

Research Article

Design and Development of a Big Data Platform for Disease Burden Based on the Spark Engine

Chengcheng Li,¹ Jing Gao,¹ Qingwei Pan,² Zhihua Zhou,¹ Yue Yang,³
and Shangcheng Zhou ¹

¹School of Public Health and Management, Guangzhou University of Chinese Medicine, Guangzhou 510006, China

²College of Physical Education and Health, Guangxi Medical University, Nanning 530021, China

³School of Life Science and Technology, Inner Mongolia University of Science and Technology, Baotou 014010, China

Correspondence should be addressed to Shangcheng Zhou; zhoushangcheng@gzucm.edu.cn

Received 31 August 2022; Revised 2 October 2022; Accepted 13 October 2022; Published 6 February 2023

Academic Editor: Amandeep Kaur

Copyright © 2023 Chengcheng Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Objective. This study attempts to build a big data platform for disease burden that can realize the deep coupling of artificial intelligence and public health. This is a highly open and shared intelligent platform, including big data collection, analysis, and result visualization. **Methods.** Based on data mining theory and technology, the current situation of multisource data on disease burden was analyzed. Putting forward the disease burden big data management model, functional modules, and technical framework, Kafka technology is used to optimize the transmission efficiency of the underlying data. This will be an efficient and highly scalable data analysis platform through embedding embedded Sparkmlib in the Hadoop ecosystem. **Results.** With the concept of “Internet + medical integration,” the overall architecture design of the big data platform for disease burden management was proposed based on the Spark engine and Python language. The main system composition and application scenarios are given at four levels: multisource data collection, data processing, data analysis, and the application layer, according to application scenarios and use requirements. **Conclusion.** The big data platform of disease burden management helps to promote the multisource convergence of disease burden data and provides a new path for the standardized paradigm of disease burden measurement. Provide methods and ideas for the deep integration of medical big data and the formation of a broader standard paradigm.

1. Background

The explosive prosperity of Internet of things technology and artificial intelligence has promoted profound changes in the industry worldwide [1]. Information resources are gradually becoming one of the most essential strategic resources. This large-scale, fast-growing, diverse data structure and multidimensional value density data set is often called big data [2]. How to identify and extract the key features of information in a timely, accurate, and efficient manner from the data ocean and conduct rapid analysis has become one of the hot issues [3, 4].

There are many positive explorations in the construction of big data platforms in many fields around the world. A study from Spain explored the construction of a social media

big data platform, which can monitor early signs of depression in 2020 [5]. In the field of transportation, some scholars have realized the dynamic benefits of monitoring of traffic violations through a big data platform [6]. Moreover, the education big data platform achieved the informational management of student courses and effectively improved the teaching quality in the field of education [7]. Some scholars have built a big data platform based on the lambda architecture and applied it in the energy field to realize the visual monitoring of residential power consumption and effectively improve energy utilization efficiency [8]. Not only that, but the big data platform based on the lambda architecture can also dynamically monitor and analyze marine activities and integrate various parameters [9]. In addition to it, cloud computing technology provides new solutions and

computing potential for the storage and management of large-scale data, relying on the scientific Earth big data platform [10].

Big data has wider application scenarios and development potential in the field of medical research. The United States promotes the opening and sharing of big data to the medical industry to realize technological innovation. This is a strategic plan to encourage multinational pharmaceutical enterprises to deeply explore the hidden application value in the massive medical information by implementing the “big data research and development plan” [11]. On this basis, multinational pharmaceutical enterprises try to achieve accurate drug development by deeply integrating big data technology with precision medicine [12]. These highly personalized precision treatment schemes have provided great help to further complete the “human genome project” [13]. In 2015, the United States proposed to establish a global open cancer genomics database and named it the “moon landing project” for cancer [14]. The United States has realized the construction of local full-coverage medical big data through 12 electronic case data centers, 9 medical data centers, and 8 biological information databases in 2018 [15]. Meanwhile, Korean doctors have obtained a large amount of data to implement precision medicine by creating a big data platform for cancer [16]. The application of big data medical platforms can reduce neonatal mortality and disability rates effectively [17]. This is a rapidly developing trend in medical big data research around the world.

China has introduced a series of favorable policies to promote technological development and the industrial landing of big data. In 2016, China issued guiding opinions on promoting and standardizing the application and development of big data in health care, aiming to promote the “healthy China 2030” plan to realize the medical mode of “Internet +” for the whole industry chain [18]. Meanwhile, policymakers respond to national strategic needs by innovating the talent training mode and strengthening the talent training of cross-disciplinary fields such as artificial intelligence and big data. In the “13th five-year plan” for the development of national population health informatization in 2017, it is clearly pointed out that the health care big data service system in 2021, which is composed of regional medical big data centers and supporting clinical data analysis centers, will be initially established and issued by the China Health Commission [19]. The “14th five-year plan” for national informatization in 2021 further emphasizes the exploration of the application effect of artificial intelligence in intelligent clinical assistant diagnosis and treatment, intelligent public health service, and artificial intelligence-assisted drug research and development, and emphasizes the key role of the construction of a scientific research big data platform in the development of big data [20]. Codelot, the national life big data platform, was established in Shenzhen with strong policy support and can provide various functions, including gene detection, biological information analysis, and data mining. Chinese scholars have developed a medical imaging platform that relies on cloud computing technology to enhance the diagnostic efficiency of obstetric imaging [21]. The medical big data platform has also played an active role in the rapid gene diagnosis of the digestive department in clinical practice [22].

