# Design of Healthcare Data Analysis System Based on Operational Research and Differential Evolution Algorithm

Xue Jin[1(✉)] and Bin-bin Liu[2]

[1] Hongshan College, Nanjing University of Finance and Economics, Nanjing 210003, China
jx42154@163.com
[2] Nanchang Normal University, Nanchang 330032, China

**Abstract.** Considering the existing medical data has many characteristics, such as many and complex classification, based on operational research and differential evolution algorithm, a medical data analysis system is designed, which consists of distributed storage module, medical intelligent assistant decision module, data cache module, data analysis module, authority verification module, overall decision module and processor module. Among them, data analysis module and overall decision module are the core of the whole system. In the data analysis module, the data output from the data cache module is split and processed by differential evolution algorithm and clustering mining algorithm, and transmitted to the overall decision module. The decision theory in operational research is used to describe and analyze the data analysis process of decision makers, According to the analysis results, the system makes overall decision, and then realizes the design of the medical and health data analysis system. The experimental results show that the system has a high proportion of practical rules extraction, and the time consumption is shorter, which can further improve the level of medical and health data analysis.

**Keywords:** Operational research · Differential evolution algorithm · Health care data · Hadoop system · Data analysis system

## 1 Introduction

In recent years, due to the explosive development of Internet technology, massive and complex user data is generated every moment on the network, and human society has entered the era of big data [1]. Among them,research on big data of medical industry informatization has become the focus at home and abroad. The Research Report of hit2013 in the United States shows that in 2011, the amount of medical data in the United States was about 150eb, and the Caesar group alone stored nearly 32pb of medical data. In 2012, the total amount of medical data in the world has reached 2.7zb, nearly three times that in 2010 [2]. IDC, an international data company, predicts that by 2020, the total amount of global data will be close to 40zb.


S. Wang et al. (Eds.): IoTCare 2021, LNICST 414, pp. 119–135, 2022.
https://doi.org/10.1007/978-3-030-94185-7_9

In China, the process of medical information construction started late compared with foreign countries. In order to better integrate with the international development trend, "smart medical industry" is listed as a key planning project in the 12th Five Year Plan. According to the statistics of the Ministry of Health, in 2014, China invested 27.51 billion yuan in the informatization construction of the medical industry, and in 2015, the total investment scale exceeded 30 billion yuan. Although the investment is greatly enhanced, the contradiction between the huge medical examination data and the backward medical information storage technology is becoming increasingly significant. According to the latest survey report of Jishi information, under normal circumstances, the amount of medical information data for a healthy young person is basically within 1MB, for middle-aged people with some physical problems it is about 40MB, and for the middle-aged and elderly people with several chronic diseases, it will reach 3-5gb. Due to the limitation of technical means, the vast majority of medical institutions and hospitals adopt the traditional as.

The system of literature [3] was first proposed by renowned American scholar Morris Colen in the 1990 s, which belonged to the typical C/S architecture, the client/server working mode. In the system, the clients distributed in various departments are connected with the server of the data center through the enterprise office network. Under normal circumstances, SQL server or Oracle database is installed on the server of data center to store medical data, and the electronic medical records generated by various medical examination reports and medical diagnosis results of patients, as well as various medical audio and image data are stored in the database. The client system submits the query, insert or modify instructions to the server's medical database through UI, and the instruction execution results are fed back to users at all levels through UI. The advantages of his are simple structure, mature technology and relatively easy system design and implementation.

However, when dealing with large-scale data sets, due to the limitation of processing capacity, this paper proposes a healthcare data analysis system based on operational research and differential evolution algorithm to solve the problem that the traditional relational database can not meet the needs of deep mining and big data analysis.

## 2 Design of Healthcare Data Analysis System Based on Operational Research and Differential Evolution Algorithm

### 2.1 Design Distributed Storage Module

In the distributed storage module, in order to realize the classified storage of heterogeneous data in the traditional medical database, it needs the help of Hadoop system component collection and third-party integration tools [4–6]. The component system of Hadoop project is shown in Fig. 1.

