is noticeably increased (as shown in Figures 2 and 3), especially when the network is idle (as shown in Figure 2). This paper analyzes the reasons for this result: the system deploying the R-tree

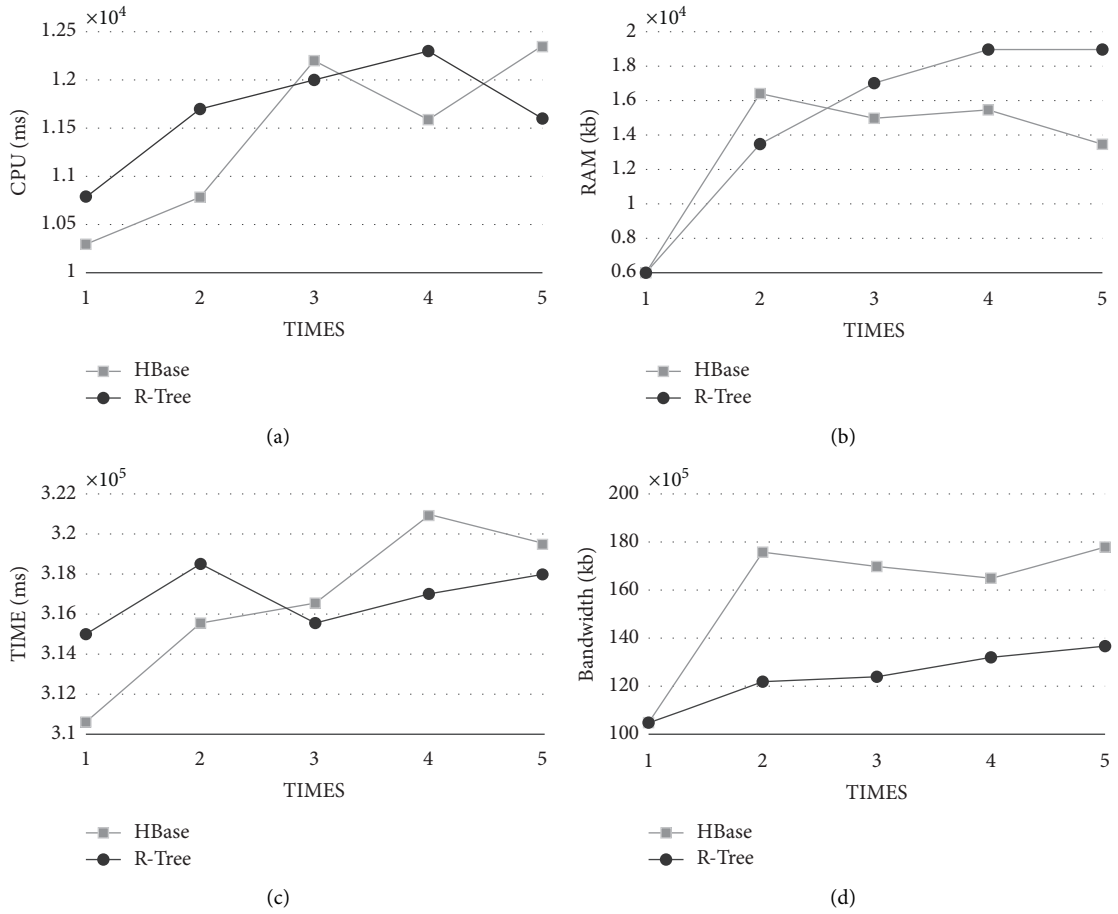

Figure 2: The R-tree index effect in terms of CPU running time, RAM usage, system running time, and bandwidth ($\alpha = 0, \beta = 0$). (a) CPU running time. (b) Ram occupancy. (c) System runing time. (d) Bandwidth.