Disease burden data has the characteristics of high capacity, rapid growth, diverse types, and low value density [23]. Some scholars have examined a lot of high-quality research in the field of disease burden [24–26]. Therefore, the technological breakthrough of big data in disease burden research is very important. At present, the research on disease burden focuses more on manual analysis using machine learning algorithms. Of course, there is a lack of research on the intelligent extraction of massive medical data [27]. At present, there are few unified big data platforms for disease burden research. Hadoop architecture provides us with inspiration and ideas for our research based on Java and Python languages [28].

This study attempts to solve the following problems from the perspective of artificial intelligence: (1) the big data platform for disease burden is designed and created based on the Hadoop distributed computing framework. (2) Optimize the calculation efficiency and accuracy based on the Spark engine in the traditional distributed computing framework. (3) We try to realize the visual design of the analysis results in order to provide an analysis paradigm for the related research on disease burden in the future.

2. Methods

This study attempts to build a big data platform for disease burden analysis based on the Spark engine, which can achieve the whole process of data dynamic capture, storage, analysis, detection, and visual output in disease burden analysis. It can further optimize the efficiency of machine learning algorithms by embedding Spark into Hadoop [29, 30]. There is a wide gap between Hadoop and Spark in the actual construction process, as shown in Table 1.

There are many similarities between Spark and Hadoop, both of which are open-source cluster computing environments. We call it Sparkmlib to realize real-time calculation, access tracking, and anomaly detection of disease burden data.

2.1. Workflow of Spark Streaming. A discretized stream is the data over time, and it is also a sequence composed of the RDD of each time interval [31]. DStreams will be formed from multiple input sources such as Flume, Kafka, or HDFS, as shown in Figure 1.

Specifically, Spark streaming treats streaming computing as a series of continuous small-scale batch processing. Data will be divided into small batches by reading data from different input sources, with the creation of new batches occurring at uniform intervals. A new batch will be formed at the beginning of each time interval. The data received during the interval will be added to the batch. The batch will stop growing at the end of the time interval.

2.2. Big Data Measurement Paradigm of Disease Burden. The design of the mlb algorithm package based on Spark is simple [32]. First, the data is expressed in the form of an RDD, and then various algorithms are called on the distributed dataset. We try to embed JOINTPOINT software

TABLE 1: Technical differences between Spark and Hadoop.

| | Hadoop | Spark |
|------------------------|---|--|
| Type | Basic platform, including calculation, storage, and scheduling | Pure distributed computing tools |
| Scene | Mass data batch processing (disk iterative calculation) | Massive data batch processing (memory iterative calculation, interactive calculation), massive data stream calculation |
| Price | Low | High |
| Programming paradigm | MAP + REDUCE API level is relatively low, and algorithm adaptability is poor | RDD is a DAG directed acyclic graph The API is top-level and easy to use |
| Data storage structure | The calculation result is on the HDFS disk with a large delay | RDD intermediate operation results are stored in memory with a small delay |
| Operation mode | Tasks are maintained in process mode, and the task starts slowly | Tasks are maintained in a threaded manner, with fast task startup, and can be created in batch to improve the parallel ability |

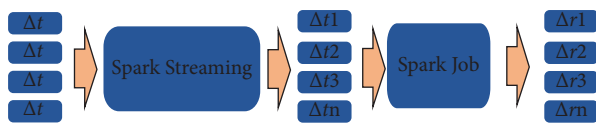


FIGURE 1: Workflow of Spark streaming.

and DISMOD software into our disease burden big data platform. We have built a whole process big data analysis platform based on Spark engine in this study. The algorithm is shown in Figure 2.

Apache Hadoop Yarn is an ideal Hadoop resource manager [33]. As a general resource management system, it can uniformly schedule applications on the platform. The advantages of strong compatibility can bring great benefits to the cluster. Resource management and scheduling are realized by creating application managers and global managers of MapReduce traditional applications for HDFS [34]. As a supplement to MapReduce, hive improves the ability for rapid development of big data platforms and reduces the difficulty of building big data platforms through SQL-like syntax [35].

In general, we split the big data analysis algorithm into the following four steps, as shown in Figure 3.

2.3. Feature Extraction Algorithm

2.3.1. TF-IDF Algorithm. TF-IDF (term frequency-inverse document frequency) is a classic weighting technique for information retrieval and text mining [36].

TF represents the probability of a keyword's occurrence in the text. Normalization can prevent deviations in text mining. The formula is as follows:

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{i,j}}, \quad (1)$$

where $n_{i,j}$ is the number of times the word appears in the file i, j .

Furthermore, this term can have an excellent ability to distinguish categories in the case of fewer documents of T and large IDF. The calculation method is as follows:

$$idf_i = \log \frac{|D|}{|\{j: t_i \in d_j\}|}, \quad (2)$$

where $|D|$ is the entire number of files in the corpus. $\{j: t_i \in d_j\}$ denotes the number of files containing the word t_i (i.e., the number of files with $n_i, j \neq 0$). Usually, $1 + |\{j: t_i \in d_j\}|$ is used to avoid the case that the word is not in the corpus and the denominator is 0. Actually, TF-IDF is better at filtering out common words while retaining important words. The calculation method is as follows:

$$TF - IDF = TF * IDF. \quad (3)$$

2.3.2. FP Growth Algorithm. In order to further optimize the keyword screening strategy, we tried to introduce a FP growth algorithm based on the TF-IDF algorithm. [37] This algorithm finds and recommends high-frequency word pairs by looking at the words used on the Internet. The data consistency of medical electronic cases is weak due to the wide range of sources. Therefore, the FP growth algorithm can comprehensively extract and collect data. The algorithm logic is as follows:

First, building an FP Tree based on a certain data structure, as shown in Figure 4.