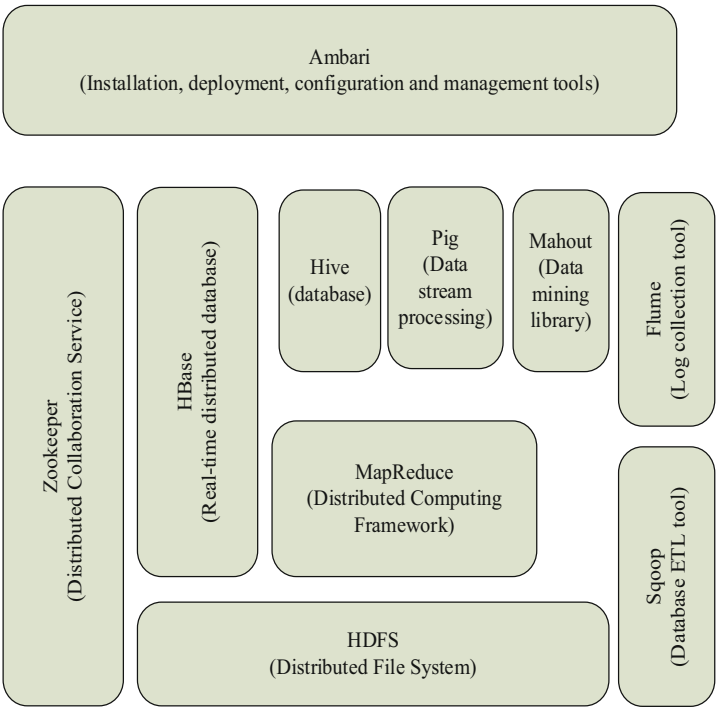


**Fig. 1.** Hadoop component systems for the project

During the design of distributed storage module, hive database, HBase database and sqoop tool are used in the component system.

(1) Hive is a database based on HDFS designed by Facebook. Its feature is that it can convert database files into data forms, and all SQL commands will be converted into map/reduce parallel processing program, which enables users to complete data query, writing and other related operations in the form of commands without programming.

(2) HBase is a real-time distributed database based on "column mode" running on the top of HDFS. Different from the traditional relational database based on parallel schema, HBase uses BigTable data model to store data in the form of table, and divides the table into rows and columns. The sparse permutation mapping table (key/value) composed of row keywords and column keywords can be used for map/reduce processing. Compared with hive, HBase has the function of random read-write and real-time access to large-scale data, and BigTable's data model perfectly combines data storage and parallel computing, which is very suitable for unstructured data storage and management.

(3) Sqoop tool is an independent open source sub project developed by Apache project team, and it is the first third-party integration tool of Hadoop to realize data transmission between traditional database (such as SQL server or Oracle) and Hadoop.

Distributed storage module is the key part of the system, and it is also the basis of medical intelligent decision support and medical big data analysis.

Before distributed storage and processing of original healthcare data, cellular and HBase are installed on named nodes, and then use the Java API provided by sqoop tool to connect with the traditional medical database. Then, for all kinds of data that need to be imported, the structure properties are judged.

(1) If it is structured data, sqoop tool will connect hive through JDBC/ODBC interface, and then query whether the storage form corresponding to the data already exists, if not, create a new table and save it in hive; If it already exists, judge whether the amount of data exceeds the set threshold. If not, store it directly in a hive; If it exceeds, you need to add a partition before saving it in the hive.
(2) When the data is unstructured, sqoop will connect to HBase through the HBase interface and submit the insertion request; After the request is responded, scan the HBase table and locate the insertion position. At the same time, set the timestamp to insert the data into the HBase database [7–9].

In order to write data correctly, it is necessary to configure Hadoop cluster, including the running parameters of the guard and the running environment of Hadoop. The corresponding relationship between configuration items and configuration files is shown in Tables 1 and 2.

**Table 1.** Daemon and configuration items

| Serial number | Daemon name | Corresponding configuration item |
|---|---|---|
| 1 | Name node | HADOOP_NAMENODE_OPTS |
| 2 | Data node | HADOOP_DATANODE_OPTS |
| 3 | Job tracker | HADOOP_JOBTRACKER_OPTS |
| 4 | Task tracker | HADOOP_TASKTRACKER_OPTS |

Each daemons runs independently in the background of Linux system. The namenode process is responsible for namespace management and file access, the datanode process is responsible for connecting data nodes, and the jobtracker and tasktracker are responsible for user job scheduling and slicing execution.

**Table 2.** Hadoop main configuration files for clusters and related functional descriptions

| Serial number | Profile name | File function description |
|---|---|---|
| 1 | /etc./host | Specify the corresponding relationship between name node, datanode and IP address |
| 2 | /etc./profile | System environment variable configuration file |
| 3 | /Hadoop/conf/hadoop-env.sh | Configure Java home and Hadoop environment variables |

(*continued*)