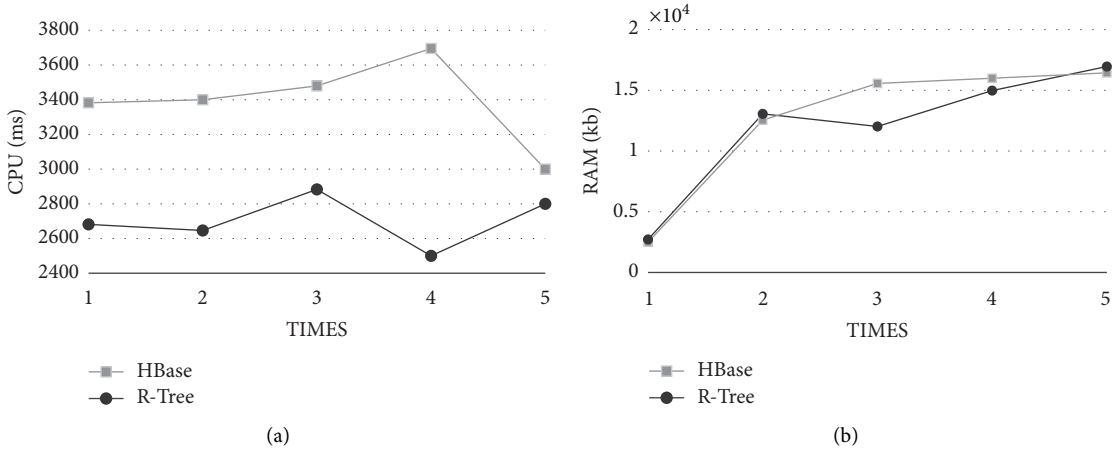

Figure 3: Continued.

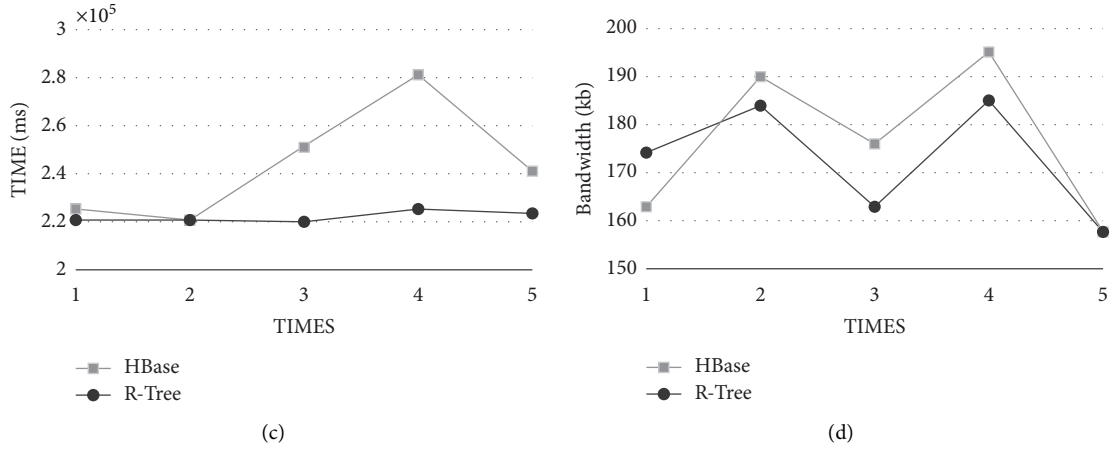

Figure 3: The R-tree index effect in terms of CPU running time, RAM usage, system running time, and bandwidth ($\alpha = 1, \beta = 0$). (a) CPU running time. (b) Ram occupancy. (c) System runing time. (d) Bandwidth.