FP Tree is not used for simple decision trees, so a class should be created to save each node of the tree. The FP Tree will store the occurrence frequency of item sets. Only when the sets are completely different, the tree will fork.

Second, the conditional pattern library is obtained from the FP Tree, and frequent item sets are mined so as to build a larger set on the basis of a single element itemset. This is an effective way to create a conditional FP Tree, which can repeatedly cycle other single-element items for each frequent item.

3. Model Optimization Algorithm

3.1. ALS (Alternating Least Squares) Algorithm. The Spark platform integrates the ALS algorithm. The optimization of matrix decomposition can be realized quickly by constructing different loss functions [38]. The final task of matrix decomposition is to find two matrices, P and Q , and

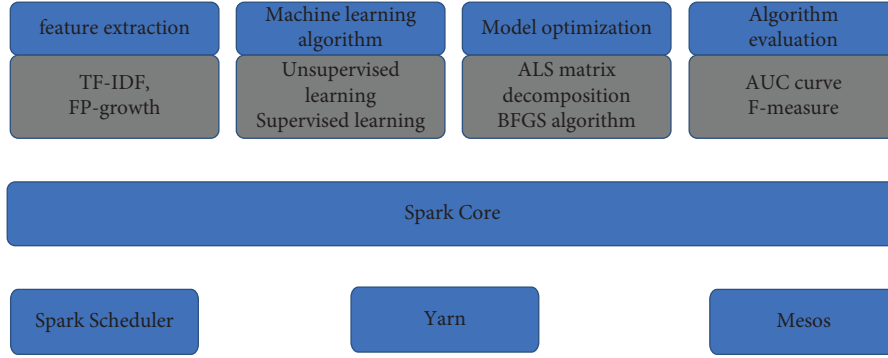


FIGURE 2: Big data measurement paradigm based on Spark engines.

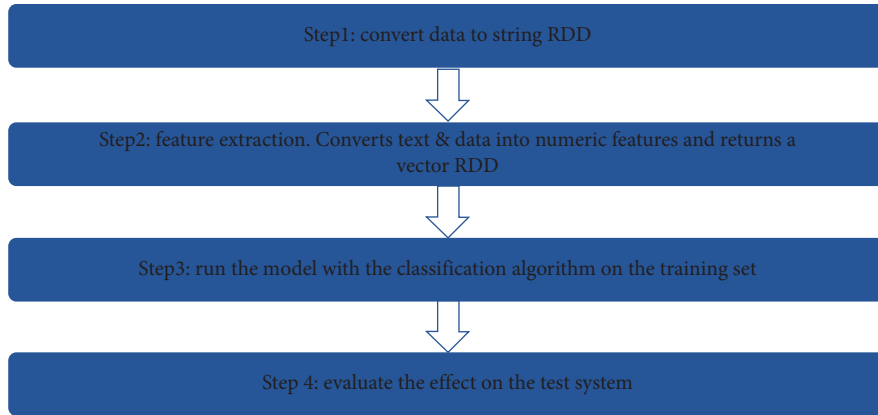


FIGURE 3: Big data analysis steps.

make them approximately equal to the original matrix R after multiplication. The specific algorithm is as follows:

$$R_{m*n} = P_{m*k} \times Q_{n*k}^T, \quad (4)$$

where P and Q are unknown. We assume that Q is known. Therefore,

$$P_{m*k} = R_{m*n} \times Q_{n*k}^{-1}. \quad (5)$$

This means that the R matrix is multiplied by the inverse matrix of the Q matrix, and the result is obtained through

iteration. It is assumed that the solving process is carried out alternately until the error is acceptable.

3.2. L-BFGS Optimization Algorithm. This optimization algorithm evolved from the Newton method and the quasi-Newton method and has been widely used commercially [39]. The specific algorithm is as follows:

Let $f(x)$ be a quadratic differentiable real function, set up again $\chi^{(k)}$ is an estimate of a minimal point of $f(x)$. We expand $f(x)$ into Taylor series at $\chi^{(k)}$ and take the second-order approximation.

$$f(x) \approx \varnothing(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2} (x - x^{(k)})^T \nabla^2 f(x^{(k)}) (x - x^{(k)}), \quad (6)$$

where the middle part of the last item present Hesse matrix of $f(x)$ at $x^{(k)}$. The following formula can be obtained by deriving the abovementioned formula and making it equal to 0:

$$\nabla f(x^{(k)}) + \nabla^2 f(x^{(k)}) (x - x^{(k)}) = 0. \quad (7)$$

Assuming that the Hesse matrix is reversible, the iterative formula of the Newton method can be obtained from the abovementioned formula as follows:

$$x^{(k+1)} = x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}), \quad (8)$$

$$x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}, \quad (9)$$

$$d^{(k)} = -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}), \quad (10)$$