**Table 2.** (*continued*)

| Serial number | Profile name | File function description |
|---|---|---|
| 4 | /Hadoop/conf/core-site.xml | Configure HDFS temporary directory, address and port number |
| 5 | /Hadoop/conf/hdfs-site. xml | Configure the log file directory and the number of data backups |
| 6 | /Hadoop/conf/mapred-site.xml | Configure the IP address and port of job tracker |
| 7 | /Hadoop/conf/master | Add named node IP |
| 8 | /Hadoop/conf /slave | Add data node IP |
| 9 | /Hive/conf / hive-site. xml | Add HDFS root directory, jobtracker IP and port |
| 10 | /HBase/conf/hbase-site.xml | Add master node IP, port and zookeeper location directory |
| 11 | /Sqoop/conf sqoop-env.sh | Add HDFS directory, hive and HBase location directory |

After the Hadoop cluster configuration is completed and the original data is imported successfully, the client starts to create the distributed file system HDFS. The data writing process is as follows:

(1) The data layer client development library starts the data node and sends RPC connection access request to the named node of the access control layer.
(2) The named node checks whether the file to be created already exists and the operation permission of the creator. If the check is successful, a record is created for the file; If the check fails, an exception will be thrown to the client [10, 11].
(3) When the RPC write request is responded, the client development library divides the files to be written into multiple packets, applies for new blocks from the named node, and generates a "block report" from the mapping list of local files and HDFS data blocks to submit to the named node.
(4) The named node returns the configuration information of the managed data node to the client, and the client will write it to each data node in the form of pipeline according to the IP address of the data node.

In the process of data writing, the data in HBase database can be imported into HDFS through its own writing tool "hbase org.apache.hadoop.hbase.mapreduce. Export"; The data in hive database can be imported into HDFS by executing the "insert rewrite" command. The rest of the implementation process is completed by the bottom layer of Hadoop without user intervention.

### 2.2 Design of Medical Intelligent Auxiliary Decision Making Module

In the process of patients' actual medical treatment, a large number of medical examinations are usually needed. However, due to the difference in patients' physique, the medical examination items for the same type of diseases may show different data results in the examination process of different patients. Therefore, after the medical examination, some patients need to undergo a period of observation and treatment to determine the specific types of the disease. The research purpose of medical intelligent decision-making module is that doctors can quickly identify patients' condition and get more accurate pre diagnosis conclusions by means of advanced IT technology and supported by medical and health data analysis system, so as to optimize clinical process and improve the efficiency of hospital [12–14].

In the existing electronic medical records of hospitals, there are many medical examination data and personal information of patients with confirmed diseases. Many hospitals are limited by IT technology level, and effectively extract and make full use of many practical information in the electronic medical records. The intelligent decision-making module can map/reduce all the electronic medical records stored in HDFS, summarize the data values of various medical examination items of different diseases and generate auxiliary detection template for doctors' reference. The study can reduce the rate of misdiagnosis to some extent, and provide valuable advice for doctors.

The process of map/reduce processing is as follows:

Mapper algorithm:
(1) Read EMR file, if file < > Φ & Not EOF (all files) then loop to read the string into the variable STR;
(2) If STR = diagnosis result then key = disease name;
Else continues to read backward;
End If
(3) If STR = medical examination data then key = disease name & medical examination item name;
Value = the value of a medical examination item corresponding to the disease;
Else continues to read backward;
End If
(4) Write each group (key, value) key value prior to the intermediate result file;

Because a certain disease may correspond to multiple medical examinations, the mapper algorithm adopts the way of key value combination in the design process, which generates multiple (key, value) key value pairs for each medical examination result of the disease and outputs them to the intermediate result file for subsequent processing by the reducer algorithm.

Reduce algorithm:

(1) If intermediate result file Φ & Not EOF (all intermediate result files) then circularly reads each group of (key, value) key value pairs;

(2) When a key value is read for the first time

If key = name of a specific disease & name of a specific medical examination item then

Max= Value; Min= Value

(3) Cycle to read all (key, value) key value pairs, for the same key value;

If Value' > Max then Max = Value';

If Value' < Min then Mim = Value';

(4) The (min, max) interval corresponding to each key value is written into the final result file; The innovation of the algorithm is that the mapper process adopts the variable way of key value combination, and transforms the merging result from data statistics to interval generation, which solves the problem of different forms of key values of the same nature, and makes the merging result get good extension and expansion, which is more practical.

After mapping/reduce processing of all electronic medical records in the hospital, all medical examination items corresponding to each type of disease form a reliable interval value. However, there is no effective data association between the independent interval result files, and many similar diseases may be accompanied by similar medical examination results. Therefore, in order to achieve the accurate characterization of the disease, it is necessary to further mine and analyze all kinds of data.

