Classification and Storage Method of Medical Health Monitoring Data Based on Bayesian Algorithm

Xiaofeng Li and Zhongwei Chen(✉)

College of Information Engineering, Guangxi University of Foreign Languages, Nanning 530222, China
top00112233@163.com

Abstract. In the storage of medical health monitoring data, the overall storage is chaotic and takes up more backup space. A classified storage method of medical health monitoring data based on Bayesian algorithm is designed. Collect and integrate medical health monitoring data. Data collection is realized by relying on the unified data exchange platform in the big data platform. Each collection node uploads the required data by accessing the data exchange platform. Data integration is mainly about data preprocessing of collected medical and health big data. Based on Bayesian algorithm, an integrated naive Bayesian algorithm based on knowledge transfer is designed to implement the classification and processing of medical health monitoring data. A hybrid storage model is designed to implement the classified storage of medical health monitoring data. Test the data classification and storage performance of the design method. The test results show that the method has less startup nodes, less computing power requirements as a whole, can realize the sharing of medical and health monitoring data, and the data classification time is relatively short as a whole, which proves that the method has a broad application space.

Keywords: Bayesian Algorithm · Medical Health · Monitoring Data · Data Classification Storage

1 Introduction

With the improvement of people's material living standards, the trend of population aging in many countries is gradually obvious. With that comes a rise in the number of people with geriatric and chronic diseases. According to the "Report on Nutrition and Chronic Disease Status of Chinese Residents" released in 2015, the prevalence of hypertension and diabetes among adults aged 18 and over in my country in 2012 was on the rise compared with 2002, and the number of deaths from chronic diseases in the country that year accounted for It accounts for 86.6% of the total number of deaths, which has attracted much attention [1]. Chronic disease treatment is a process that requires long-term monitoring. In order to save medical resources, it is a good solution to use the mobile medical method to monitor the human body for a long time and in real time by the health monitoring terminal.

S. Wang (Ed.): IoTCare 2022, LNICST 501, pp. 88–101, 2023.
https://doi.org/10.1007/978-3-031-33545-7_7

In recent years, the development of medical informatization has promoted the integration of medical and mobile communication technology, and gave birth to the concept of mobile medicine. Mobile medicine generally refers to the provision of medical services and information through mobile communication tools, such as PDA, mobile phone, etc., including remote patient monitoring, online consultation, online consultation, wireless access to electronic medical records, etc. [2]. It provides an effective way to solve the problem of shortage and uneven distribution of high-quality medical resources. It not only changes the traditional treatment mode, but also enables patients to perform simple operations on the mobile medical service platform established by medical institutions, so that they can complete the appointment for treatment and avoid the toil of queuing up for registration in the hospital. At the same time, it also realizes telemedicine monitoring with the help of mobile communication technology, and effectively manages the health information of patients.

Reference [3] proposed a method to build a safe storage model of health data based on the alliance blockchain. This method divides medical institutions based on the distribution of medical resources, and uses the share authorization certification mechanism and the practical Byzantine mechanism to achieve the safe transmission of health data. According to the characteristics of the blockchain, such as decentralization, security and trustworthiness, and tamper prevention, health data is stored on the blockchain to improve the security of health data. Reference [4] proposed a method to build a safe storage model of health data based on the blockchain. This method designs a safe storage model of health data based on the blockchain storage structure to improve the practicality of the storage model. And optimize the storage process of the blockchain, and improve the storage efficiency through the consensus mechanism of random number election. Although the above methods can complete the classification and safe storage of health data, there is still a high intrusion rate and it is difficult to resist external attacks. Therefore, in this study, a classification storage method of medical health monitoring data based on Bayesian algorithm is proposed.

2 Classification and Storage of Medical Health Monitoring Data

2.1 Data Collection and Integration

Collect and integrate medical health monitoring data. Data collection is realized by relying on the unified data exchange platform in the big data platform. Each collection node uploads the required data by accessing the data exchange platform. Data acquisition and exchange include a variety of access methods, including but not limited to data tables, front-end libraries, real-time communication, documents, data distribution, etc. when accessing the new system, the appropriate and best access method should be selected according to the exchange requirements and characteristics of the new system. Data collection methods are divided into document based and intermediate library based forms [5]. Based on document collection, nodes can upload documents directly, or convert and upload the original library through the document conversion tool. Based on the collection of the intermediate library, the node writes the data to the front-end computer and uploads it. Combine two ways to collect.