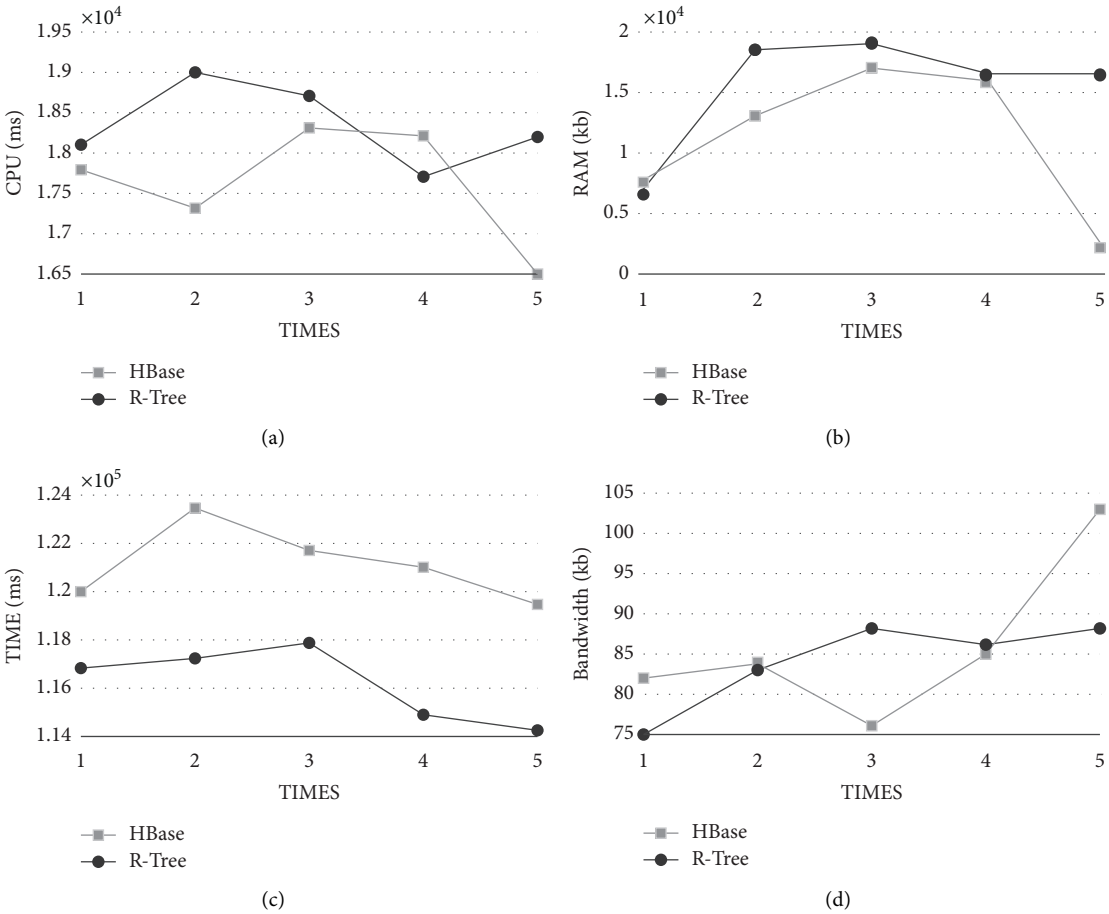

Figure 4: The R-tree index effect in terms of CPU running time, RAM usage, system running time, and bandwidth ($\alpha = 0, \beta = 1$). (a) CPU running time. (b) Ram occupancy. (c) System runing time. (d) Bandwidth.

clustering model needs to establish the R-tree at the beginning of retrieval. The time complexity of R-tree built by implementing the proposed techniques and algorithms is $O(n^k \times t)$, where k represents the total amount of clusters, n represents the total amount of data, and t characterizes the total number of iterations. Furthermore, the hash index method computational complexity is $O(1)$. Therefore, at the beginning of system operation, the R-tree clustering model index was observed significantly slower than the hash index. However, the R-tree clustering model created