where λ is the compensation obtained by one-dimensional search, which means

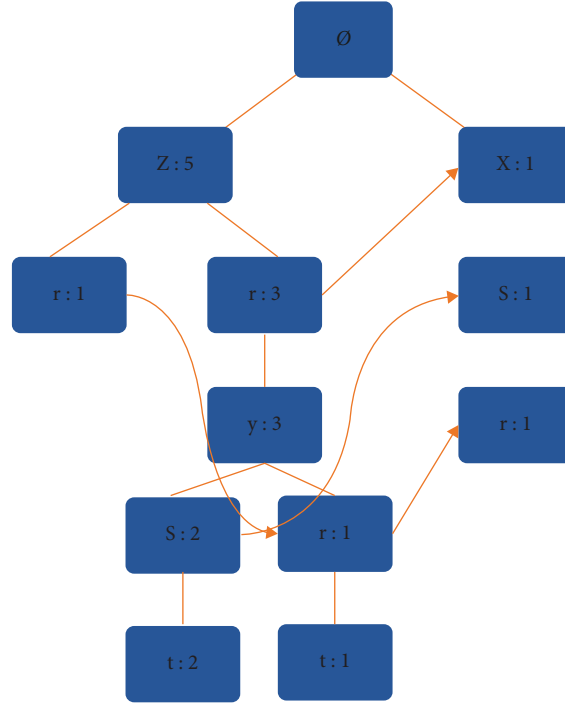


FIGURE 4: FP tree based on certain data structures.

$$f(x^{(k)} + \lambda_k d^{(k)}) = \min f(x^{(k)} + \lambda_k d^{(k)}). \quad (11)$$

We try to construct the approximate matrix of the inverse matrix of the Hesse matrix by analyzing the association

between the inverse matrix and the first derivative. Assume that $\chi^{(k+1)}$ is obtained after the k -th iteration. We expand the objective function $f(x)$ into the Taylor series at point $\chi^{(k+1)}$ and take the second-order approximation to obtain

$$f(x) \approx f(x^{(k+1)}) + \nabla f(x^{(k+1)})^T (x - x^{(k+1)}) + \frac{1}{2} (x - x^{(k+1)})^T \nabla^2 f(x^{(k+1)}) (x - x^{(k+1)}). \quad (12)$$

It can be seen that in the vicinity of $\chi^{(k+1)}$,

$$\nabla f(x) \approx \nabla f(x^{(k+1)}) + \nabla^2 f(x^{(k+1)}) (x - x^{(k+1)}), \quad (13)$$

$$\nabla f(x^{(k)}) \approx \nabla f(x^{(k+1)}) + \nabla^2 f(x^{(k+1)}) (x^{(k)} - x^{(k+1)}), \quad (14)$$

$$p^{(k)} = x^{(k+1)} - x^{(k)}, \quad (15)$$

$$q^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}), \quad (16)$$

$$q^{(k)} \approx \nabla^2 f(x^{(k+1)}) p^{(k)}, \quad (17)$$

$$p^{(k)} \approx \nabla^2 f(x^{(k+1)}) q^{(k)}, \quad (18)$$

$$p^{(k)} = H_k q^{(k)}. \quad (19)$$

Therefore, formula (19) is called the quasi-Newton condition.

When the inverse matrix of the Hesse matrix is a symmetric positive definite matrix, the matrix $H_{(k)}$

satisfying the quasi-Newton condition should also be a symmetric positive definite matrix. We assume that

$$H_{k+1} = H_k + \Delta H_k, \quad (20)$$

$$H_{k+1} = H_k + \frac{(p^{(k)} - H_k q^{(k)})(p^{(k)} - H_k q^{(k)})^T}{q^{(k)T}(p^{(k)} - H_k q^{(k)})}. \quad (21)$$

Then, we define that

$$\Delta H_k = \frac{p^{(k)} p^{(k)T}}{p^{(k)T} q^{(k)}} - \frac{H_k q^{(k)} q^{(k)T} H_k}{q^{(k)T} H_k q^{(k)}}, \quad (22)$$

$$H_{k+1} = H_k + \frac{p^{(k)} p^{(k)T}}{p^{(k)T} q^{(k)}} - \frac{H_k q^{(k)} q^{(k)T} H_k}{q^{(k)T} H_k q^{(k)}}. \quad (23)$$

We swap H equals B , p and q ; therefore,

$$q^{(k)} = B_{k+1}p^{(k)}, \quad (24)$$

$$B_{k+1} = B_k + \frac{q^{(k)}q^{(k)T}}{q^{(k)T}p^{(k)}} - \frac{B_k p^{(k)}p^{(k)T}B_k}{p^{(k)T}B_k p^{(k)}}. \quad (25)$$

We assume that $B_{(K+1)}$ is reversible, then,

$$H_{k+1} = B_{k+1}^{-1}. \quad (26)$$

Finally, the BFGS formula for H was obtained as

$$H_{k+1}^{BFGS} = H_k + \left(1 + \frac{q^{(k)T}H_k q^{(k)}}{p^{(k)T}q^{(k)}}\right) \frac{p^{(k)}p^{(k)T}}{p^{(k)T}q^{(k)}} - \frac{p^{(k)}q^{(k)T}H_k + H_k q^{(k)}p^{(k)T}}{p^{(k)T}q^{(k)}}. \quad (27)$$

The iteration of the D -matrix can be realized through iterative calculation. Furthermore, storage space can be effectively saved by transforming the matrix into a vector. Therefore, the algorithm convergence process for big data can be realized by another approximation of the BFGS algorithm, which is also called the L-BFGS algorithm.