Document-based collection means that each data collection node collects according to the document format specified in the data collection specification, and directly uploads the exchanged document to the big data platform by calling Web service; the platform reviews the document format and quality information and feeds it back to the collection node. Document-based data collection is suitable for the collection of unstructured data or structured data whose update frequency is relatively slow. Document-based data collection has the following characteristics: high real-time data collection and scalability, no need to adjust the service interface due to the increase of collection services; Lower maintenance costs, more conducive to management; If the collected data content is large, the pressure on the network and the exchange server will be very large.

The access node takes the document as the carrier and can upload data by calling the Web service. However, due to the different data storage forms of the collection node, the process of data uploading will be different, which can be uploaded in three ways:

(1) Access nodes upload documents directly: for data in the form of documents, access nodes can directly call web services to upload documents;
(2) The access node uploads the data in the document library: the local data stored in the document library in the form of documents. The platform can directly call the service interface of the platform to upload the data of the local document library;
(3) Data is converted into document format and uploaded: local data stored in relational database can be converted into documents and uploaded through document conversion tools.

The document library includes documents in any format, such as XML, CDA, XLS, PDF, DOC, TXT, etc.

In the process of data exchange based on the intermediate library, the intermediate library will be used as the intermediate link of the exchange between the platform and the access node. Database table exchange is a sharing method of medical and health big data based on database table structure. The front-end database table is used as the interface for data acquisition and push between the data exchange system and the access node. The data exchange system exchanges data through the front-end database, and the access node obtains the data of the front-end database or pushes it to the front-end database through bridging. The data and database exchange method has the following characteristics: It is relatively safe not to access the business database; Simple configuration and less operation and maintenance workload; Clear boundaries, clear responsibilities and rights; The data transmission efficiency is high, the real-time performance is high, the exchange mode is flexible, and it can adapt to many scenarios [6].

Data integration is mainly to preprocess the collected medical and health big data. Medical and health big data has the characteristics of large data volume, heterogeneity, asynchronous (sequential) and incompleteness. Therefore, it is undoubtedly difficult to analyze and apply the original data of medical and health data. In order to provide higher quality for the data analysis stage Generally, data preprocessing operation is required

for the target data set of the data processing. There are generally four methods for data preprocessing: data cleaning, data integration, data reduction and data transformation.

(1) Data cleaning: Data cleaning is to serve the high-quality data requirements of subsequent data analysis. The tasks of data cleaning mainly include:

① Fill in missing data values.
Fill in data values that are vacant in the record.
② Noise data smoothing.
Noise refers to random errors in the data. A common way to remove noisy data is to divide the noisy data equally into nearby data.

(2) Data integration: data integration refers to the process of merging data from multiple data sources into one data set. According to different data conditions, the conversion work of data integration is different, such as null value filling, standardized and unified data format, data splitting, data correctness verification, etc.
(3) Data specification refers to the compression of data sets, but the integrity of data must be maintained. There are three types of data protocols:

① Quantity specification: replace the original data with smaller data. Simple original data can be replaced by sampling data. For complex data, the cluster center in cluster analysis needs to replace the original data first.
② Data compression: transform the original data into a compressed form through the compression algorithm, which can keep the content and characteristics of the data unchanged, mainly for image and video data.
③ Dimension reduction: extract the dimensions from the original data, remove those dimensions that have no use value, and only retain the main dimensions, so as to reduce the size of the data set [7].