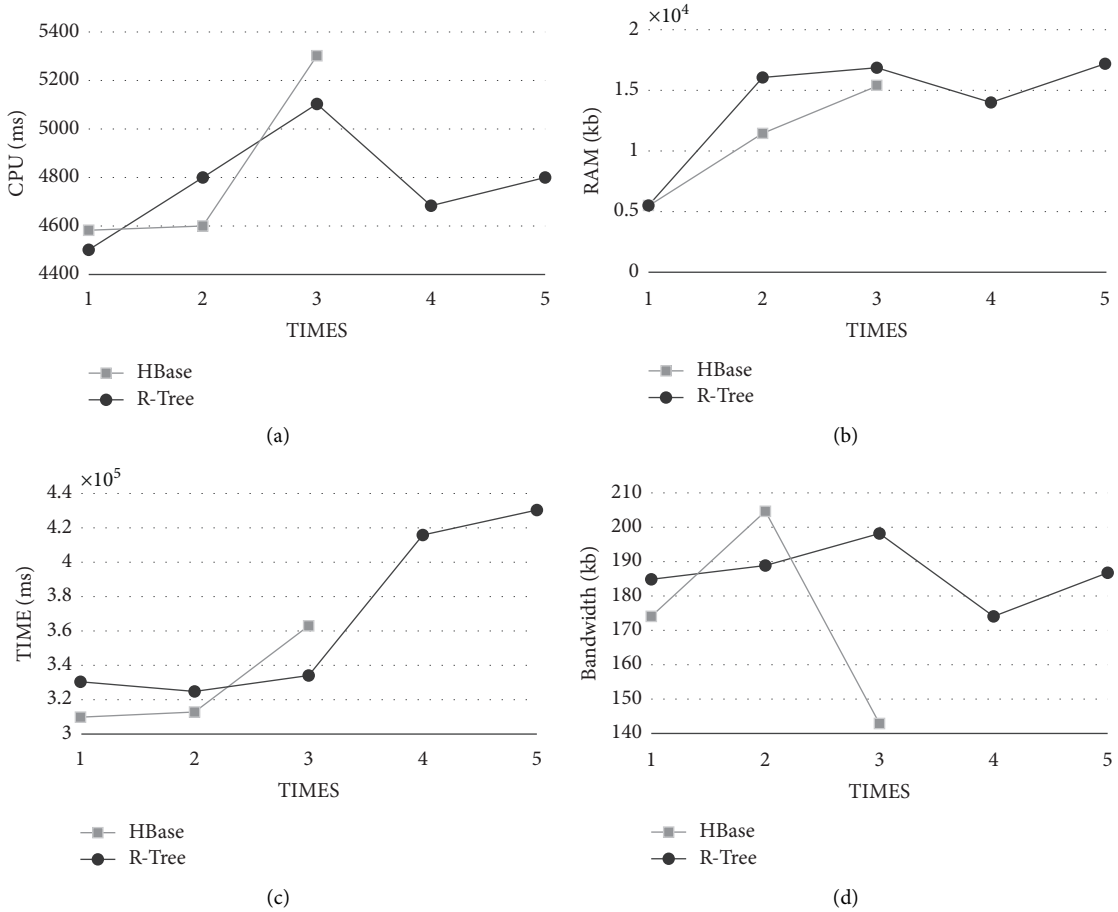

Figure 5: The R-tree index effect in terms of CPU running time, RAM usage, system running time, and bandwidth ($\alpha = 1, \beta = 1$). (a) CPU running time. (b) Ram occupancy. (c) System runing time. (d) Bandwidth.

and proposed in this paper can effectively reduce the coverage and overlap amongst MBRs. Moreover, it can make the generated tree dense and have less multipath retrieval, and improve the retrieval efficiency. After the system runs for a period of time, the retrieval efficiency of the R-tree clustering model is significantly higher than that of the hash index [25].

When the quantity of tasks is huge, then the performance and quality of the proposed system for using the R-tree clustering model is significantly better than that of the hashing method (as shown in Figure 4), specifically when the communication link of the network is full of activity (as shown in Figure 5). When the network is busy, the system that deploys the hash index crashes and cannot run after several simulations, resulting in the failure of subsequent simulations. However, the system that employs the R-tree clustering model can still perform subsequent operations. This is because of R-tree, the more times it is searched, the more data it contains and the more search paths it has, thus affecting the retrieval response time. When the amount of data is enormous, it will have a negative effect on the system performance after operating for a while, since the creation and maintenance of hash tables places a heavy demand on the computational performance of the computer.