3.3. Validation of the Model. The accuracy of the model can be effectively verified through the accuracy test. Accuracy, classification error rate, precision, recall, and $F1_score$ were the five indicators of score used to evaluate the effectiveness of machine learning algorithms. The specific formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (28)$$

$$Classification\ error\ rate = 1 - accuracy, \quad (29)$$

$$Precision(p) = \frac{TP}{(TP + FP)}, \quad (30)$$

$$Recall(R) = \frac{TP}{(TP + FN)}, \quad (31)$$

$$F1score = \frac{2 * (P * R)}{(P + R)}, \quad (32)$$

where TP means true positive; TN means true negative; FP means false positive; FN means false negative. In the $F1_score$ calculation formula, precision is abbreviated as P , and recall is abbreviated as R . The $F1_score$ value ranges from 0 to 1, with 1 indicating the best and 0 the worst.

4. Results

4.1. Big Data Platform Construction Framework. The big data infrastructure is a stack-type technology architecture [40], Mainly including the following: (1) the foundation layer; (2) the management level: not only the storage and management of existing data, but also the calculation of some data; (3) the analysis layer: embed the corresponding statistical model and machine learning algorithm to analyze the data according to the research objectives; (4) application layer:

mainly for the user's front-end development and visual output, as shown in Figure 5.

4.2. Storage Engine. As the bottom layer of the big data architecture of disease burden, the analysis layer is included in the platform building framework of the basic layer in this study. The big data platform building ideas of Internet enterprises are used for reference and optimized to form a highly automated computing platform that can be expanded horizontally. Specifically, the Kudu storage engine is used for large-scale data storage to balance the performance of HDFS and HBase random reading and writing and batch analysis [41]. The access to a remote MySQL database is implemented by Federated, and on this basis, the basic layer of the big data platform for estimating disease burden is built, as shown in Figure 6.

4.3. Big Data Operation. After the storage engine is selected, since the traditional Hadoop ecosystem cannot complete the ETL and data cleaning work in one MapReduce, the data calculation and processing are realized through the hybrid construction of Spark and MapReduce, as shown in Figure 7.

In order to improve the calculation efficiency and reduce the fault tolerance of the distributed system, the Spark process builds an elastic distributed data set by referring to the idea of functional programming. As a read-only and partitioned data set, RDD forms a directed acyclic graph through operator connection, which significantly improves the computational efficiency. The transformation between various operators is realized through stream, as shown in Figure 8.

Therefore, Spark, MapReduce, and Sparkstreaming jointly complete the cloud computing process of the disease burden big data platform. The sorting and coordination of data resources are jointly completed by Zookeeper and Apache Hadoop Yarn. The zookeeper server processes the client's request through FIFO, allowing the client to connect to any subserver and providing higher performance.

4.4. Multisource Data Acquisition, Cleaning, and Integration. For the construction of the big data platform for disease burden, the management level should unify the management and identification of structured data and unstructured data. Meanwhile, the real-time data should have the ability for rapid query and error identification, and the system response time should be shortened as much as possible. At the same time, sufficient operable space should be provided for the future upgrading of the system.

Therefore, the data types are mainly divided into three categories, mainly including the following:

- (1) Front-end logs: big data from the Internet, medical and health institutions, and mobile phones;
- (2) Back-end log: summarize and transmit data from subservers around the world;
- (3) data from MySQL database of public security, civil affairs and other institutions.

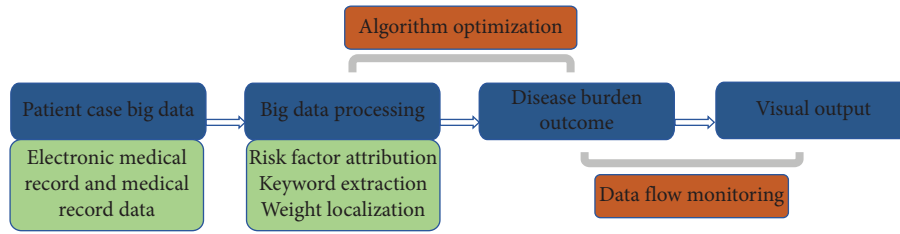


FIGURE 5: Disease burden big data cloud platform architecture.

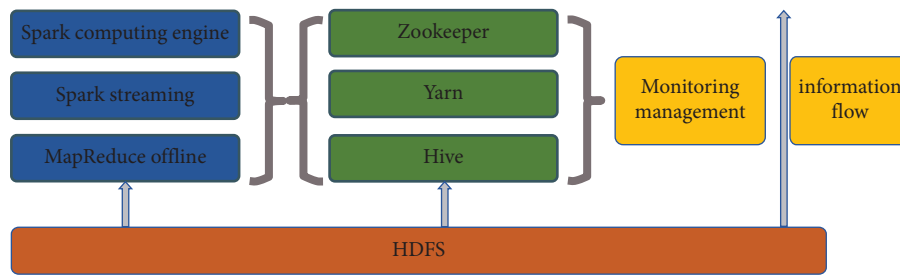


FIGURE 6: Organizational framework and construction idea of the foundation layer.

