Hindawi
Complexity
Volume 2021, Article ID 6211475, 13 pages
https://doi.org/10.1155/2021/6211475

Research Article

Privacy-Preserving Efficient Data Retrieval in IoMT Based on Low-Cost Fog Computing

Na Wang[1], **Yuanyuan Cai**[2], **Junsong Fu**[3] **and Jie Xu**[3]

[1]*School of Cyber Science and Technology, Beihang University, Beijing, China*
[2]*National Engineering Laboratory for Agri-Product Quality Traceability, Beijing Technology and Business University, Beijing, China*
[3]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China*

Correspondence should be addressed to Yuanyuan Cai; caiyuanyuan@btbu.edu.cn

Received 9 May 2021; Accepted 12 June 2021; Published 22 June 2021

Academic Editor: Fei Xiong

The rapid development of Internet of Medical Things (IoMT) is remarkable. However, IoMT faces many problems including privacy disclosure, long delay of service orders, low retrieval efficiency of medical data, and high energy cost of fog computing. For these, this paper proposes a data privacy protection and efficient retrieval scheme for IoMT based on low-cost fog computing. First, a fog computing system is located between a cloud server and medical workers, for processing data retrieval requests of medical workers and orders for controlling medical devices. Simultaneously, it preprocesses physiological data of patients uploaded by IoMT, collates them into various data sets, and transmits them to medical institutions in this way. It makes the entire execution process of low latency and efficient. Second, multidimensional physiological data are of great value, and we use ciphertext retrieval to protect privacy of patient data in this paper. In addition, this paper uses range tree to build an index for storing physiological data vectors, and meanwhile a range retrieval method is also proposed to improve data search efficiency. Finally, bat algorithm (BA) is designed to allocate cost on a fog server group for significant energy cost reduction. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.

1. Introduction

Until now, more than 150 million people worldwide have been suffering from COVID-19, resulting in more than 3 million deaths. Mortality rates of COVID-19 are different across the world. In areas of poverty and lack of medical resources, more death occurrences because of the exhaustion of medical resources pose a significant threat to health and safety of healthcare workers. To protect medical workers, it is particularly important for them to use IoMT to manage, diagnose, and provide treatment advices to patients remotely. The medical devices or health detection sensors are connected with a computer which is then connected to a cloud sever via a network. Physiological data of patients such as blood sugar, blood pressure, heart rate, and neutrophils are uploaded to the computer and cloud server for storage. Retrieving relevant data on a cloud sever enables healthcare workers to remotely interact with patients for pathological analysis and effective diagnosis.

At present, the industrial standards of Internet of Medical Things have been improved, and technological innovations have been emerging fast. These have made personalized medicine increasingly popular. Medical workers are able to analyze patients' physiological data remotely in their own environment. The development of this remote approach depends on advancement of sensing, monitoring, and data processing technologies. This paper uses the Internet of Medical Things to deal with the challenges. Under the premise of protecting privacy of patients' data, the patients' data are to be retrieved and analyzed, through remote monitoring of patients, diagnosis, analysis, and provision of telemedicine suggestions. Apparently, the Internet of Medical Things involves a large number of detection devices or sensors. They are widely distributed and create a large amount of data.

In order to manage these IoMT devices and organize data more efficiently, some medical institutions and medical workers outsource large-scale data to cloud servers [1–4]. However, a "distance" between medical workers and cloud servers likely leads to a high delay for medical workers to diagnose and analyze diseases. At the same time, with development of intelligent medical treatment, medical workers need immediate and efficient retrieval of patients' data. This is reasonable considering that certain diseases such as heart disease and stroke have a rapid onset, requiring immediate and rapid diagnosis by medical workers. It is difficult for existing solutions to meet all the above actual needs.

In order to reduce network latency, He et al. [5] use fog computing to efficiently utilize cloud resources in the network, which brings sustainability to data processing in IoMT. Fog computing was first proposed by Professor Stolfo of Columbia University. Cisco redefined fog computing and proposed an application method, which made fog computing famous. The fog computing system is located between the cloud service and the medical worker, with features of low latency, high computing efficiency, and decentralization. The architecture of fog computing system is close to the edge of network and presents the characteristics of distribution. Because fog computing is "closer" to healthcare workers than to cloud services, fog computing preprocesses requests sent by medical workers and then uploads them to cloud servers for retrieval. Therefore, efficiency and latency are both considered. It is better than retrieval based only on cloud services. Therefore, fog computing based IoMT emerges [6–9].

Because cloud server is "curious and honest" [10], traditional retrieval is generally based on plaintext retrieval. However, directly uploading patients' physiological data or index to a cloud server leads to privacy disclosure of patients. If the encrypted data are uploaded to a cloud server and the retrieval is not processed based on ciphertext, then medical workers need to decrypt the ciphertext before retrieving it. Development of ciphertext retrieval based on multiple keywords improves efficiency and accuracy of retrieval [11, 12]. It also ensures the privacy and security of medical workers, promoting retrieval services based on the IoMT to a certain extent.

Different from existing ciphertext retrieval schemes in the cloud computing, this paper constructs a ciphertext retrieval scheme based on the fog computing system. Our scheme adopts a vector space model. Each physiological data is regarded as a point in a high-dimensional space, and a corresponding data vector is generated by a medical institution. At the same time, data vectors are preprocessed to construct a range tree index, so as to improve the efficiency of retrieval. Finally, the data set and the range tree index are encrypted and sent to the cloud server for storage. After the medical workers are authorized by the medical institution, the query vector is uploaded to the fog computing system. The fog computing system is "safe and reliable" and it shares the security key with medical workers. It is also responsible for encrypting the query vector sent by medical workers. The fog computing system then sends a query trapdoor and a retrieval range vector to the cloud server. By range retrieval on cloud server, ciphertext data are returned to the fog computing system. Finally, the fog computing server sends decrypted data to medical workers for disease diagnosis and analysis.

This paper adopts a two-layer fog computing architecture. The first layer is composed of high-end intelligent devices, such as routers, switches, and gateways, which are used to collect data from IoMT devices. The second layer is a fog computing server group made up of multiple high-performance servers that process data and execute orders sent by the healthcare workers and the IoMT. However, with high efficiency and low delay, management of resources becomes challenging in the face of frequent command requests from medical workers and IoMT. In addition, the fog computing system has a risk of high energy cost while executing orders. The energy cost mainly comes from the execution environment, refrigeration equipment, and power regulation. For fog computing system, high energy cost is a key issue. Deploying the system costs a lot of energy. The main source of energy is fossil fuel, which potentially causes a serious greenhouse effect. Therefore, optimizing orders configuration of server improves efficiency of fog computing system.

The main contributions of this paper are summarized as follows:

(1) This paper proposes an IoMT retrieval service based on fog computing, which enables medical workers to efficiently obtain IoMT data. The fog computing system makes the data transmission of IoMT devices and the retrieval request of medical workers efficient with low latency.

(2) A range tree is adopted to construct the index of data, which significantly improves retrieval efficiency. In order to prevent the privacy leakage of medical workers during retrieval, a ciphertext retrieval scheme based on multibody feature data was proposed. At the same time, the scheme also improves retrieval accuracy.

(3) Within the fog computing system, a scheduling algorithm is designed to reduce energy cost. This algorithm can not only ensure the high efficiency of retrieval in our scheme, but also significantly reduce energy cost of the system.

(4) In-depth analysis of efficiency and accuracy of the retrieval data of the scheme is provided in this paper. Moreover, we conduct simulation on the actual data set. Simulation results show that the proposed scheme achieves high efficiency and accuracy, while significantly reducing energy cost.