(4) Data transformation: The purpose of data transformation is to eliminate the difference in the data type and data format of the original data. The specific operation methods are as follows:

① Data normalization: Reducing the attributes with huge differences between the maximum value and the minimum value in the original data according to an appropriate ratio can effectively improve the performance of the data algorithm.
② Data aggregation: Aggregate raw data according to the granularity required for data analysis, such as aggregating daily data into annual data.
③ Attribute construction: Combine several attribute constructions in the original data into a new attribute.
④ Discretization: Replace the data type values in the original data with interval labels, such as the eastern, western, and central regions for regional distribution, and divide the ages into children, youth, middle-aged, and elderly.

2.2 Data Classification

Bayesian classification algorithm is a classification method of statistics, which is a kind of classification algorithm using probability and statistics knowledge. In many occasions, Naïve Bayes (NB) classification algorithm can be comparable to decision tree and neural

network classification algorithm. This algorithm can be applied to large databases, and the method is simple, the classification accuracy is high, and the speed is fast.

Based on Bayesian algorithm, an integrated naive Bayesian algorithm based on knowledge transfer is designed to implement the classification and processing of medical health monitoring data. Ensemble learning is an effective strategy to deal with concept drift in streaming data classification. In ensemble learning, the historical model can be updated and reorganized to meet the current distribution of the latest data blocks. However, there will be a disadvantage when using this method only to deal with the concept drift problem, that is, when the new data block arrives, it only filters the historical model and integrates it in some way, without making corresponding adjustments to the historical model to make it more suitable for the current data. Although this method is simple and easy to understand and implement, it can not fully mine the useful knowledge for learning the latest data distribution only by simply combining the historical models. Therefore, the combination of ensemble learning method and knowledge transfer makes the model more effective in dealing with the problem of concept drift in data flow classification. That is, knowledge transfer is introduced into FWNB algorithm and combined with naive Bayesian algorithm based on forgetting mechanism weighting. In transfer learning, the source domain and the target domain belong to the same learning task, so they are interrelated. It can also use the knowledge of the historical model after knowledge migration to assist in the learning of the latest data. Therefore, an integrated learning algorithm FTENB based on knowledge migration is proposed, which takes the historical model as the initial solution, and then migrates it based on the latest data to make it more suitable for the latest data distribution after concept drift.

The model usage strategy based on knowledge transfer is as follows: The existing ensemble learning models in the classification of data flow have a same characteristic. Whenever the newly arrived data block is classified, the historical model is only analyzed by integrating various methods, without taking into account that the historical model is adjusted according to the latest data and then integrated. So these approaches differ mainly in the various integration and combination approaches to the preserved historical models. While these methods are less complex and easier to implement, doing so alone may not fully exploit the knowledge that is most useful for classification learning on current state-of-the-art data.

Suppose there are two data distributions, P1 and P2, respectively. P1 represents some historical data distribution in the data stream, and P2 represents the current latest data distribution. P1 is different from P2, which means that concept drift has occurred [8]. The classification performance of the historical model learned from the historical data distribution P1 may not be high on the latest data, so in the ensemble model, the importance of this historical model should be reduced, that is, give it a lower weight, some extreme case may be zero, then all the knowledge in the model will not be used. If P1 and P2 are regarded as the source domain and target domain of transfer learning, respectively, then P1 and P2 belong to the same classification learning task, and there is a certain connection between the two. Based on this, after knowledge transfer, the knowledge of the transferred historical model can be used to assist the learning of the latest data. Therefore, it is very advantageous to use transfer learning in ensemble models.

The classification learning algorithm is different, and the learning method is also different, so the method of transferring the historical model according to the new data is also different. In this paper, naive Bayesian algorithm is used as the basic learning algorithm, and a method based on Naive Bayesian model is designed and proposed to adjust the historical model, aiming to adjust the retained historical naive Bayesian model to adapt to new data.

The specific migration process is divided into the following two steps:

Step 1: Combine all the data in the newly arrived data block B_a with the historical data B_i respectively. At this time, the data in the new data block is marked, and the newly generated data block contains both historical knowledge and current knowledge, and does not belong to a specific real data distribution;

Step 2: Build a classifier E_i^a for the new data block B_i^a, obtain new class probability and conditional probability, and generate a new Naive Bayesian model.