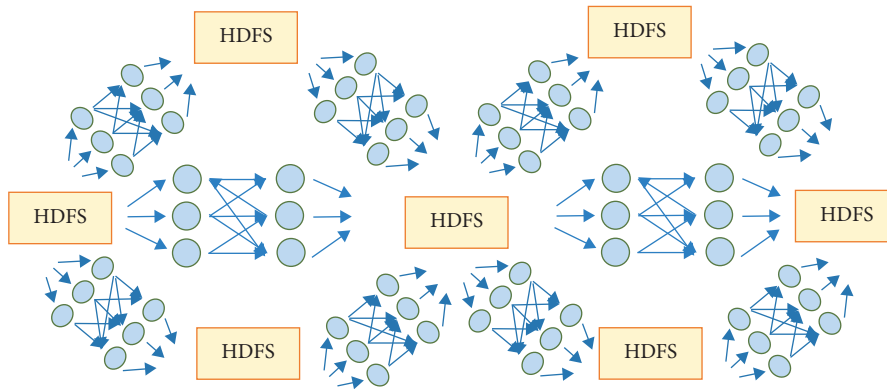


FIGURE 7: MapReduce data processing process.

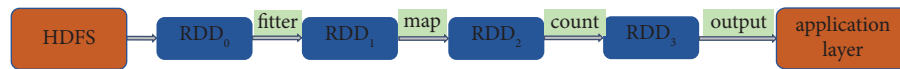


FIGURE 8: Elastic distributed data set framework.

For different types of data flows, the Kafka producer protocol is implemented based on Lua to achieve efficient data collection. The specific framework is shown in Figure 9.

4.5. *Call of Log Data.* For the front-end log collection and access, it is required to have high reliability and availability while responding in real time [42]. By referring to the

construction ideas and failure cases of Internet enterprises, a large number of tests have compared Flume, Scribe and Chukwa's various construction ideas and frameworks, which cannot meet the collection and storage of super-large amounts of disease burden data. Therefore, a set of data acquisition gateway can be developed based on Kafka to complete data acquisition and realize through nginx Lua. The back-end log collection and access can use Go language

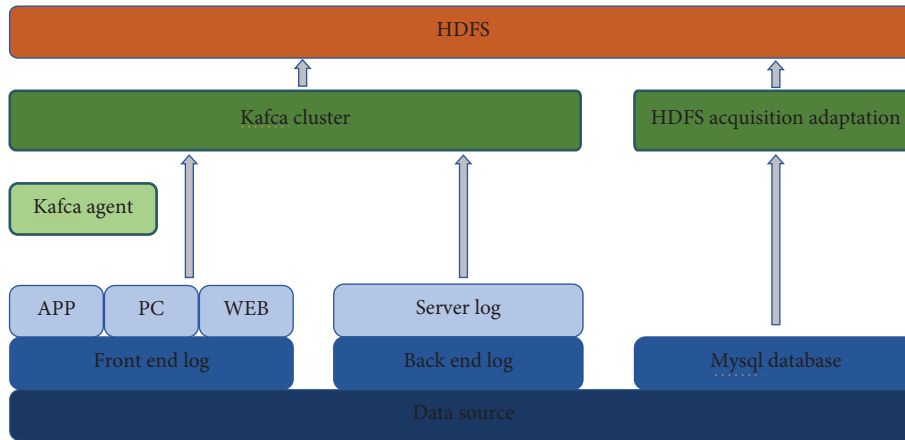


FIGURE 9: Data acquisition framework.

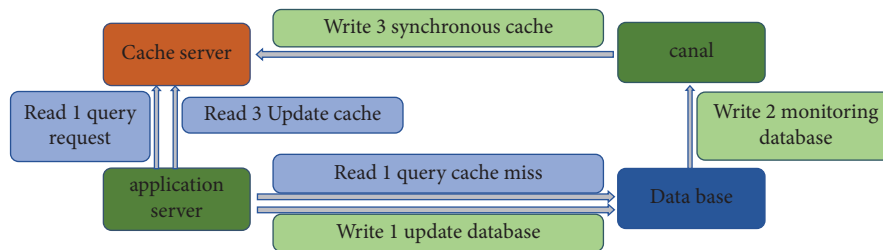


FIGURE 10: MySQL database data collection optimization strategy.

to realize file collection because the server logs are relatively stable. At the same time, multiple reconfigurations and optimizations can be carried out according to the needs of future research. The traditional database collection and access method uses canal to update the cache, which leads to slow MySQL query speed and no QPS. It is easy to cause paralysis if a large number of requests are sent to MySQL. Therefore, we proposed the solution of adding a cache in front of MySQL during the construction, as shown in Figure 10.

Specifically, when the cache is exhausted, MySQL will write another copy to the cache. When the data is inconsistent (MySQL database changes), modify it asynchronously, and then start a canal service to monitor MySQL to make the synchronous cache consistent.

4.6. Application Layer Construction. In the construction of the application layer, we mainly highlight two functions. The first is the real-time monitoring of various types of data streams on the Internet and the stability monitoring of front-end log data streams. The second is the visual presentation of disease burden measurement results. By using AI to design

the UI interface for the visual output of disease burden results, we hope to improve the big data analysis ability of disease burden in China. The elk monitoring system has the following advantages:

- (i) Business data analysis: collect key steps from the front-end information to the back-end for business process analysis.
- (ii) Error log analysis: similar to bugly, after the error log is reported, errors can be summarized, displayed by category, and analyzed in the back end.
- (iii) Data early warning: with elk, it is easy to establish an early warning mechanism for monitoring fields and conduct early warning before large-scale errors occur.