The rest of this paper is organized as follows: in Section 2, this paper introduces relevant research and illustrates innovation of this paper's scheme. In Section 3, the architecture and system model of the IoMT are described, and functions of their parts are introduced. Finally, the threat model and symbol description are introduced. In Section 4, the algorithm based on ciphertext retrieval is introduced in detail. The construction method of range tree index and

functions of the application layer of IoMT by using range retrieval are also introduced. Section 5 presents bat algorithm (BA) which allocates resources for orders on the fog computing system. In Section 6, security, retrieval efficiency, and energy cost of the scheme are simulated and analyzed, and the rationality and effectiveness of the scheme are proved. Finally, the article is summarized in Section 7.

2. Related Work

Previous IoMT research focused on applications of physical testing equipment. For example, Hijazi et al. [13] employed IoMT in detection of heart sound, through signal processing and auxiliary diagnosis, but they did not mention how to retrieve data of patients, and data privacy protection. Redlarski et al. [14] proposed a machine learning algorithm to filter and analyze patient data uploaded by the IoMT to achieve disease warning. However, this scheme, without fog computing system, provides privacy protection with low efficiency and long delay. Rizk et al. [15] elaborated on privacy protection, but they did not propose how to retrieve data. Mishra et al. [16] proposed a fog computing service scheduling algorithm and discussed how to reduce energy cost of fog computing system, but it was not combined with a retrieval scheme of the IoMT.

Previous studies on patients' data analysis and retrieval are commonly based on plaintext, which violates patients' privacy rights. Therefore, the current research direction turns to ciphertext retrieval. At present, research studies on ciphertext retrieval are mainly based on cloud services, and there are few ciphertext retrieval studies based on fog computing system and even fewer ciphertext retrieval studies on IoMT. Cao et al. [17] first proposed a privacy-preserving multikeyword ranked search over encrypted data in cloud computing (MRSE). In this scheme, the secure KNN algorithm is used for retrieval, and a reversible matrix and random split indicator are used to encrypt data vectors and retrieval request vectors, so as to realize ciphertext retrieval. However, the index is not processed effectively in this scheme, which results in low retrieval efficiency. Fu et al. [12] realized personalized search by encrypting and outsourcing data. In the scheme, the index is partitioned into blocks, and then the index tree is constructed by blocks. In addition, the interest model of medical workers is added to improve the retrieval efficiency. However, their retrieval accuracy is low because of the truncated index tree. Xia et al. [10] proposed a secure dynamic multikeyword sorting search scheme based on cloud data. The scheme is designed based on a clustering algorithm, which clusters the data first and then builds a tree to improve retrieval efficiency. However, due to different values of the clustering algorithm, the clustering and retrieval results are different, resulting in inaccurate retrieval.

In applications of IoMT, previous studies [5, 15, 16, 18] likely involved no ciphertext retrieval and no fog computing. In this paper, we study and propose relevant schemes to protect the privacy of patients, improve the retrieval efficiency of medical workers, and reduce the delay of data transmission. At the same time, this paper adopts the scheduling algorithm of fog computing system, which reduces the energy cost of the whole system. In terms of retrieving encrypted data, Cao et al. [17] proposed the concept of multikeyword ciphertext retrieval, but there was no efficient index processing, resulting in low retrieval efficiency. The scheme proposed by Fu et al. [12] divided the index into blocks, but the blocks are simple and the structure is complex, which results in low retrieval efficiency. Xia et al.'s scheme [10] uses clustering algorithm, but this scheme leads to low search accuracy. In this paper, we improve the efficiency and accuracy of retrieval.

3. Problem Description

3.1. Architecture of the Internet of Medical Things. Figure 1 shows the architecture of the Internet of Medical Things used in this paper. The architecture in the figure is divided into three layers: perceptual layer, network layer, and application layer. The following is an introduction to functions of each layer:

(1) Perceptual layer is composed of medical sensors, identification QR codes, transmission paths, and gateway. Patients' blood pressure, blood sugar, neutrophils, and other data are collected by sensors, and they are identified by two-dimensional code. At last, the data are uploaded to the network layer for further processing from WIFI and other transmission channels.

(2) Network layer is composed of fog computing system and public cloud server. The fog computing system is responsible for collecting the patient's physiological data. First, the cloud server normalizes them and then generates a data vector F_i in which each value represents a physical feature of the patient. Finally, the cloud computing system sends the patient's data vector set $\mathcal{F}$ to medical institutions for encryption before uploading them to the cloud server, which aims to protect the patient's privacy. The cloud server is responsible for collecting and storing the encrypted physiological data of patients and the encrypted index. Medical workers retrieve the physiological data of patients through the range and use them for medical analysis in the cloud.

(3) Application layer is responsible for realizing the specific functions of the IoMT. This paper mainly proposes three specific functions.

First, the application layer manages patient information, by collecting physiological data of patients and establishing a patient information database to centrally manage patients.

Second, medical workers retrieve suspected patients' information from the cloud by inputting range of physiological data of a disease's characteristics. For example, the body temperature of COVID-19 patients is severally above 37.4°C, with creatinine value of more than 100 μmol/L and interleukin-6 of 150 pg/mL. Previous retrieval schemes, such as KNN algorithms, retrieve the most relevant first k value. However, normal physiological indexes of a human body are

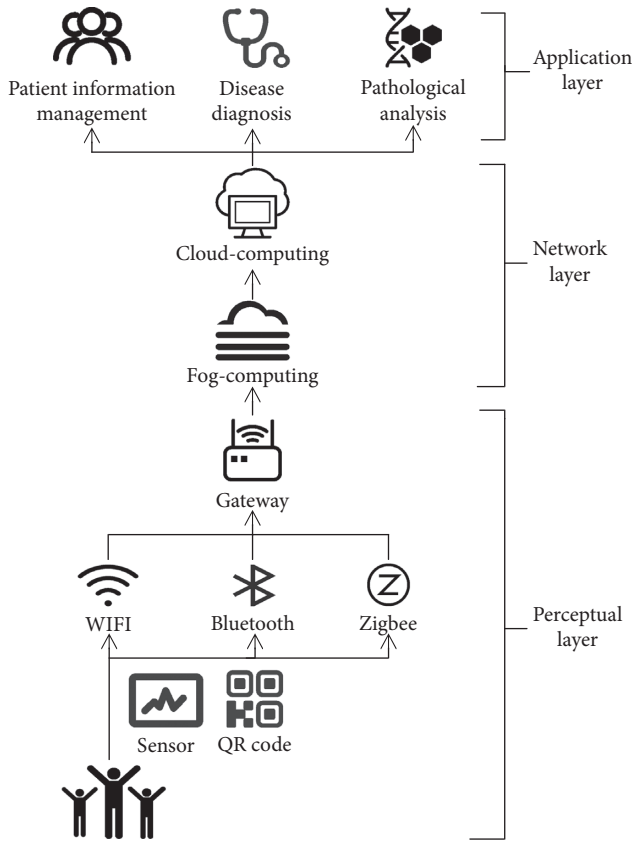

Figure 1: Architecture of the Internet of Medical Things (IoMT).

more likely a range rather than a specific value. Therefore, this paper uses range search, which requires the medical workers to input an appropriate range to preliminarily screen out suspected patients, and then conduct further detailed screening. This enables remote diagnosis and treatment, with protection for health workers from infection.

Third, the application layer also implements pathological analysis. For example, in the case of an unknown disease, healthcare workers want to know the physical characteristics of the patient with the disease. The medical workers identify the patient with the disease by "coloring" the patient and searching for the range of certain physiological indicators. If there is a cluster of color-coded patients within a certain range, this indicates that the pathological characteristics of the disease are related to the physiological index.

3.2. System Model. Figure 2 is the system model of the scheme designed in this paper. The following is a functional introduction of each part of the model:

Medical institutions

The medical institution encrypts the data set $\mathcal{F}$ transmitted by the fog computing system as $\mathcal{C}$, then constructs a range tree index according to $\mathcal{F}$, and encrypts it as $\mathcal{I}$. The medical institution then delivers the encrypted data set $\mathcal{C}$ to the cloud server and the encrypted index $\mathcal{I}$ to the fog computing system. In addition, medical institutions transfer the shared keys to trusted medical workers.