The coupling degree model is applied to the generation of association rules, and combined with Apriori algorithm, the optimization algorithm is described as follows:

Algorithm 1: the process of frequent itemsets discovery algorithm is as follows:

Input: initial transaction set B, and set the minimum support count threshold min_ supp

Output: frequent item set L

```
L1 =fmd_ frequent_ l-item sets(B); //Get 1 itemset
For(k=2; Lk -1 ! = null;   k++) do{
Ck=apriori_ gen(Lk-1); //Generating candidate sets
For each t in B do {
Ct =subset(Ck, t);
For each c∈C do
C.Count + + ;
}
Lk={ c∈Ck| C.Coun t≥min_ supp }
}
Return L; //Return frequent itemsets
```

After generating frequent projects, a subroutine is designed to generate candidate sets_Rule() and pruning subroutine calculate_X(), which is the auxiliary association rule algorithm.

(1) generate_ Rules subroutine
The function of this subroutine is to generate a candidate k-item set according to the (k-1) item frequent set in Apriori algorithm, and modify the steps of II [1] < II [2] <… < II [k] in the connection process of the original algorithm to: compare the current two sets Im and in, if there is (Im [1] = in [1]) ∧ (Im [2] = in [2]) ∧… ∧ (Im [k−2] = in [k−2]) and the last one satisfies (Im [k−1] = in [k−1]), Then Im and in are connected to produce k-item sets.

(2) calculate_ X subroutine
The function of the subroutine is to introduce the mathematical model calculation into the pruning process of the original algorithm, prune and delete the new error strong association rules in the k-term set, and reduce the time and space cost of the algorithm [11, 12].

Algorithm 2: the implementation process of association rule generation algorithm is as follows:

Input: frequent item set L, set the minimum confidence threshold min_ Conf, minimum coupling threshold min_ Cou

Output: association rule set

```
For(m=2; m++) do{
Hm ={rule file of Lk has m items} ;
Generate rules by Lk and Hm ;
Generate Hm+1 by Hm using Apriori algorithm ;
Calculate confidence conf / / calculate confidence
If conf < min_ conf then
Delete hm+1 ;
Else{
Calculate cou by EQ (3.10) / / calculate coupling degree
If |Cou| > min_Cou then
Output rule（Lk→hm）to Hm+1 ;
End if
Generate rules by Lk and Hm+1 ;
End if
 }
Return rules; // Return to build rule
```

The innovation of the algorithm lies in: introducing the coupling degree model, and dividing the positive and negative confidence interval through the calculation of the coupling degree value, which effectively inhibits the generation of wrong association rules in the interest degree model.

### 2.3 Design Data Writing Interface Module

The data writing interface module provides the data writing interface. The data mainly comes from the data source provided by the data provider. The data source needs to complete the data cleaning and the corresponding data formatting process. The data writing interface module mainly has the following sub functions: identity verification function and data submission function. Before the data provider writes the information to the database, it needs to submit the identity information to the module. The module then sends the information verification request to the permission verification module. The permission verification module will return the consent or refusal according to the submitted identity information. When the application is submitted, the module helps the data provider to complete the data submission.

### 2.4 Design Data Cache Module

The data writing interface module provides the data writing interface. The data mainly comes from the data source provided by the data provider. The data source needs to complete the data cleaning and the corresponding data formatting process. The data writing interface module mainly has the following sub functions: identity verification function and data submission function. Before the data provider writes the information to the database, it needs to submit the identity information to the module. The module then sends the information verification request to the permission verification module. The permission verification module will return the consent or refusal according to the submitted identity information. When the application is submitted, the module helps the data provider to complete the data submission.

### 2.5 Design Data Analysis Module

Data analysis module is designed based on differential evolution algorithm. The main function of data analysis module is to realize the integration of differential evolution algorithm and clustering mining algorithm. The module includes the realization of privacy protection in data analysis and the corresponding clustering algorithm.

The module can be divided into two sub modules: differential privacy mechanism module and clustering algorithm module. The main function of the differential privacy protection mechanism module is to query the corresponding data in the database according to the query function provided by the differential evolution algorithm. The process is mainly divided into two steps. First, read the value of the privacy operation from the database table, then determine the size of the local sensitivity according to the data and the query function, and calculate the noise size through the privacy operation and the local sensitivity, and the real query results and the noise are added to return the query results with noise. The clustering algorithm module mainly reads the data in the cache database, reads the center point from the database center point data table, marks the data after calculation, and stores the data in the final database after marking. When a batch of cache data is clustered, the clustering module will further calculate the

clustered data. In this step, the cluster distance and intra cluster distance of each cluster will be calculated, and the inter cluster similarity and inter cluster similarity will be generated, and then the corresponding clusters will be split or combined after comparing with the threshold.