4.7. Visual Output of Disease Burden Results. The results of the analysis of disease burden are visually output through the UI interface. Specifically, it includes four modules: an overall overview, data analysis, data prediction, and data application (Figures 11 and 12). This study designed a visual display of a big data platform based on the calculation

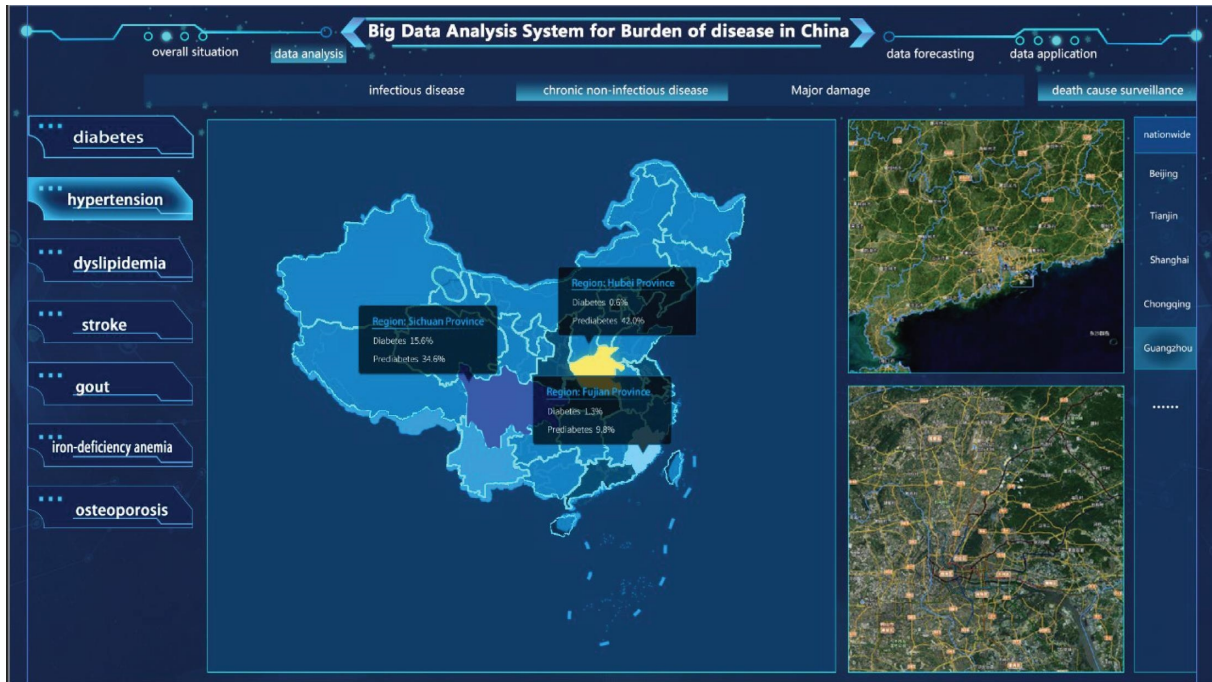


FIGURE 11: Overall overview of China's disease burden big data analysis system.



FIGURE 12: Data analysis module of China disease burden big data analysis system.

results of the burden of diabetes in Guangzhou, China. In the overall overview module, Baidu Map is embedded in the whole disease analysis big data platform, which can realize real-time tracking and analysis of data and visually output the overall situation of diseases according to different disease classifications.

In the data analysis module, the disease burden is mainly measured and demonstrated by big data. Based on the research on regional disease burden, a big data analysis system for urban disease burden has been formed. We try to form an integrated solution from data collection to analysis to decision-making.

5. Conclusion

Medical and health data, especially the massive microdata in the measurement of disease burden, have the characteristics of complex, multisource, and diverse data. At the same time, it also has a complexity and diversity different from other data types, which often makes medical workers and health managers unable to use conventional software tools to acquire, manage, and integrate medical and health data in a short time, making it valuable information. It is particularly important to use big data technology to solve this problem. The construction of the big data platform for disease burden is

different from that of enterprises and businesses. Since the service objects of the big data platform are mainly health department decision-makers and relevant researchers, what application layer can display the analysis results most intuitively and objectively? Can the disease burden analysis results on the IHME website be further improved to enhance readability? Domestic direct research on the construction of a health big data platform is relatively rare, and more research focuses on theoretical research and empirical measurement.

A large number of big data enterprise-level application practices have proved that the disease burden big data platform based on Spark engine can effectively realize the collection and intelligent management of multisource heterogeneous medical data. On the basis of the traditional distributed computing framework, it has greatly optimized and improved different links and levels, especially in the distributed computing of big data, which has high application value and practical significance in this field. In the future, we can apply big data technology to build a unified information management platform and strengthen the construction of data set standards, technical standards, and data sharing and exchange standards, so as to realize the effective application of medical and health data information and promote data integration and information sharing.

There are some shortcomings in this study. (1) Due to the updating of technology, the convergence and quasi-combination of the algorithm do not reach the optimal solution. At the same time, the degree of matching of different data types to the algorithm is not nearly the same, which needs more detailed research in the future. (2) Some prediction algorithms are still under active design and development and are not included in the design framework of the current big data platform. (3) Due to the heavy programming workload required for platform construction, the big data analysis platform built in this study is still under active construction and has not been put into use. The platform will be built and put into use in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (Grant nos. 81973979, 71774049, 71273083, 71333005), Natural Science Foundation of Guangdong Province (Grant no. 2019A1515011496), Social Science Foundation of Guangdong Province (Grant no. GD19CSH04), and the Key Projects of Philosophy and Social Sciences Research of the Education Department of Hubei in China (Grant no. 17ZD024).