Medical workers

When the medical worker obtains the shared key of the medical institution, the medical worker transfers the retrieval range, key k, and query vector Q to the fog computing system according to their own retrieval demands. The fog computing system returns required data to healthcare workers after it completes the retrieval on the cloud server. In addition, medical workers issue orders to fog computing systems to remotely control IoMT devices.

IoMT device or sensor

An IoMT device or sensor collects physiological data from a patient and uploads it to a fog computing system. The fog computing system then preprocesses the data and transmits it to a medical institution. In addition, medical workers remotely control IoMT devices through fog computing system, such as adding or deleting devices and adjusting patients' intelligent medical devices.

Fog computing system

The fog computing system is "safe and reliable" and it is responsible for receiving data uploaded by IoMT devices or sensors. The fog computing system then collates the data and sends it to the medical institution to update the data set. In addition, the fog computing system receives the retrieval range and query vector Q sent by the medical worker. Because the fog computing system is safe and reliable, it shares the key with the medical workers. The system encrypts the query vector and generates trapdoor according to the key shared by the medical workers. At the same time, the fog computing system generates the retrieval range vector R according to the retrieval range and then uploads it and query trapdoor T_Q to the cloud server for retrieval. It is also responsible for receiving the results returned by the cloud server. Finally, the fog computing system uses the key to decrypt the data set and send it to the medical workers, minimizing the workload of the medical workers.

Public cloud server

The public cloud server is responsible for storing the encrypted data set $\mathcal{F}$ and encrypted range tree index $\mathcal{I}$ uploaded by the medical institution. In addition, the cloud server receives query trap T_Q and retrieval range vector R according to the retrieval range. Then, the cloud server uploads the retrieval range and query trapdoor T_Q to the cloud server for retrieval, and it is also responsible for receiving the results returned by the cloud server.

3.3. Threat Model. In this paper, cloud servers are "curious and honest" and follow the orders of healthcare workers. At the same time, they "curiously" analyze the data retrieved by

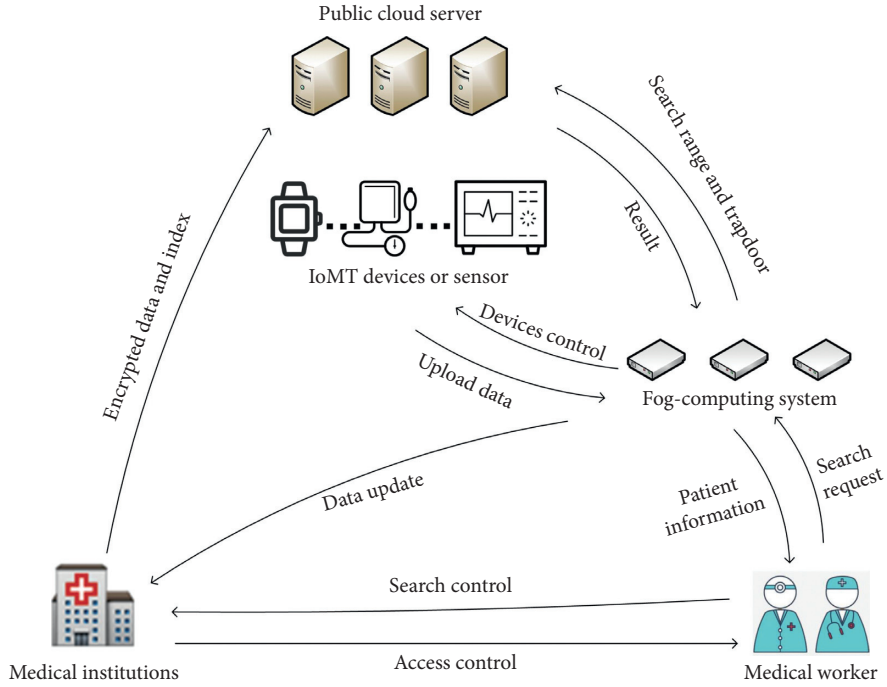

Figure 2: System model.

medical workers, which eventually leads to disclosure of the privacy of medical workers and data. Based on the available information of cloud server, this paper establishes two threat models:

Known ciphertext model

The cloud server obtains the encrypted data set $\mathscr{C}$ and encrypted index $\mathscr{I}$ sent from the medical institution. They know nothing else and only attack ciphertext to gain privacy.

Known background knowledge model

Under the condition of known background knowledge model, the cloud server analyzes the retrieval process of medical workers. Then, it tries to find the connection between ciphertext and index by statistical information of medical workers' search records and get the connection between keyword frequency and physical characteristics data.

3.4. Symbol Description. For convenience, some notations are first defined as follows:

(i) $\mathscr{F}$: the plaintext data set $\mathscr{F} = \{F_1, F_2, \ldots, F_m\}$ contains m patient physiological data.

(ii) $\mathscr{C}$: data set $\mathscr{F}$ encryption form $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$, a total of m encrypted data.

(iii) F_i: plaintext data vector $F_i = \{f_{i1}, f_{i2}, \ldots, f_{in}\}$, where each dimension is the normalized value of the patient's body index data.

(iv) $\mathscr{I}$: the range tree index is built by the data vector $F_i\,(0 < i \le m)$, and then the range tree index is encrypted to get $\mathscr{I}$.

(v) Q: if a dimension of query vector Q is 1, it represents the medical worker to retrieve the physiological index data. If the value is 0, it indicates that the physiological indicator data was not retrieved.

(vi) T_Q: the query vector Q is encrypted to generate trapdoor T_Q.

(vii) R: the retrieval range of the patient's physiological index data was generated by the fog computing system according to the retrieval range sent by the medical workers to generate the retrieval range vector $R = \{e_{w_1}, e_{w_2}, \ldots, e_{w_n}\}$.

(viii) VM: the fog computing system VM is composed of m high-performance servers, $VM = \{V_1, V_2, \ldots, V_m\}$.

(ix) S: the list of orders sent to the fog computing system by an IoMT device or medical worker consists of n orders. $S = \{s_1, s_2, \ldots, s_n\}$.

4. Secure Storage and Retrieval of Medical Data

4.1. Framework of Ciphertext Retrieval. The IoMT devices upload data and send it to the fog computing system, which is responsible for collating these data and generating data sets. Then, the fog computing system sends the data sets to the medical institution to update them. The medical institution uses random numbers; a pair of reversible matrices M_1^T, M_2^T; master key sk; and key k to encrypt the data vector set $\mathscr{F}$ and the range tree index constructed from the data vector. Then, it sends the data set and index to the cloud server. Medical workers submit retrieval request to fog service system. The fog server generates trapdoor by encrypting query vector and uploads it to the cloud along

with the range search vector R. At last, the results are returned to the medical worker. The scheme framework designed in this paper mainly includes the following algorithms:

(i) Key generation $(1^{l(n)}) \longrightarrow (\mathrm{sk}, k)$: this step mainly generates a master key sk and key k to encrypt index and data, respectively.

(ii) Constructing an encrypted index $(\mathscr{F}, \mathrm{sk}) \longrightarrow \mathscr{I}$: the medical institution first uses the data vector set to construct a range tree index and then encrypts it with a secure algorithm to get index $\mathscr{I}$.

(iii) Data encryption $(\mathscr{F}, k) \longrightarrow \mathscr{C}$: the medical institution encrypts the data set using a symmetric encryption algorithm to obtain the ciphertext set $\mathscr{C}$.

(iv) Trapdoor generation $(Q, \mathrm{sk}) \longrightarrow T_Q$: the fog computing system generates a query trapdoor T_Q based on the query vector sent by the medical worker and the key shared by the medical worker.