### 2.6 Design Permission Checking Module

The authority verification module mainly controls the data writing and data display. The related functional modules mainly include data interface module and data display module. The verification module mainly verifies the content of the data provider's identity information provided by the data writing interface. If the data provider is in the trusted list, it is allowed to write to the database, otherwise, it is refused to submit the application. The function related to the data display module is to verify the validity of the identity of the data miner according to the login information requested by the data miner. If it passes the verification, it will be allowed to log in to the system and display the corresponding data, otherwise it will be refused the login application [15, 16]. □

### 2.7 Design the Overall Decision-Making Module

To design the whole decision-making module based on operations research is to make the whole decision of the system through the decision theory of operations research. Decision making is a kind of activity that people often encounter in production and life [17]. The purpose of decision-making is to summarize the basic rules of decision-making activities, formulate the rules and methods that should be followed when making decisions, so that in the problem of multiple options selection, the decision-maker can make the optimal choice, in order to obtain the best decision-making results or meet the expected purpose of decision-making. According to their nature, decision-making problems can be divided into three types: deterministic decision-making, risky decision-making and uncertain decision-making. Deterministic decision-making refers to the decision-making when the decision-making environment is certain and the decision-making result is also certain. In the deterministic decision-making problem, the desired goal and initial state are generally determined [18, 19]. At the same time, there are many alternative decision-making schemes, and the profit values of these decision-making schemes are determined and can be calculated. For this kind of deterministic decision-making problem, mathematical programming method can be used to solve the optimal decision-making scheme. Risk based decision-making refers to the decision-making when the decision-making environment and decision-making results are uncertain, but their probability of occurrence is known. When making risk decision, the expected value is usually regarded as the decision criterion. The most commonly used decision criteria are maximum expected return and minimum opportunity loss. Uncertain decision-making refers to decision-making under uncertain decision-making environment and unknown decision-making results [20–23]. In uncertain decision-making problems, decision-makers make decisions according to their own subjective attitude. The principles to solve uncertain decision-making

problems usually include pessimistic decision-making criteria, optimistic decision-making criteria, possibility criteria, minimum opportunity loss criteria and eclecticism criteria. The pessimistic decision criterion is to find the minimum profit value of each decision scheme first, and then find the maximum value of these profit values, so as to determine the decision scheme; The optimistic decision criterion is to find the maximum profit value of each decision scheme, and then determine the maximum value of these profit values, so as to determine the equal possibility criterion of decision scheme. It means that when making a decision, if it is uncertain which state has a higher probability and which state has a lower probability, it is assumed that their probabilities are equal. Based on this, the expected value of each strategy profit is calculated, and then the maximum value is found from these expected values to determine the decision scheme; The minimum opportunity loss criterion is also called the minimum regret value criterion. It first converts each element of the return matrix into the corresponding opportunity loss value, that is, the loss caused when the decision scheme with the largest return is not selected, and then determines the maximum opportunity loss value of each decision. Then the minimum value is found from all the maximum opportunity loss values, and the decision scheme is determined. The principle of eclecticism is to first set an optimistic coefficient, then use the optimistic coefficient to get the function value of the maximum and minimum return value of each strategy, and then determine the decision strategy.

## 2.8 Design Processor Module

The processor model selected for the processor module is gt436, and the specific technical data is shown in Table 3.

**Table 3.** GT436 Specific technical data

| Serial number | Project | Specific data |
|---|---|---|
| 1 | Instruction cache | 16 KB |
| 2 | Controller | LCD |
| 3 | Data cache | 16 KB |
| 4 | Guidance system | NAND flash |
| 5 | UART | 3 Channels |
| 6 | PWM timer | 4 Channels |
| 7 | I/O port | |
| 8 | RTC | |
| 9 | 10 bit ADC | 8 Channels |
| 10 | SPI | 2 Channels |
| 11 | Bus-iic interface | |
| 12 | USB device | |
| 13 | USB | |
| 14 | MMC card interface & SD main card | |
| 15 | PLL frequency multiplier clock | |

# 3 System Performance Testing

## 3.1 Experimental Environment

According to the actual operation of the system, the experiment is carried out in the network center of Nanhua Hospital Affiliated to Nanhua University. Combining with the existing test system database and using Hadoop cluster to realize data classification and replication, the experiment is imported into HDFS, which lays the foundation for the subsequent intelligent decision support test and big data statistical analysis test.

## 3.2 Experimental Equipment and Topology

The main hardware equipment used in the experiment includes: hospital internal medical server, personal PC, optical switch and disk array storage platform. The equipment list is shown in Table 4.