After performing knowledge transfer on historical models based on new data, the newly generated models not only contain previous knowledge, but also current latest knowledge. Therefore, the newly generated model E_i^a is a new distribution that mixes various data, and does not really belong to the real distribution of a certain piece of data in the dataset. The naive Bayesian model after knowledge transfer can be more adapted to the latest data B_i^a, but it will also reduce the model difference between E_i^a, so these transferred models will be discarded after use, just for the classification of the current data block service without being retained by the system.

In FTENB algorithm, selecting N historical model can be defined as selecting n storage from a model. Take the stored n models as the starting model, and then implement knowledge migration based on the latest data. The migrated model will be used as the base model in the subsequent integration algorithm.

The integration strategy of models in streaming data classification is actually an effective combination of multiple models. In order to have a better combination effect, it is bound to choose the best base model for integration [9]. Based on this, FTENB algorithm uses the classification effect of historical model on the latest data block as the screening criterion of historical model. Following this criterion, the FTENB algorithm selects the historical model for knowledge migration as follows.

Suppose FTENB can hold n historical models. After receiving B_a, test all saved historical models based on B_a, and at the same time, build FWNB model E_a based on this data. Then check whether the number of models saved in the current system reaches the threshold n. If not, save E_a, otherwise, add E_a to the ensemble model first, and then select the worst model according to the performance of each model on the current data, and remove it from the system. In addition, the removed model is selected from all models in the current system, so both E_i and E_a may be removed.

Each data block $B_1, B_2, \cdots, B_a$ arrives sequentially in the form of a stream. The FTENB algorithm first establishes a naive Bayesian model E_1 on the data block B_1 and saves it in the system. When B_a arrives, based on the latest data block B_a, the knowledge migration is performed on all the stored historical models E_i, and the migrated model E_i^a is obtained. At the same time, a weighted model based on the forgetting mechanism is established on B_a as a new base model E_a. After integrating the historical model with knowledge transfer and E_a, the integrated model at time a can be obtained. At the same

time, the FTENB algorithm also needs to update the historical model in the system in time to ensure that the ensemble model uses the optimal base model.

When integrating classification models, we also need to consider the weight calculation of each model. AUE2 algorithm has the best classification learning ability among the current integration algorithms, so we refer to the integration method of AUE2 algorithm. For all migrated historical models, formula (1) is used to calculate the weight of the model. For the base model E_a trained by Weighted Naive Bayes based on forgetting mechanism on the latest data block B_a, it is the most "perfect" model based on B_a, and formula (2) is used to calculate the weight of E_a.

$$\omega_i^a = \frac{1}{M_i^a + M_i^{a'} + \chi} \tag{1}$$

In formula (1), M_i^a refers to the classification error of E_i^a on B_a; $M_i^{a'}$ refers to the mean square error of the random classification model; χ refers to the positive value.

$$\omega_a = \frac{1}{M_i^{a'} + \chi} \tag{2}$$

The formula for calculating M_i^a is as follows:

$$M_i^a = \frac{1}{|B_a|} \sum\nolimits_{\{c,d\} \in B_a} \left(1 - l_i^a(d|c)\right)^2 \tag{3}$$

In formula (3), $l_i^a(d|c)$ refers to the posterior probability of model E_i^a after migration; c and d refer to the medical and health monitoring data flow in B_a.

The formula for calculating $M_i^{a'}$ is as follows:

$$M_i^{a'} = \sum\nolimits_d l^a(d)\left(1 - l^a(d)\right)^2 \tag{4}$$

In Eq. (4), $l^a(d)$ refers to the prior probability of data stream d in B_a.

Complete the classification of medical health monitoring data.

2.3 Data Classification Storage

Design a hybrid storage model to implement classified storage of medical health monitoring data after classification. The hybrid storage model is introduced in three parts. The first part is the related implementation in the Linux kernel, the second part is the implementation of metadata distinction, and the third part is the implementation of logical block mapping.