(v) Retrieval $(T_Q, \mathscr{I}, R) \longrightarrow C_Q$: in this process, the cloud server receives the query trapdoor T_Q and the retrieval range vector R from the fog computing system, and then it retrieves the corresponding ciphertext data C_Q in the range. Finally, it sends C_Q to the fog computing system.

(vi) Decryption $(C_Q, k) \longrightarrow F_Q$: the cloud server returns the retrieved encrypted data to the fog computing system. The fog computing system decrypts the data according to the key shared with the medical workers and sends it to the medical workers.

The following is a detailed description of the main algorithms in the scheme architecture of this paper:

Key generation $(1^{l(n)})$

The medical institution generates an $(n+u+1)$ dimensional split indicator vector H, where each element is a random 1 or 0. At the same time, the medical institution generates two $(n+u+1)$-dimensional reversible matrices M_1^T and M_2^T, where each element is a random integer. In this paper, the master key $\mathrm{sk} = \{H, M_1^T, M_2^T\}$. In addition, the medical institution selects an n-bit pseudosequence to generate the data encryption key k.

Building an encrypted index $(\mathscr{F}, \mathrm{sk})$

Medical institutions construct range tree index according to $\mathscr{F}$ and then extend the n-dimensional vector A of each node in the range index tree to the dimension vector $\overline{A}$ of $(n+u+1)$, in which the dimensions from $n+1$-th to $n+u$-th are set as random integers, and the dimension $n+u+1$-th is set as 1. Then, the medical institution uses the split indicator vector H to split $\overline{A}$. If $H[i] = 0$, then $\overline{A}'[i] = \overline{A}''[i] = \overline{A}[i]$; If $H[i] = 1$, then $\overline{A}'[i]$ is g' random number, $\overline{A}''[i] = \overline{A}[i] - g'$. Finally, the medical institution obtains the encrypted index $\mathscr{I} = \{M_1^T\overline{A}', M_2^T\overline{A}''\}$ and sends it to the cloud server.

Data encryption $(\mathscr{F}, k)$

Symmetric encryption algorithm (for example, AES encryption) is adopted in medical institutions [17] to encrypt the plaintext data set $\mathscr{F}$, and the encrypted ciphertext set $\mathscr{C}$ is outsourced to the cloud server.

Generating (Q, sk) by trap gate

The healthcare worker generates the query vector Q and sends it to the fog computing system. Similarly, the fog computing system first extends the n-dimensional query vector Q to the $(n+u+1)$ dimensional vector $\overline{Q}$. It randomly selects b values between the $n+1$-th dimension and the $n+u$-th dimension and set them to 1. It sets the remaining values to 0 and the value of the $n+u+1$-th dimension to a random number $t \in [0, 1]$. The fog computing system generates $\overline{Q} = (r \cdot Q, t)$ by multiplying the previous $(n+u)$ dimension vector by a random number r and then splits $\overline{Q}$ according to the split vector H. When $H[i] = 1$, $\overline{Q}'[i] = \overline{Q}''[i] = \overline{Q}[i]$. When $H[i] = 0$, $\overline{Q}'[i]$ is a random number g, $\overline{Q}''[i] = \overline{Q}[i] - g$. Finally, the fog computing system generates an encrypted retrieval trap $T_Q = \{M_1^{-1}\overline{Q}', M_2^{-1}\overline{Q}''\}$ and sends it to the cloud server.

Retrieving $(T_Q, \mathscr{I}, R)$

The fog computing system sends trap T_Q and range retrieval vector R to the cloud server, where $R[i]$ $(i = 1, 2, \ldots, n)$ represents the retrieval range of the physiological data of the patient by the medical worker. The cloud server retrieves the list of data required by medical workers in the range tree based on the range tree index $\mathscr{I}$, the query trap T_Q, and the dynamically updated retrieval range R. The retrieval process of range trees is described in detail in Section 5.2 of this paper. The physiological data are calculated as follows:

$$\begin{aligned}\text{Physiological value} &= T_Q \cdot I \\ &= \left\{M_1^{-1}\overline{Q}', M_2^{-1}\overline{Q}''\right\} \cdot \left\{M_1^T\overline{A}', M_2^T\overline{A}''\right\} \\ &= \overline{Q}' \cdot \overline{A}'' + \overline{Q}'' \cdot \overline{A}' \\ &= Q \cdot A.\end{aligned} \tag{1}$$

4.2. Structure of Range Tree Index. Range tree is an improvement of kd-tree. Although range tree needs more storage space than kd-tree, it has a significant improvement in retrieval efficiency. Because the cloud server has a large amount of storage space, it is not necessary to consider the storage space taken by the range tree. The scheme designed in this paper mainly considers efficiency and accuracy of medical workers' data query, so the range tree is adopted to construct the index. The construction process is as follows:

(1) For the data set $\mathscr{F}$ (data vector set $\mathscr{F} = \{F_1, F_2, \ldots, F_m\}$), this paper constructs a balanced binary search tree TR from bottom to top according to the first vital sign data of all data vectors, and the data vector is stored in the leaf node.

(2) In a subtree of a nonleaf node L_1 in the balanced binary search tree TR, the data vector set corresponding to all leaf nodes under this subtree forms a subset $\mathcal{F}'$ (namely, $\mathcal{F}' \subset \mathcal{F}$) of $\mathcal{F}$, which is called the regular subset corresponding to L_1 and denoted as $\mathcal{F}(L_1)$.

(3) The data vectors in the regular subset of nonleaf node L_1 are organized according to their second vital sign data to establish the second dimensional range tree and form a joint structure with the first dimensional range tree. Nonleaf node L_1 has a pointer to the root of the new tree $TR_1(L_1)$ and the regular subset of the leaf nodes under this node is the subset $\mathcal{F}''$ of the previous one-dimensional data vector set, namely, $\mathcal{F}'' \subset \mathcal{F}'$.

(4) After recursive (2) and (3) steps, an n-dimensional range tree is constructed, as shown in Figure 3.

4.3. Data Retrieval on the High-Dimensional Range Tree. The retrieval process of the scheme designed in this paper is elaborated as follows:

(1) After the cloud server receives the query trap T_Q, it conducts the range retrieval according to the first dimension of the retrieval range vector R from the range root node in the first dimension of index $\mathcal{I}$. If the corresponding value of the query trap gate T_Q is 1, then the range retrieval is carried out in the first dimensional range tree to find the required data vector (leaf node).

(2) Based on the set of leaf nodes found in the first dimension, this paper takes them as a regular subset and finds their lowest common ancestor nodes easily through middle order traversal. Then, from this lowest common ancestor node to the balanced binary search tree in the next dimension, the process excludes leaf nodes that are not included in the first dimension range search. Finally, apply the steps in (1) to find the required data set. If the query gate $T_Q[i] = 0$ (that is, the medical worker does not retrieve the physical characteristics data of the dimension), the i-th dimension is retrieved directly from the root node to the tree of the $i + 1$ dimension. In addition, in order to prevent too many retrieval times and a narrow retrieval range width d value and for other reasons, the datum is not searchable. Therefore, this paper presets a minimum returned data quantity k. If the number of leaf nodes retrieved in the balanced binary search tree TR of a dimension is less than k, then the backtracking algorithm is called to find the leaf node closest to the lower bound of the retrieval range until k data is retrieved. Then, the retrieval is stopped and k data is returned. This guarantees that at least k data are returned per retrieval.

(3) The cloud server recurses the above two steps, and the required data vector set is found after range retrieval. Then, according to the id of these data, the corresponding ciphertext is returned to the fog computing system. Finally, the fog computing system decrypts it and sends it to medical workers.