**Table 4.** List of hardware test equipment

| Equipment name medical server | Number | Equipment model | Parameter configuration | | |
|---|---|---|---|---|---|
| | | | CUP | Memory | Hard disk |
| Personal PC | 1 | Lenovo x3850 | Intel(R) Xeno(R) E7-4820V3 @ 1.9 GHz | 256 GB | 5 TB |
| Disk array | 3 | Lenovo Qitian M4350 | Intel (R) Core (TM) i5–3470 @ 3.2 GHz | 8 GB | 1 TB |
| Storage platform | 1 | EMC vnx 5300 | Maximum number of drives: 125; Known number of virtual resource blocks: 8 total capacity: 10 TB | | |
| Optical switches | 1 | EMC DS-300B | Interface/Quantity: optical fiber/24 ports; Throughput: 8 gb/s; Fuselage structure: 1RU | | |

The system is developed with Java language tools, and the built Hadoop cluster runs on the Linux operating system platform. The list of experimental software equipment is shown in Table 5.

**Table 5.** List of software laboratory equipment

| Serial number | Equipment name | Version number |
|---|---|---|
| 1 | Development tool | JDK 1.7 |
| 2 | Operating system | Ubuntu Linux 14 |
| 3 | Distributed cluster | Hadoop-2.6.28 |

All the hardware devices used in the experiment are deployed in the central computer room of Nanhua Hospital Affiliated to Nanhua University. The EMC vnx

5300 disk array storage platform is used to realize the storage expansion of his medical server, and the devices are interconnected through the EMC ds-300b optical switch.

The experiment uses three Ubuntu Linux nodes to build a distributed cluster, including a name node and two data nodes. Hadoop distributed architecture software is installed respectively. User interface and operation center are all encapsulated in name node, hive and HBase databases and data export tool scoop are installed, and all user operations will be completed in name node.

### 3.3 Cluster Building

Due to the limitation of experimental conditions, it is impossible to import all the existing medical data into HDFS during the implementation of distributed storage of medical data. Therefore, 50535 patient electronic medical records were collected as the test data set.

Before the construction of Hadoop distributed cluster, the host name and IP address of each node (master node and slave node) in the cluster should be set to ensure that the host name and IP address can form a correct mapping relationship after the cluster is running and can be resolved normally. The configuration information of platform nodes is shown in Table 6.

**Table 6.** Hadoop platform node configuration information

| Name of node | Node type | Host name | IP address | Gateway |
|---|---|---|---|---|
| Name | Master | Master | 192.168.1.14 | 192.168.1 |
| Node | – | Hadoop | 1 | 1 |
| Data Node 1 | Slave | Slave1. Hadoop | 192.168.1.142 | 192.168.1.1 |
| Data Node 2 | Slave | Slave2. Hadoop | 192.168.1.137 | 192.168.1.1 |

After the information of each node is set up, the Hadoop distributed experimental platform should be built according to the actual information in its configuration table. The main steps of cluster implementation are as follows:

(1) Using VI text editor, modify / etc. / host name file to specify host name and IP address for each node; Then modify the / etc. / host file, configure the host DNS server information, and add the host names and IP addresses of all nodes in the cluster.

(2) In the actual running process of Hadoop, name node will manage (start or stop) the daemons of each datanode through SSH (secure shell). When executing instructions between nodes, there is no password input support, so it is necessary to set SSH free password access for each node to log in to each other.
In order to realize public key authentication without password, a key pair, including a public key and a private key, should be generated on the server master, and then the public key should be copied to all the client slave. When

master connects to salt through SSH, salt will generate a random number, encrypt the random number with master's public key, and then send it to master; After receiving the encrypted data, the master uses the private key to decrypt, and then sends the decrypted data back to slave; After slave confirms that the decryption number is correct, master is allowed to connect with it.

The generated key pair is stored in the/home/Hadoop/. SSH directory, and then cat ~ /. SSH/ID is executed_ rsa.pub >> ~ /.ssh/authorized_ Keys, which sets the ID_ Rsa.pub is added to the authorized key, and then the SSH copy ID command is used to transfer the public key to the slave node.

All relevant settings need to be completed after the SSH service is restarted. The above steps are copied to other nodes to realize SSH password free access to all nodes in the cluster.

(3) Install JDK 1.7, edit / etc. / profile file, and configure Java environment variables. The SCP command is used on the master node to copy all files after Java installation to all slave nodes, and set the same environment variables to ensure that the cluster can run Java programs normally.