In the hybrid storage model, the relevant implementations in the Linux kernel are as follows: The Device Mapper mechanism in the Linux kernel exists in the kernel in the form of modules, which did not appear until the kernel version 2.6, and is mainly used to implement the mapping mechanism for logical volume management. It implements a modular kernel architecture for creating virtual logical devices and managing multiple underlying storage resources. If developers want to use the Device Mapper mechanism to develop, it must be included in the kernel, and a management tool for users to be

installed. The current management tool of Device Mapper uses dmsetup, which mainly includes related configuration and interface library functions. In fact, the role of Device Mapper can be regarded as combining multiple underlying storage devices into a logical device, so that users cannot feel the existence of the underlying device, but can only see the created virtual logical device. In addition, Device Mapper is a "stack" structure, and its underlying device can be composed of Device Mapper. It is mainly composed of 3 kinds of objects: (1) Mapper Device: the logical device displayed to the outside world; (2) Mapping Table: custom rules to link different underlying devices; (3) Target Device: the actual physical device.

Several mapping rules have been defined under the Device Mapper architecture of the kernel to implement different logical devices. First of all, the target driver module is inserted into the Linux kernel in the form of a module. The main function of this module is to redirect or modify the I/O requests of the upper layer. At present, there are many different types of mapping rules in the kernel, including linear rules, soft raid, et al.The data request sent by the user is passed to the block layer, which will be converted into a request of bio structure, and then the bio request will be processed by the unified processing method of the Device Mapper mechanism. Corresponding processing, the algorithm will modify the corresponding address contained in the bio request, and the device pointed to in the bio structure also specifies the underlying physical device corresponding to the physical address. The bio request processed by the mapping rule will still be put into the request queue owned by the block device, and then the method generic_make_request will be used again to submit the request that has been redirected by the Device Mapper mechanism. Therefore, it can be considered that the Device Mapper mechanism just performs a simple redirection operation on the received request, and converts the I/O request to the logical device into a request to the underlying device. The Device Mapper mechanism has an interface that can accept the submission and processing of the upper general block layer and re-initiate the bio request. Therefore, developers can easily implement their own mapping rules according to their own needs, and only need to follow the Device Mapper architecture definition. Rules to write modules, and then insert the module into the kernel, do not need to worry about the interaction with other modules in the kernel, so it has good generality.

In the process of implementing the hybrid storage model, it is not only necessary to distinguish all metadata requests, but also to reduce the movement of the head when accessing the disk, it is also necessary to aggregate all scattered metadata and data requests in the disk, so that different kinds of data are gathered in different areas of each block group. Therefore, the active identification method is used, but on the basis of active identification, Make a few modifications to the model to distinguish all metadata [10].

The layout of the model disk can distinguish some static metadata requests. However, since the directory item is a dynamically allocated data block, the allocation function of the data block needs to be modified (for example, ext2_get_block function, et al.) to make the dynamically allocated metadata block and the statically allocated metadata block together. First, select to reserve some data blocks after the fixed metadata blocks of each block group for allocation to the dynamically allocated directory, so that all statically allocated metadata blocks will be stored after the dynamically allocated metadata blocks,

so that the former part of each block group stores all metadata, while the ordinary files are allocated to the latter part outside the reserved area of the block group. In this way, Aggregate all metadata and data in each block group in the file system.

The metadata in the Ext2 file model mainly exists in two ways, one is fixedly allocated metadata, and the other is dynamically allocated metadata. For fixed allocation metadata, such as superblock, group descriptor table, and inode table, disk blocks are allocated at fixed positions in each block group of the device when mkfs formats the file model. For example, when the Ext2/Ext3 file model divides the disk storage space into 128 MB block groups, at this time, the disk blocks at the beginning of each group are reserved for storing metadata as data blocks of statically allocated metadata, but For a directory (the disk block of which is used to store directory items, which is also a kind of metadata), the disk block is dynamically allocated in the corresponding block group when the user creates it, which is a kind of dynamically allocated metadata.

For statically allocated metadata blocks, the number and storage location of disk blocks occupied can be calculated from the relevant information recorded in the superblock when creating the file system. However, for dynamically allocated metadata, in order to put it together with statically allocated metadata blocks, it is necessary to reserve a certain area of data blocks after fixing the disk blocks reserved for metadata during file model formatting to store these dynamically allocated metadata.