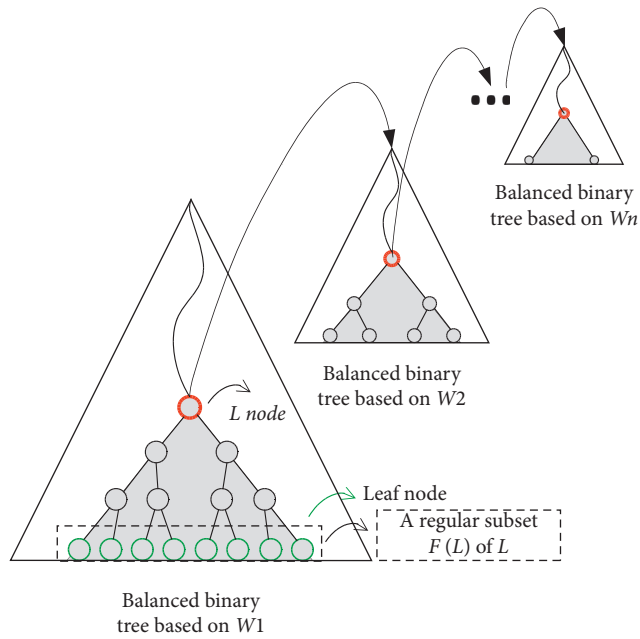

Figure 3: High-dimensional range tree construction.

The advantages of using range tree index and range retrieval are described below:

(1) Different from the retrieval scheme of Euler distance, this paper adopts the range retrieval scheme, which is more suitable for range tree index with higher efficiency. Meanwhile, in the medical field, most of the physiological data is a certain range rather than a specific value, and hence using the range retrieval is more suitable for the IoMT.

(2) After retrieving the balanced binary search tree of each dimension, a part of leaf nodes (i.e., the data vectors corresponding to leaf nodes) is excluded. It only needs to be retrieved in the balanced binary search tree composed of this regular subset of dimension. In this way, we significantly improve retrieval efficiency.

(3) As for range retrieval, currently most schemes adopt kd-tree. Simulation results show that the range tree retrieval efficiency is higher than kd-tree.

4.4. The Application Layer Function of IoMT Realized by Using Range Search. In this paper, the encrypted data is outsourced to the cloud server, which not only preserves the data of patients, but also protects the privacy of patients. In addition, the range tree index is constructed according to the patient's data vector set. In this way, the function of centralized management of patient information in the application layer of IoMT is realized, and the efficiency of retrieving patient information is improved.

For the disease diagnosis function of the IoMT application layer, medical workers are required to input the query vector and multidimensional physical signs of a disease data range in the fog computing system. The fog computing system normalizes the data range and uploads it to the cloud server for retrieval according to the retrieval vector generated on the trapdoor. Finally, information about suspected patients is returned to help medical workers diagnose the disease.

For example, the diagnosis of uremia has three important indicators: the glomerular filtration rate is less than 15 ml/min, the serum creatinine is greater than or equal to 707 umol/L, and the serum potassium is less than 3.5 mmol/L. By uploading the data range of these indicators, medical workers receive information of the suspected patient within this range. The proposed scheme also helps medical workers to carry out pathological analysis. For a disease with unknown pathology, medical institutions first "color" the patients with the disease. Medical workers select appropriate body index data for range retrieval. If a large number of color-coded patients appear in the search results within a certain range, it is determined that the disease has this physiological characteristic. This helps medical workers achieve function of pathological analysis.

4.5. Update of the Range Tree. IoMT devices generate data and upload them to fog computing systems, where the data are processed into a data set and transmitted to the medical institution for updating. Medical institutions encrypt the data and upload them to cloud servers.

4.5.1. Node Insertion. Medical institutions calculate the data vector based on the updated data and insert each dimension of the data vector into the balanced binary tree according to the bottom-up rule.

4.5.2. Node Deletion. Medical institutions delete nodes in each dimension of the range tree according to the deletion rules of balanced binary tree. The node deletion of the range tree is completed after recursion to the dimension.

5. Efficient Processing of Medical Data

5.1. The Order Assignment Problem of Fog Computing System. As shown in Figure 4, this paper adopts the two-layer fog computing architecture. The first layer is composed of intelligent devices, such as routers, switches, and gateways, which are used to collect data sent by IoMT devices.

The second layer is the fog computing server group VM composed of multiple high-performance servers, which is used to process the collected data and execute the orders sent by medical workers and IoMT devices, such as the retrieval request of medical workers and the data upload request of IoMT devices. Each server V_i has a unique ID_{v_i}, main memory, bandwidth, and storage. Each order S_i is calculated

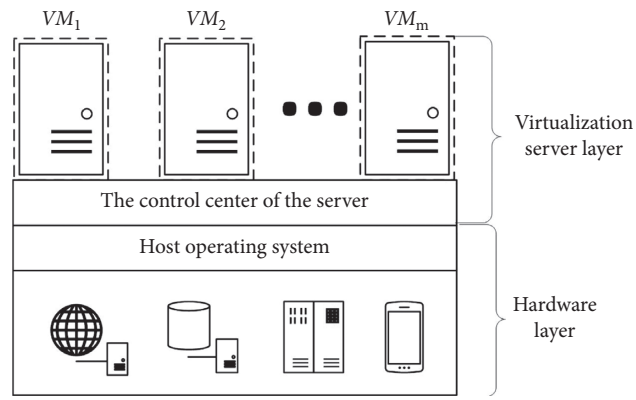

Figure 4: The architecture of fog computing system.

by the order ID_{s_i}, and the workload is calculated in millions of orders per second (MIPS). For each order S_i, only one fog server V_i is assigned, and service migration is not allowed until the order is completed. Therefore, for the order request list S to be assigned to the sever group VM, how to allocate the VM on the premise of meeting *sla* [16] without reducing QoS (quality of service) becomes a new challenge.

In this paper, the following assumptions are made for the fog computing system:

(1) Stipulate the expected running time of an order S_i on fog server V_j as ETC_{ij}. In addition, all order requests are independent and heterogeneous.

(2) The resource capabilities of all VM are heterogeneous.

(3) An order is only allowed to execute on one V_i.

Order list S is composed of n heterogeneous orders, and the order length L_i of each order S_i is represented by millions of orders (MI). These orders run in a server group VM with m servers, so you can build an $n \times m$ ETC matrix (see Figure 5).

In addition, each fog server V_j has a processing speed P_j, so in the ETC matrix, the order S_i in server V_j has $ETC_{ij} = (L_i/P_j X)$.

Suppose the energy consumed by the server V_j in the fog computing server group (Joule) is expressed as follows: First, the paper gives the energy consumed by V_j per unit length (J/MI):

$$E_{v_j} = \begin{cases} \beta_j, & \text{if } V_j \text{ is active,} \\ \alpha_j, & \text{if } V_j \text{ is not active.} \end{cases} \tag{2}$$

Let X_{ij} be the decision variable for whether to assign orders to a particular VM. If the order S_i is assigned to a server in the fog compute server group V_j, then the X_{ij} value is 1; otherwise, it is 0. Then, the total execution time of all orders of the j-th server V_j is

$$ET_j = \sum_{i=1}^{n} X_{ij} \times ETC_{ij}. \tag{3}$$

Makespan M is the maximum time value for all VM:

ETC matrix					
	VM_1	VM_2	VM_3	⋯	VM_m
S_1	ETC_{11}	ETC_{12}	ETC_{13}	⋯	ETC_{1m}
S_2	ETC_{12}	ETC_{22}	ETC_{23}	⋯	ETC_{2m}
⋮	⋮	⋮	⋮	⋮	⋮
S_n	ETC_{n1}	ETC_{n2}	ETC_{n3}	⋯	ETC_{nm}

Figure 5: ETC matrix.

$$M = \max\left(ET_j\right), \quad 1 \le j \le m. \tag{4}$$

The total energy cost of V_j is

$$E\left(V_j\right) = \left[ET_j \times \beta_j + \left(M - ET_j\right) \times \alpha_j\right] \times \text{MIPS}_j. \tag{5}$$

Thus, the energy consumed by the entire fog computing system can be obtained:

$$\varepsilon = \sum_{j=1}^{m} E\left(V_j\right). \tag{6}$$

The goal of this paper is to minimize the total cost ξ, which is a two-objective problem. On the one hand, it is hoped that the energy cost of fog service system is reduced as far as possible; on the other hand, it is hoped that the makespan M of fog service processing orders is shortened and hence the efficiency of fog computing system processing orders is improvable. In this paper, the total energy cost is expressed as

$$\text{Minimize}\, \xi = M \times \sigma + \varepsilon \times (1 - \sigma). \tag{7}$$

In this paper, a penalty value σ is set to represent the importance that medical workers attach to efficiency and energy cost. If healthcare workers are focusing on reducing energy use, they try to set a low σ value. If efficiency is important, set a high σ value.