(4) After installing Hadoop, add the location of Hadoop directory to / etc. / profile file, edit hadoop-env.sh file and add Java home directory. In the process of Hadoop configuration, core-site.xml, hdfs-site.xml and mapred-site.xml are mainly involved, which correspond to HDFS distributed file system and map / reduce processing architecture respectively.

When configuring Hadoop on the master, you need to add the address and port number of name node in the core-site.xml file first; Then, the backup number of data is configured in hdfs-site.xml file; Finally, set the job tracker address and port number in the mapred-site.xml file. After copying all the configurations to slave node, formatting HDFS file system and closing firewall, the cluster can be built.

The working state of the built distributed cluster is shown in Fig. 2.

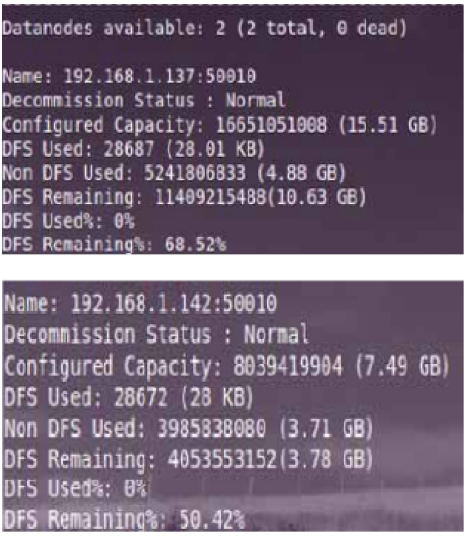


**Fig. 2.** Working state of distributed clusters

After the successful construction of the distributed cluster, sqoop tool and JDBC interface program developed on the Java language platform are used to connect with the SQL Server 2012 database of the medical server and import the data to the master node.

### 3.4 Practical Rule Extraction Test

After the test data set is successfully imported, a test template is generated. The result interval of each disease check item in the template is saved in the Reduce Out Put output file. In order to further test the practical rule extraction performance of the system, part of the data in the training data set accideiits.data, which is commonly used in the field of data mining, is selected as the test data source, and extracted 500 rules for testing, of which 320 rules are defined as practical strong association rules, and the rest are non-practical rules. The test results are shown in Table 7.

**Table 7.** Test results

| Threshold setting | min_Cou = 0.3 | | | min_Cou = 0.4 | | |
|---|---|---|---|---|---|---|
| | Dig total (Article) | Practical rules (Article) | Practical proportion | Dig total (Article) | Practical rules (Article) | Practical proportion |
| min_conf = 0.7 min_supp = 0.1 | 384 | 315 | 82% | 293 | 226 | 77% |
| min_conf = 0.7 min_supp = 0.2 | 202 | 149 | 74% | 165 | 104 | 63% |
| min_conf = 0.7 min_supp = 0.3 | 97 | 66 | 68% | 82 | 48 | 59% |

It can be seen from the above results that when the minimum confidence (min_-conf) threshold is the same, setting a smaller minimum support (min_supp) threshold can mine more association rules, and the proportion of practical strong association rules is also higher. high. Under the same minimum confidence and minimum support settings, the system has a high proportion of practical rule extraction, thus effectively completing the pruning of useless association rules.

### 3.5 Time Consumption Test

Table 8 shows the time-consuming changes of the design system when the number of electronic medical records continues to grow.

**Table 8.** Time consumption test results

| Serial number | Number of electronic medical records (copies) | Time consuming (ms) |
|---|---|---|
| 1 | 10000 | 103 |
| 2 | 20000 | 148 |
| 3 | 30000 | 169 |
| 4 | 40000 | 187 |
| 5 | 50000 | 206 |

The experimental results show that, with the increasing number of electronic medical records, the consumption time of the design system grows slowly, and the total consumption time is short, mainly because the system uses differential evolution algorithm to optimize the performance of the system transmission module.

## 4 Conclusion

With the rapid development of medical industry information construction, all kinds of heterogeneous data show an explosive growth trend. However, the traditional systems have encountered varying degrees of technical bottlenecks, so it is difficult to achieve the two-way expansion of hardware performance and work efficiency. Therefore, a healthcare data analysis system based on operational research and differential evolution algorithm has been studied and implemented, which makes up for the shortcomings of the existing system to a certain extent. However, in the process of this study, only the sample data was analyzed, which led to the results having a certain impact, and the next step will continue to be analyzed.