The data blocks allocated to the directory are uncertain, because directories, like regular files, are created according to the needs of users, so the creation of directories is related to application scenarios. Under different load scenarios, the number of data blocks occupied by directories will be very different. For example, for deep directory loads, many directories will be created. By testing different loads, estimate the number of directories created under different conditions, and then reserve almost the same data blocks to store the directory items of the directory, and the number of data blocks reserved here to store the directory can also be dynamically adjusted.

Each driver type (target_type) of Device Mapper has a target type name as an identifier, which also corresponds to a module in the kernel. There are multiple callback functions for each driver type, including some necessary and non-essential callback functions, such as constructors, destructors, and mapping functions that must be implemented. The callback function is optional. The Device Mapper mechanism is used in the hybrid storage model, a target device type called raid_target is defined as the identifier, and four callback functions are implemented as required. The name of the callback function and the introduction of related functions are shown in Table 1.

Table 1. Names and related functions of callback functions

Serial number	Function name	Callback function
1	Constructor	raid_ctr()
2	Destroy function	raid_dtr()
3	Mapping function	raid_map()
4	State function	raid_status()

Constructor raid_ctr(). This function is mainly used to allocate the required memory and data structure, parse the target rules configured by the user, and use the various parameters obtained by parsing to initialize each field in the data structure.

The destruction function raid_dtr() is just the opposite of raid_ctr(), so this function will release the data structure and memory allocated in raid_ctr(). So when the user exits, this function will release the memory space and various resources allocated by the constructor and return it to the model.

Mapping function raid_ map () is a very important function. It is the interface for the whole model to obtain the bio request from the upper layer and process the bio request to the lower layer. In the constructor, the received bio request needs to be segmented according to the required size, and then the segmented request is submitted to the mapping function for filtering and redirection. Finally, the processed bio request is submitted to the underlying device for further processing.

Status function raid_ status () is provided for the user to call, so that the user can obtain various states of the current device from the recorded information.

The function dm_register_target registers the defined target_typ. This method will be called when inserting target_type using insmod. After calling this function, a corresponding descriptor target_type will exist in the kernel, which can then be used for some of the above functions.

The function dm_unregister_target unregisters the target_type that has been registered in the kernel. Each target type of the kernel is unregistered when the module rmmod is used, and this function is used at this time.

The implementation process of model initialization is as follows: The hybrid storage model logical device is described by a data structure raid_c.

The process of reading and writing is mainly controlled by raid_ map () function implementation. First of all, the requests sent from the user space will be processed by the modified file system. After processing, the requests will be handed over to the general block layer. In the general block layer, the user requests on the upper layer will be converted into bio structure for description, and then processed by the unified processing module of the Device Mapper mechanism to divide the bio requests. After pre-processing by Device Mapper, the bio is handed over to raid_ map function, and then use the configuration you have created to map bio to the underlying device disk. Among them, the most important process is to modify the sector number field Bi corresponding to the bio request_ sector and the underlying storage device Bi corresponding to the bio request_ bdev, in this way, can retransmit the bio redirected using the mapping rules to the block layer for processing.

In the callback method raid_map of map, the bio request that has been divided and modified by Device Mapper will be received. First, the corresponding logical block device number (LBN) will be calculated through the bi_sector field of the record in the request bio structure, because the file model Modify so that in each block group of the file model, the first half of the blocks are used to store metadata and the second half of the data blocks are used to store data. All the currently requested LBNs that can be obtained are in each block group. The first half or the second half, so as to judge whether the current bio request is a metadata request or a data request, and decide to redirect the bio request to the SSD or HDD according to the judgment result. At the same time,

modify the bi_bdev and bi_sector fields to point to the corresponding underlying device. And the corresponding physical address. For example, after the request sent by the user is modified in the Linux kernel I/O stack, the bio will eventually be sent to the created logical device, and the redirection algorithm will check the LBN requested by the bio. If the number is less than EXT2_METADATA_COUNT, the request will be redirected to RAID1 consisting of high-speed solid state disks (SSD). If the requested LBN number is greater than or equal to EXT2_METADATA_COUNT, it will be redirected to RAID5 consisting of ordinary disks (HDD).