From the above analysis, it is observed that assigning the order queue to the fog server group is an NP-hard problem [16] to minimize energy cost. In order to solve this problem, a service allocation algorithm is designed based on Algorithm 1.

5.2. Efficient Order Assignment Based on Bat Algorithm. This section discusses the order allocation problem of fog computing system. The retrieval request of medical workers and the upload request of IoMT device data both belong to the order. In the process of handling these orders, if the orders are not allocated in advance, the fog computing system consumes a lot of energy. In order to reduce energy consumption, bat algorithm (BA) [16] is adopted to assign orders to the fog computing system.

Suppose that, for a server group VM consisting of m servers, the order sequence S consisting of n orders enters into the server group VM. An order assignment vector is specified in the following as shown in Figure 6.

Algorithm 1 is a bionic computing technology that imitates bats to capture food and avoid obstacles by emitting ultrasonic pulses and acquiring echoes at night. It is stipulated that bats (i.e., commands) fly at speed v_i at position x_i, and each command has a fixed frequency $\Gamma_{\min}$, signal strength G, and signal pulse rate z. By iteratively calculating and updating the position and speed of the order, an optimal allocation scheme is finally converged.

In the first step of the algorithm, an order set is initialized, and the order assignment vector, velocity vector, signal strength G_0 of the initial order, and pulse rate z_0 of the initial order are specified. Then, the total energy cost is calculated according to the initial input service allocation vector, and the initial position is set as the optimal position. In steps 4–7 of the algorithm, the orders calculate the new position and speed in terms of frequency, speed, and previous position. In the 8–10 steps of the algorithm, if the generated random number is greater than the signal pulse rate of the order, the current optimal distribution vector is slightly disturbed to generate a new optimal distribution vector.

In steps 11–15 of the algorithm, if the generated random number is less than the signal strength of the instruction and the frequency is greater than the frequency of the previous iteration, then the change of the optimal distribution vector is accepted, the signal strength of the instruction is attenuated, and the signal pulse rate of the instruction is increased. Consequently, the algorithm is iterated to find the optimal solution.

6. Analysis and Simulation of Efficiency and Energy Cost

In this section, the retrieval efficiency and energy cost proposed by the scheme are analyzed theoretically and verified by simulation. In terms of retrieval efficiency, this paper adopts the common corpus on the network as the data set and uses C++ for simulation. In the energy cost simulation, this paper adopts MATLAB 2019a for simulation. The simulated hardware environment is Intel Core i5-8300H CPU, 8 GB memory, and Microsoft Window 10 operating system.

6.1. Safety Analysis. In this paper, the symmetric encryption algorithm AES is adopted to encrypt the data set $\mathscr{F}$ and generate ciphertext data set $\mathscr{C}$, which is uploaded to the public cloud server, effectively ensuring the security of the patient's physiological data itself. Then, reversible matrices M_1 and M_2 are generated randomly, and the index of range tree and query vector Q are encrypted to generate secure index $\mathscr{I}$ and query trap T_Q. Then, the fog computing server uploads them to the public cloud server. Since the space of the key matrix is infinite, each randomly generated key matrix has only one reversible matrix. The probability that the public cloud server correctly forges the key matrix to crack the security index $\mathscr{I}$ and query trap T_Q is almost 0, effectively ensuring the security of the information contained in the range tree index and query vector. Under the

Input: ETC matrix, VM processing speed, maximum number of iterations.
Output: the order assignment results for the VM (fog computing server group), the total cost ξ.
(1) Initialize random order set: a random order assignment vector x, velocity vector v and frequency of each order Γ, order signal strength G, order signal pulse rate z. In addition, I has a random variable $\eta \in (0, 1)$;
(2) Use formula (7) to calculate total energy cost ξ;
(3) x_{best} is the optimal location for each order;
(4) Update the position and speed of the order according to steps (5), (6) and (7);
(5) $\Gamma = \Gamma_{\min} + (\Gamma_{\max} - \Gamma_{\min})\eta$;
(6) $v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{\text{best}})\Gamma$;
(7) $x_i^t = x_i^{t-1} + v_i^t$;
(8) **if** $(\text{rand}_1(0, 1) > z_i)$ **then**
(9) $\quad x_i = x_{\text{best}} + 1$;
(10) **end if**
(11) **if** $\text{rand}_2(0, 1) < G_i$ and $\Gamma_i^t > \Gamma_i^{t-1}$ **then**
(12) $\quad x_i = x_i^t$;
(13) $\quad G_i^{t+1} = \text{rand}_3(0, 1) \cdot G_i^t$;
(14) $\quad z_i^{t+1} = z_i^0[1 - e^{-\text{rand}_4(0,1)\cdot t}]$;
(15) **end if**
(16) Repeat steps 4–15 for each order;
(17) Repeat steps 2–16 until a satisfactory convergence result or a maximum number of iterations is achieved.

Algorithm 1: BA.

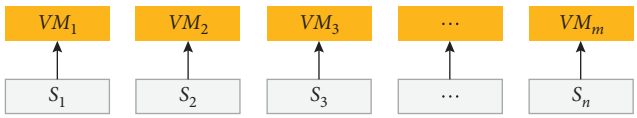

Figure 6: Order assignment vector.

known ciphertext model, the public cloud server only obtains ciphertext data set $\mathscr{C}$, security index $\mathscr{I}$, and query trapdoor T_Q. It is not allowed to obtain any useful data information unless it is ensured that the master key sk and key k are not artificially disclosed. In such cases, the scheme is safe.

In order to further prevent public cloud servers from mining and leaking data privacy information based on known background knowledge (that is, based on the internal connection between the security index and the query trapdoor), the mater key sk in our scheme is a random split indicator vector H, which is used to randomly split the expanded index vector $\overline{A}$ and query vector $\overline{Q}$. At the same time, random numbers g' and g are introduced in this random splitting process. Through such a series of operations, it is ensured that multiple regional tree indexes and query vectors are unrelated. Even if the medical workers repeat the same query operation for many times, the query trapdoor received by the public cloud server is different, which displays the unlinkability of the query trapdoor and effectively resists the statistical analysis attack. Consequently, our scheme is also safe for the known background knowledge model.

6.2. Retrieval Efficiency Evaluation. In the simulation of retrieval efficiency, the scheme in this paper is compared with the kd-tree scheme [19], the binary balanced tree scheme in literature [10], and the MRSE scheme in literature [17]. For comparison, the simulation of our scheme is to set the number of physical feature data records to be queried to 5. We analyze the time complexity of index construction and retrieval of range tree.

Theorem 1. *Given a data vector set consisting of n data vectors, the storage space occupied by the corresponding p-dimensional range tree is* $O(n\log_n^{p-1} n)$.