3.4. Computational Complexity. The time complexity of R-tree built by implementing the proposed techniques and algorithms is $O(n^k \times t)$, where k represents the total amount of clusters, n represents the total amount of data, and t characterizes the total number of iterations. Furthermore, the hash index method computational complexity is $O(1)$. Therefore, at the beginning of system operation, the R-tree clustering model index was observed significantly slower than the hash index. However, the R-tree clustering model created and proposed in this paper can effectively reduce the coverage and overlap amongst MBRs. The complexity of the proposed method increases along with increase in the number of clusters, amount of data, and the index system.

4. Conclusions and Future Work

Although, data retrieval is frequently a significant bottleneck, the Internet of things and big data technologies have considerable potentials for applications in the domain of material management, hospital personnel, and the development of medical technologies. In order to upsurge the usefulness of applications and the advantages of medical information technology, data exchange, and efficient retrieval must be realized. In this paper, a new retrieval technique grounded on the R-tree clustering model index is

suggested for medical data with various types and large amounts of data. Firstly, the medical data is clustered and divided by the dynamic determination of cluster center (DCC) algorithm. By selecting the optimized cluster center, the data in the same subspace are clustered under the same subtree. Furthermore, a layer-by-layer effective R-tree method is constructed from root nodes to leaf nodes. The medical data is retrieved through the R-tree clustering model index to increase the efficiency of the information retrieval system. Then, the experiments were carried out in the area of the hospital information system. As a final step, the system quality of service and performance was assessed under various network tasks and states. Through comparing with the system performance after deploying the hash index, it was empirically proved and validated that the proposed method significantly improves the information retrieval efficiency of the system.

From our experimental outcomes, it can be seen that the index structure of the R-tree clustering model is constructed, layer-by-layer, from top to bottom through dynamically determining the clustering center using the algorithm proposed in this paper. We noted that the proposed system can effectively improve the information retrieval efficiency of medical information by deploying the R-tree clustering model in the medical information system. However, due to the large amount of computation when clustering data with the clustering algorithm, the CPU utilization is too high when the retrieval task is small. Therefore, future research also needs to select the appropriate model according to the actual data characteristics, and use the data mining algorithm to study the more universally applicable retrieval methods. Furthermore, the recent advancement of the edge-cloud servers setting can also be used to improve the CPU running time of the proposed algorithm. In that context, the database should be placed on a cloud server, while the information retrieval module related to the clustering will be implemented over the cloud. On the edge, each hospital will run the patient monitoring system that will optimize the patient statistics based on the data stored on the cloud. In ongoing work, we plan to implement the proposed R-tree clustering information retrieval method over the edge cloud. Finally, the deep learning network can also be integrated to further optimize the parameters and data.

Data Availability

The data used to support the findings of this study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. T. Azar and A. E. Hassanien, "Dimensionality reduction of medical big data using neural-fuzzy classifier," *Soft Computing*, vol. 19, no. 4, pp. 1115–1127, 2015.

[2] K. Jee and G. H. Kim, "Potentiality of big data in the medical sector: focus on how to reshape the healthcare system," *Healthcare Informatics Research*, vol. 19, no. 2, pp. 79–85, 2013.

[3] M. Gao, "Design of medical big data file retrieval system based on distributed NoSQL," *Electronic design engineering*, vol. 28, no. 21, pp. 24–28, 2020.

[4] Y. Wang, K. Bao, T. Huang, and Shaoxin, "Research on the application of big data analysis based on power Bi tools in multidimensional data analysis of medical devices," *China Medical equipment*, vol. 17, no. 5, pp. 169–173, 2020.

[5] Chengzhiqun, C. Zhang, and G. Han, "Research on data retrieval technology based on Solr," *Journal of Hangzhou University of Electronic Science and Technology (NATURAL SCIENCE EDITION)*, vol. 37, no. 1, pp. 11–15, 2017.

[6] Tengzhijun and Y. Zhang, "Improved D-S evidence theory map matching algorithm under the background of intersection," *Chinese Journal of inertial technology*, vol. 28, no. 3, pp. 380–385, 2020.

[7] G. Wanghuifang and P. Gao, "A minimum polygon construction algorithm based on quadtree index," *Geospatial information*, vol. 18, no. 11, pp. 118–120, 2020.