3 Storage Test

3.1 Experimental Data and Test Environment

For the designed classification and storage method of medical health monitoring data based on Bayesian algorithm, its classification and storage performance is tested through experimental data. In a hospital, some medical health monitoring data were collected as experimental data, a total of 1000000 pieces, which were divided into five types: blood pressure data, blood oxygen data, ECG data, blood glucose data, body fat data. The performance format of each type of data is shown in Table 2.

Table 2. Scope and basic data types of various types of medical data

Serial number	Data type	Basic data type	Company	Medical data range
1	Blood pressure	Integer	mmHg	80–140
2	Blood oxygen	Float	%	88–100
3	Electrocardiogram	Double	Times/minute	62–105
4	Blood sugar	Integer	Bomol/L	3.2–7.0
5	Body fat	Integer	index	15.2–45.0

The test environment in the experiment is as follows: both Target and Initiator use centos 6.5 system, Target kernel version uses 2.6.34.1, and Initiator uses kernel version 2.6.32.26. On the Target side, two 120 G SSDs are used to form RAID1 to store metadata, and three 1T HDDs are used to form RAID5 to store data. The initiator uses the discovery command to mount the devices in the Target node to the local. Then, the mapped logical device is formatted as an Ext2 file model on the initiator side. In order to eliminate the impact of network bandwidth, 10 Gigabit network cards are used on both the initiator and target sides.

The basic information of the configuration environment in the test is shown in Table 3.

Test the classified storage performance of the design method in this environment. The testing tools used mainly include Postmark, Filebench and fio.

After setting the experimental data and environment, set the experimental process: verify the classification performance of the text method by taking the number of starting nodes, computing power requirements, and data classification time as indicators; Taking

Table 3. Basic information of the configuration environment in the test

Serial number	Project	Specific information
1	Memory	Kingston 32GB DDR
2	Operating system	CentOS 8.2
3	Linux kernel version	Linux 2.6. 22
4	HDD	6TB
5	SDD	240G
6	Network card	Intel 8 Gigabit

the intrusion rate as the indicator, and the reference [3] method and the reference [4] method as the comparative experimental method, the security of the method in this paper is verified.

3.2 Test Results

First, test the number of start-up nodes and computing power requirements of the design method, and the test results are shown in Table 4.

Table 4. Number of startup nodes and computing power requirements for the design method

Data volume (GB)	Number of startup nodes (PCs.)	Computing power demand	Whether to realize data sharing
5	5	Small	Yes
10	8	Small	Yes
15	11	Small	Yes
20	15	Small	Yes
25	17	Small	Yes
30	19	Small	Yes
35	21	in	Yes

The test results in Table 4 show that the number of startup nodes of the design method is small, and the overall demand for computing power is small. At the same time, it can realize the sharing of medical and health monitoring data.

Then test the classification time of medical health monitoring data of the design method, and the specific test results are shown in Fig. 1.

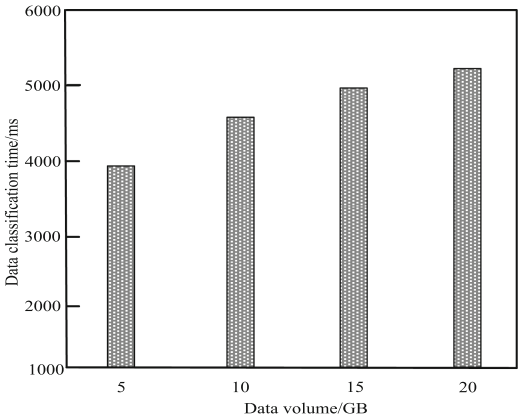

Fig. 1. Classification time of medical health monitoring data

According to the test results in Fig. 1, the classification time of the medical health monitoring data of the design method is relatively short as a whole. With the increase of the amount of data, the classification time also increases, but the overall increase is low.

In order to further verify the security storage performance of the method in this paper, the intrusion rate is taken as the comparison index, and the method in this paper is compared with the method in reference [3] and the method in reference [4]. The intrusion rate comparison results of the three methods are shown in Fig. 2.