Proof. In each dimension of the range tree, each element in the leaf node set (i.e., data F_i) is stored only once at each depth. For a set of data vectors consisting of n data vectors, the height of constructing an equilibrium binary search tree is $\log n$. Since the storage space of each dimension in the range tree of each data vector is $O(\log n)$, the storage space required to construct a one-dimensional range tree of n data vectors is $O(n \log n)$. For a p-dimensional range tree, each dimension of storage space needs $O(n \log n)$, so the size of storage space required for a p-dimensional range tree is p times that of a one-dimensional range tree, that is, $O(n\log_n^{p-1} n)$. □

Theorem 2. *The p-dimensional range tree is constituted by n data vectors, and the query time complexity is* $O(\log^p n + k)$ *when it is retrieved in the range tree.*

Proof. To retrieve a p-dimensional range tree, the first dimension of the range tree should be retrieved (that is, a balanced binary search tree retrieval process), and the time complexity required is $O(\log n)$. Next, search the remaining $p - 1$ dimensional range tree to obtain the following time complexity relationship:

$$O_p(n) = O(\log n) + O(\log n) \times O_{p-1}(n), \tag{8}$$

where $O_p(n)$ represents the lookup time complexity of p-dimensional range tree and $O_{p-1}(n)$ represents the lookup

time of $p-1$-dimensional range tree. Then, according to (9), it is deduced as follows:

$$O_2(n) = O(\log\ 2\ n). \tag{9}$$

From the recursive formula (9) the time complexity of dimensional range tree retrieval is $O(\log^p n)$. In addition, this paper needs to record the obtained k data; the total time complexity is $O(\log^p n + k)$.

The retrieval efficiency of the scheme is mainly determined by the index structure of the data set and the score of calculated data similarity. For the range tree index in this paper, the number of queries for medical workers is set to 5 in the simulation. This paper first analyzes the effect of retrieval range width d on retrieval efficiency of medical workers. If the d value is large, the result is returned with low accuracy. If the d value is small, fewer nodes are retrieved in the range. In order to return k results, the scheme needs to invoke the backtracking algorithm many times, which reduces the retrieval efficiency. The relation between the retrieval time and the retrieval range width d of our scheme is shown in Figure 7. As seen from Figure 7, if the d value is larger, the retrieval time is shorter; if the d value is smaller, the retrieval time is longer. In addition, it is shown in Figure 7 that the return of k data also has a great influence on retrieval time. When the retrieval range width d value is greater than 0.03, the retrieval time tends to converge. In order to improve retrieval efficiency, it is necessary to select a high d value, but this leads to reduction of retrieval accuracy. Therefore, the most appropriate retrieval range width d value for medical workers is between 0.03 and 0.05 in order to balance the retrieval efficiency and accuracy under the scheme proposed. A d value 0.03 is selected for this simulation.

The retrieval efficiency of the proposed scheme is compared with that of other schemes. In the scheme proposed by Cao et al. [17], tree-type index structure was not used, and the retrieval time increased linearly with the increase of the amount of patient data, resulting in lower retrieval efficiency compared with other schemes. Although both Xia et al. [10] and the scheme in this paper adopted balanced binary search tree for retrieval, the multidimensional link structure of range tree was embedded in this paper, with high retrieval efficiency, as shown in Figure 8. Since both this study and [10] adopt a tree structure to construct indexes, the retrieval time is prolonged with the increase of data quantity in a logarithmic manner. Figure 8 shows that the retrieval time of the scheme is less than that in [10] and the retrieval efficiency is higher. Chen et al. [19] designed an index structure based on kd-tree, whose time complexity is close to $O(\sqrt{n}^{\,p} + k)$, which is higher than the time complexity $O(\log^p n + k)$ of our scheme. Our scheme has less retrieval time and higher retrieval efficiency than the scheme in [19], as shown in Figure 8. Both the scheme in this paper and the schemes in [10, 19] show a linear increase in the retrieval time with the increase in the number of return orders, while the retrieval time in [17] is almost unaffected by the increase in the number of orders, as shown in Figure 9. However, because the range tree index structure is used in

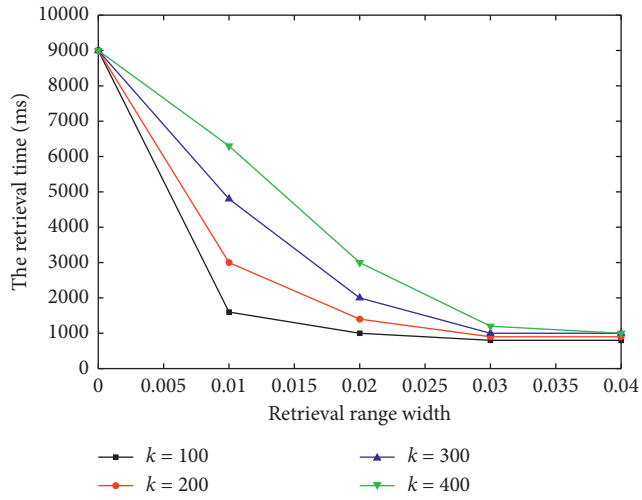

Figure 7: Relationship between retrieval time and retrieval range width.

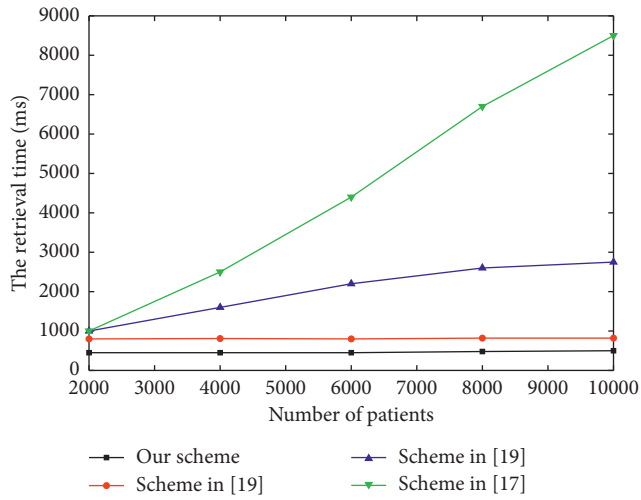

Figure 8: The retrieval time changes with the number of patients $m(k = 100, d = 0.03)$.

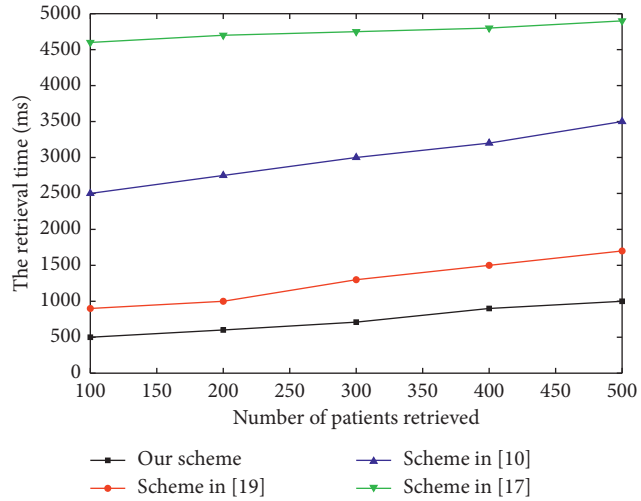

Figure 9: The retrieval time varies with the number of data records returned $(m = 6000, d = 0.03)$.

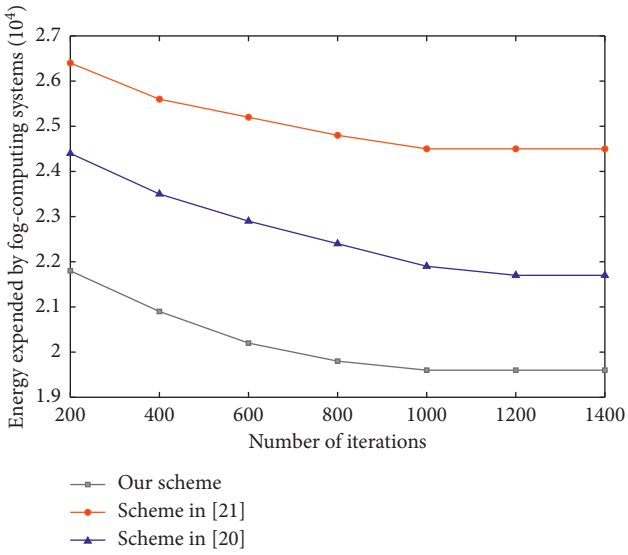

Figure 10: Iterative convergence of scheduling algorithm (number of orders = 600).