[8] J. Sunlele, "Research on the construction and application of distributed r∗ tree index based on NoSQL," *Geography and geographic information science*, vol. 37, no. 5, pp. 9–15, 2021.

[9] Q. Houhaiyao, C. Ying, H. Zhang, and Y. Zhao, "Spatio-temporal query algorithm based on hierarchical index of hilbert-r tree," *Computer Applications*, vol. 38, no. 10, pp. 2869–2874, 2018.

[10] E. Al-Nsour, A. Sleit, and M. Alshraideh, "SOLD: a node-Splitting algorithm for R-tree based on objects' locations distribution," *Journal of Information Science*, vol. 45, no. 2, pp. 169–195, 2019.

[11] J. W. Li, K. Wang, L. Huang, and G. Wang, "Skyline query algorithm based on R-tree index in MapReduce model," *Journal of Jilin University (Science Edition)*, vol. 54, no. 4, pp. 833–838, 2016.

[12] N. B. Huyupu, "Construction of R-tree spatial index based on dynamic k-value clustering algorithm," *Computer science and exploration*, no. 2, pp. 173–181, 2016.

[13] X. Chen, "Fast synchronization clustering algorithms based on spatial index structures," *Expert Systems with Applications*, vol. 94, no. 1, pp. 276–290, 2018.

[14] X. Wang, W. Meng, and M. Zhang, "A novel information retrieval method based on R-tree index for smart hospital information system," *International Journal of Advanced Computer Research*, vol. 9, no. 42, pp. 133–145, 2019.

[15] B. Kao, L.F. K. F. Sau Dan, D. W. Cheung, and H. Wai-Shing, "Clustering uncertain data using voronoi diagrams and r-tree index," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 9, pp. 1219–1233, 2010.

[16] Z. Zhao, Z. Jian, G. S. Gaba, R. Alroobaea, M. Masud, and S. Rubaiee, "An improved association rule mining algorithm for large data," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 750–762, 2021.

[17] Y. Xie, "Research on application of ultrasound medical imaging technology in big data mining of regional medical imaging," *Journal of Medical Imaging and Health Informatics*, vol. 11, no. 3, pp. 930–937, 2021.

[18] X. Dengbintao, "Differential evolution k-center clustering algorithm based on dynamic twin population," *Computer and Modernization*, no. 7, pp. 54–59, 2021.

[19] X. Zang, P. Hao, X. Gao, B. Yao, and G. Chen, "Qdr-tree: an efficient index scheme for complex spatial keyword query," in *Proceedings of the International Conference on Database and*

Expert Systems Applications, Bratislava, Slovakia, September 2018.

[20] S. Avasthi, R. Chauhan, and D. P. Acharjya, "Techniques, applications, and issues in mining large-scale text databases," *Advances in Information Communication Technology and Computing*, pp. 385–396, Springer, Singapore, 2021.

[21] S. Abbas, Q. Nasir, D. Nouichi et al., "Improving security of the Internet of Things via RF fingerprinting based device identification system," *Neural Computing & Applications*, vol. 33, pp. 14753–14769, 2021.

[22] T. S. D. Student, "Turning healthcare challenges into big data opportunities: a use-case review across the pharmaceutical development lifecycle," *Bulletin of the American Society for Information Science and Technology*, vol. 39, no. 5, pp. 34–40, 2013.

[23] B. Kan, W. Zhu, G. Liu, X. Chen, D. Shi, and W Yu, "Topology modeling and analysis of a power grid network using a graph database," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, p. 1355, 2017.

[24] S. Mustafa, B. Nazir, A. Hayat, R. Khan, and S. A. Madani, "Resource management in cloud computing: taxonomy, prospects, and challenges," *Computers & Electrical Engineering*, vol. 47, pp. 186–203, 2015.

[25] Y.-H. Zhang, C. Wen, M. Zhang, K. Xie, and J. B. He, "Fast 3D visualization of massive geological data based on clustering index fusion," *IEEE Access*, vol. 10, no. 2022, pp. 28821–28831.