## References

1. Domadiya, N., Rao, U.P.: Improving healthcare services using source anonymous scheme with privacy preserving distributed healthcare data collection and mining[J]. Computing **103** (4), 1–23 (2021)
2. Shilo, S., Rossman, H., Segal, E.: Axes of a revolution: challenges and promises of big data in healthcare. Nat. Med. **26**(1), 29–38 (2020)
3. Stahlhut, R.W., Porterfield, D.S., Grande, D.R., et al.: Characteristics of population health physicians and the needs of healthcare organizations. Am. J. Prev. Med. **60**(2), 198–204 (2021)
4. Fan, K., Zhu, S., Zhang, K., et al.: A lightweight authentication scheme for cloud-based RFID healthcare systems. IEEE Netw. **33**(2), 44–49 (2019)
5. Mbizvo, G.K., Bennett, K.H., Schnier, C, et al.: The accuracy of using administrative healthcare data to identify epilepsy cases: a systematic review of validation studies. Epilepsia **61**(5), 1–17 (2020)
6. Liu, S., Liu, G., Zhou, H.: A robust parallel object tracking method for illumination variations. Mob. Netw. Appl. **24**(1), 5–17 (2018). https://doi.org/10.1007/s11036-018-1134-8

7. Xiao, M., Hill, C., Vacquier, M., et al.: Retrospective analysis of the effect of postdischarge telephone calls by hospitalists on improvement of patient satisfaction and readmission rates. South. Med. J. **112**(7), 357–362 (2019)
8. Johansson, M., Finizia, C., Persson, J., et al.: Cost-effectiveness analysis of voice rehabilitation for patients with laryngeal cancer: a randomized controlled study. Support. Care Cancer **28**(11), 5203–5211 (2020)
9. Hassan Zadeh, A., Zolbanin, H.M., Sharda, R., et al.: Social media for nowcasting flu activity: spatio-temporal big data analysis. Inf. Syst. Front. **21**(4), 743–760 (2019)
10. Liu, S., Bai, W., Liu, G., et al.: Parallel fractal compression method for big video data. Complexity **2018**, 2016976 (2018)
11. Golob Jr., J.F., Kreiner, A.: Prevention of surgical infections: building or renovating a new intensive care unit. Surg. Infect. **20**(2), 107–110 (2019)
12. Zhang, Y., Rodrigues, J., Seah, W., et al.: Guest editorial special issue on wearable sensor-based big data analysis for smart health. IEEE Internet Things J. **6**(2), 1293–1297 (2019)
13. Liu, S., Fu, W., He, L., Zhou, J., Ma, M.: Distribution of primary additional errors in fractal encoding method. Multimedia Tools Appl. **76**(4), 5787–5802 (2014). https://doi.org/10.1007/s11042-014-2408-1
14. Wang, Q., Liu, R., Men, C., et al.: Temporal-spatial analysis of water environmental capacity based on the couple of SWAT model and differential evolution algorithm. J. Hydrol. **569**, 155–166 (2019)
15. Pan, Z., Fang, S., Wang, H.: LightGBM technique and differential evolution algorithm-based multi-objective optimization design of DS-APMM. IEEE Trans. Energy Convers. **36**, 441–455 (2020)
16. Hua, Y., Sui, X., Zhou, S., et al.: A novel method of global optimisation for wavefront shaping based on the differential evolution algorithm. Opt. Commun. **481**, 126541 (2021)
17. Elaziz, M.A., Xiong, S., Jayasena, K., et al.: Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowl.-Based Syst. **169**(APR.1), 39–52 (2019)
18. Poczeta, K., Kukasz, Ł., et al.: Reprint of: analysis of an evolutionary algorithm for complex fuzzy cognitive map learning based on graph theory metrics and output concepts. Biosystems **186**, 104068–104068 (2019)
19. Ye, X., Chen, H., Kuang, Q., et al.: Solving time-dependent reliability-based design optimization by adaptive differential evolution algorithm and time-dependent polynomial chaos expansions (ADE-T-PCE). Microelectron. Reliab. **114**(3), 113815 (2020)
20. Dashti, A., Noushabadi, A.S., Raji, M., et al.: Estimation of biomass higher heating value (HHV) based on the proximate analysis: smart modeling and correlation. Fuel **257**(Dec.1), 115931.1–115931.11 (2019)
21. Ma, X., Zhang, K., Zhang, L., et al.: Data-driven niching differential evolution with adaptive parameters control for history matching and uncertainty quantification. SPE J. 1–18 (2021)
22. Yza, B., Hwa, C., Qla, B., et al.: Automatic data clustering using nature-inspired symbiotic organism search algorithm - ScienceDirect. Knowl.-Based Syst. **163**, 546–557 (2019)
23. Zhang, K., Zhu, G., Ma, J., et al.: Parameter analysis and estimates for the MODIS evapotranspiration algorithm and multiscale verification. Water Resour. Res. **55**(3), 2211–2231 (2019)