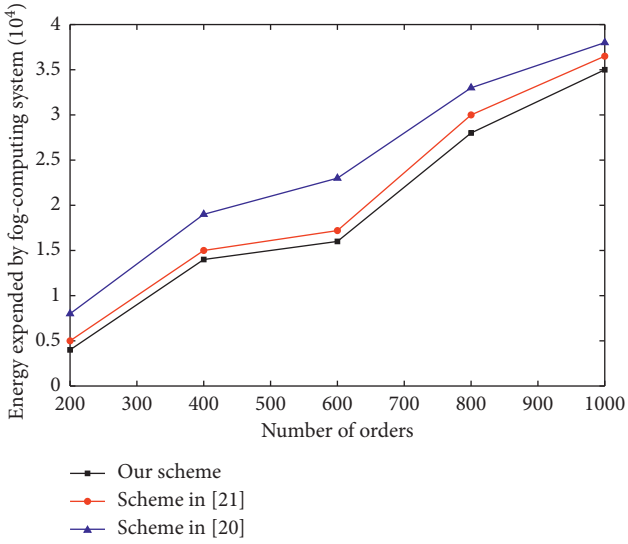

Figure 11: The relationship between the total cost of fog computing system and the number of orders.

this paper, the retrieval time is much less than that of other schemes, which further indicates that the retrieval efficiency of this paper is higher. To sum up, the retrieval efficiency of the scheme in this paper is superior to those from [10, 17, 19]. □

6.3. Analysis of Energy Cost. In this paper, MATLAB 2019a is used to implement the algorithm for server allocation. We set $\beta = 10^{-8} \times$ (MIPS) and $\alpha = 0.6 \times \beta$ (J) in (2), the penalty value M is 0.5, and the number of server group VMs in the fog computing system is 10. The scheme in this paper compares the energy cost with PSO algorithm in [20] and artificial bee colony algorithm in [21].

Figure 10 shows that different schemes have different energy costs and convergence rates with the increase of algorithm iteration times. As seen from Figure 10, the convergence speed of their scheme [20] is slow, and the convergence energy cost is high. In [21], although the convergence energy cost is low, the convergence speed is the slowest, which reduces the efficiency of order processing. Compared with the schemes in [20, 21], our scheme has a higher convergence speed and lower convergence energy cost. The relationship between the energy cost of the fog computing system and the number of orders is shown in Figure 11. With the increase of the number of orders, the scheme in this paper produces lower energy cost than that of the schemes in [20, 21].

7. Conclusion

The traditional retrieval scheme of IoMT is faced with problems such as privacy leakage, low retrieval efficiency, and high system power cost. This paper proposes a data retrieval and analysis service scheme of IoMT under privacy protection based on low-cost fog computing. We set up a fog computing system between the IoMT and cloud services to not only improve the data retrieval efficiency, but also reduce the service delay. We adopt range tree to construct data index and form a multidimensional range tree structure. This improves the retrieval efficiency on the premise of ensuring the index security and the unlinkability of the portal. Moreover, this paper adopts an algorithm to allocate resources for the orders on the fog server group, which significantly reduces the energy cost of the system while ensuring system efficiency. The simulation results show that the proposed scheme not only improves the retrieval efficiency and accuracy, but also significantly reduces energy cost compared with the existing schemes.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request (email: 1711023984@qq.com).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 62001055 and 61802025); Beijing Natural Science Foundation (no. 4204107); Funds of "YinLing" (no. A02B01C03-201902D0); Open Project Program of National Engineering Laboratory for Agri-Product Quality Traceability; and Beijing Technology and Business University (BTBU), under grant AQT-2020-YB4.

References

[1] N. Wang, J. Fu, B. K. Bhargava, and J. Zeng, "Efficient retrieval over documents encrypted by attributes in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2653–2667, 2018.

[2] H. Kim, J. Shin, Y. Song, and J. Chang, "Privacy-preserving association rule mining algorithm for encrypted data in cloud computing," in *Proceedings of the 2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pp. 487–489, Milan, Italy, July 2019.

[3] X. Shi and S. Hu, "Fuzzy multi-keyword query on encrypted data in the cloud," in *Proceedings of the 2016 4th Intl Conference on Applied Computing and Information Technology/3rd Intl Conference on Computational Science/Intelligence and Applied Informatics/1st Intl Conference on Big Data, Cloud Computing, Data Science and Engineering (ACIT-CSII-BCD)*, pp. 419–425, Las Vegas, NV, USA, December 2016.

[4] P. Pandiaraja and P. Vijayakumar, "Efficient multi-keyword search over encrypted data in untrusted cloud environment," in *Proceedings of the 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, pp. 251–256, Tindivanam, February 2017.

[5] S. He, B. Cheng, H. Wang, Y. Huang, and J. Chen, "Proactive personalized services through fog-cloud computing in large-scale IoT-based healthcare application," *China Communications*, vol. 14, no. 11, pp. 1–16, 2017.

[6] Q. Li, J. Zhao, Y. Gong, and Q. Zhang, "Energy-efficient computation offloading and resource allocation in fog computing for internet of everything," *China Communications*, vol. 16, no. 3, pp. 32–41, 2019.

[7] H. K. Apat, B. S. Compt, K. Bhaisare, and P. Maiti, "An optimal task scheduling towards minimized cost and response time in fog computing infrastructure," in *Proceedings of the 2019 International Conference on Information Technology (ICIT)*, pp. 160–165, Bhubaneswar, India, December 2019.

[8] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric internet of things services," in *Proceedings of the 2015 International Symposium on Consumer Electronics (ISCE)*, pp. 1-2, Madrid, Spain, June 2015.

[9] K. H. Abdulkareem, M. A. Mohammed, S. S. Gunasekaran et al., "A review of fog computing and machine learning: concepts, applications, challenges, and open issues," *IEEE Access*, vol. 7, pp. 153123–153140, 2019.

[10] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[11] C. Chen, X. Zhu, P. Shen et al., "An efficient privacy-preserving ranked keyword search method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 951–963, 2016.

[12] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[13] S. Hijazi, A. Page, B. Kantarci, and T. Soyata, "Machine learning in cardiac health monitoring and decision support," *Computer*, vol. 49, no. 11, pp. 38–48, 2016.

[14] G. Redlarski, D. Gradolewski, and A. Palkowski, "A system for heart sounds classification," *PLoS One*, vol. 9, no. 11, Article ID e112673, 2014.

[15] D. Rizk, R. Rizk, and S. Hsu, "Applied layered-security model to IoMT," in *Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, p. 227, Shenzhen, China, July 2019.

[16] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497–4506, 2018.

[17] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[18] M. A. Khan and F. Algarni, "A healthcare monitoring system for the diagnosis of heart disease in the IoMT cloud environment using MSSO-ANFIS," *IEEE Access*, vol. 8, pp. 122259–122269, 2020.

[19] L. Hu, S. Nooshabadi, and M. Ahmadi, "Massively parallel KD-tree construction and nearest neighbor search algorithms," in *Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2752–2755, Lisbon, Portugal, July 2015.

[20] R. Pahlevi, M. A. Murti, and E. Susanto, "The implementation of PID using particle swarm optimization algorithm on networked control system," in *Proceedings of the 2014 International Conference on Industrial Automation, Information and Communications Technology*, pp. 35–38, Bali, Indonesia, August 2014.

[21] Z. Zhang, W. Su, and K. Zhou, "Airborne radar sub array partitioning method based on artificial bee colony algorithm," in *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 484–489, Chengdu, China, March 2019.