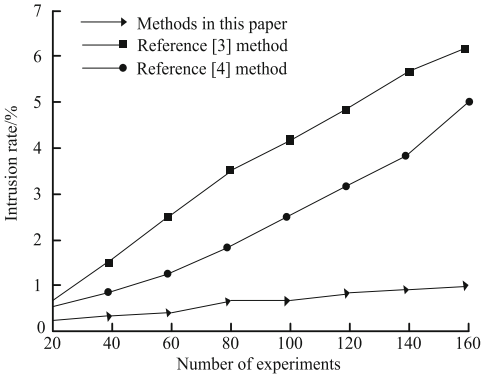

Fig. 2. Comparison Results of Medical Health Monitoring Data Security

It can be seen from the test results shown in Fig. 2 that, under many comparative experiments, the intrusion rate of the method in this paper is always lower than that of the two literature comparison methods. The maximum intrusion rate of the method in this paper is only about 1%, while the maximum intrusion rate of the method in Reference [3] and Reference [4] is more than 6% and 5% respectively. Therefore, this method can reduce the intrusion rate of health monitoring and improve the security of storage.

4 Conclusion

With the increasing emphasis on chronic diseases and the development of health monitoring terminals, medical and health big data and precision medicine have become the current development direction. Therefore, it is necessary to realize the integration and sharing of medical and health data. Therefore, a classification storage method of medical health monitoring data based on Bayesian algorithm is designed to realize the rapid classification and classification storage of medical health monitoring data. The experimental results show that this method can shorten the time of medical health monitoring data classification, reduce the intrusion rate of medical health monitoring data, improve the security of health monitoring data storage, and has great significance for the integration and sharing of medical health data. In the future research work, targeted encryption measures should be taken for different intrusion modes to effectively ensure the storage security of medical health monitoring data.

Aknowledgement. 2020 Guangxi College and University Young and Middle aged Teachers' Basic Scientific Research Ability Improvement Project "Research and Development of Panoramic Campus Roaming System Cross platform" (2020KY63022).

References

1. Nguyen, T.T., Cai, K., Immink, K., et al.: Capacity-approaching constrained codes with error correction for DNA-based data storage. IEEE Trans. Inf. Theory **67**(8), 5602–5613 (2021)
2. Cai, K., Chee, Y.M., Gabrys, R., et al.: Correcting a single Indel/Edit for DNA-based data storage: linear-time encoders and order-optimality. IEEE Trans. Inf. Theory **67**, 3438–3451 (2021)
3. Feng, T., Jiao, Y., Fang, J., et al.: Medical health data security model based on alliance blockchain. Comput. Sci. **47**(04), 305–311 (2020)
4. Bai, Y., Man, J., Zhang, H.: Secure storage model of electronic health records based on blockchain. J. Comput. Appl. **40**(04), 961–965 (2020)
5. Liu, H., Ye, L.: Implementation of health data sharing based on distributed storage and computing technology. Chin. J. Health Inf. Manag. **18**(01), 143–146 (2021)
6. Hu, Y.C., Lokhandwala, M., Te, I., et al.: Varifocal storage: dynamic multi-resolution data storage. IEEE Micro **40**(3), 47–55 (2020)
7. Zheng, D., Xue, L., Yu, C., et al.: Toward assured data deletion in cloud storage. IEEE Network **34**(3), 101–107 (2020)
8. Cheng, L., Qi, Z., Shi, J.: Blockchain based secure storage and sharing scheme for EHR data. J. Nanjing University Posts Telecommun. (Nat. Sci. Ed.) **40**(04), 96–102 (2020)
9. Chang, J., Shao, B., Ji, Y., et al.: Efficient identity-based provable multi-copy data possession in multi-cloud storage, revisited. IEEE Commun. Lett. **24**(12), 1 (2020)
10. Yue, G., Liu, J., Liu, F.: Medical big data filling and classification simulation based on decision tree algorithm. Comput. Simul. **38**(01), 451–454+459 (2021)