References

- [1] Z. Zhao and Q. Hu, "The application of a computer monitoring system using IoT technology," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–11, 2022.
- [2] G. H. Kim, C. M. Jun, H. C. Jung, and J. Ho Yoon, "Providing service model based on concept and requirements of spatial big data," *Journal of Korean Society for Geospatial Information System*, vol. 24, no. 4, pp. 89–96, 2016.
- [3] J. Wang, C. Zeng, Z. Wang, and K. Jiang, "An improved smart key frame extraction algorithm for vehicle target recognition," *Computers & Electrical Engineering*, vol. 97, Article ID 107540, 2022.
- [4] C. Li, C. Liao, X. Meng et al., "Effective analysis of inpatient satisfaction: the random forest algorithm," *Patient Preference and Adherence*, vol. 15, pp. 691–703, 2021.
- [5] R. Martínez-Castaño, J. C. Pichel, and D. E. Losada, "A big data platform for real time analysis of signs of depression in social media," *International Journal of Environmental Research and Public Health*, vol. 17, no. 13, p. 4752, 2020.
- [6] S. Asadianfam, M. Shamsi, and A. Rasouli Kenari, "Big data platform of traffic violation detection system: identifying the risky behaviors of vehicle drivers," *Multimedia Tools and Applications*, vol. 79, no. 33–34, pp. 24645–24684, 2020.
- [7] A. A. Munshi and A. Alhindi, "Big data platform for educational analytics," *IEEE Access*, vol. 9, pp. 52883–52890, 2021.
- [8] A. A. Munshi and Y. A. R. I. Mohamed, "Data Lake lambda architecture for smart grids big data analytics," *IEEE Access*, vol. 6, pp. 40463–40471, 2018.
- [9] P. Tampakis, E. Chondrodima, A. Tritsarolis et al., "i4sea: a big data platform for sea area monitoring and analysis of fishing vessels activity," *Geo-Spatial Information Science*, vol. 25, no. 2, pp. 132–154, 2022.
- [10] C. Xu, X. Du, Z. Yan, and X. Fan, "ScienceEarth: a big data platform for remote sensing data processing," *Remote Sensing*, vol. 12, no. 4, p. 607, 2020.
- [11] P. E. Bourne, V. Bonazzi, M. Dunn et al., "The NIH big data to knowledge (BD2K) initiative," *Journal of the American Medical Informatics Association*, vol. 22, no. 6, p. 1114, 2015.
- [12] P. Vicini, O. Fields, E. Lai et al., "Precision medicine in the age of big data: the present and future role of large-scale unbiased sequencing in drug discovery and development," *Clinical Pharmacology & Therapeutics*, vol. 99, no. 2, pp. 198–207, 2016.
- [13] A. G. Vaithinathan and V. Asokan, "Public health and precision medicine share a goal," *Journal of Evidence-based Medicine*, vol. 10, no. 2, pp. 76–80, 2017.
- [14] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, "The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge," *Contemporary Oncology*, vol. 19, no. 1, pp. A68–A77, 2015.
- [15] G. S. Ginsburg and K. A. Phillips, "Precision medicine: from science to value," *Health Affairs*, vol. 37, no. 5, pp. 694–701, 2018.
- [16] H. S. Cha, J. M. Jung, S. Y. Shin et al., "The Korea cancer big data platform (K-cbp) for cancer research," *International Journal of Environmental Research and Public Health*, vol. 16, no. 13, p. 2290, 2019.
- [17] H. Khazaei, C. Mcgreger, and M. Eklund, "Toward a Big Data Healthcare Analytics System: A Mathematical Modeling Perspective," in *Proceedings of the 2014 IEEE World Congress on Services*, 27 June 2014 - 02 July 2014.
- [18] http://www.gov.cn/zhengce/content/2016-06/24/content_5085091.html.

- [19] <http://wjw.ah.gov.cn/public/7001/52074961.html>.
- [20] http://www.gov.cn/xinwen/2021-12/28/content_5664872.htm.
- [21] W. Lie, B. Jiang, and W. Zhao, "Obstetric Imaging Diagnostic Platform Based on Cloud Computing Technology Under the Background of Smart Medical Big Data and Deep Learning," *IEEE Access*, vol. 8, pp. 78265–78278, 2020.
- [22] Y. Lu, W. Huang, L. Wang, F. Song, Y. Peng, and J. Peng, "Data-enabled Digestive Medicine: A New Big Data Analytics Platform," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 3, pp. 922–931, 2019.
- [23] C. Jo, "Cost-of-illness studies: concepts, scopes, and methods," *Clinical and Molecular Hepatology*, vol. 20, no. 4, p. 327, 2014.
- [24] D. Brecht, "European burden of disease network: strengthening the collaboration.[J]," *The European Journal of Public Health*, vol. 30, no. 1, 2020.
- [25] GBD 2019 Risk Factors Collaborators, "Global burden of 87 risk factors in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019," *Lancet (London, England)*, vol. 396, pp. 1223–1249, 2020.
- [26] H. Aoki, T. Kitano, and D. Kitagawa, "Disease burden of congenital cytomegalovirus infection in Japan," *Journal of Infection and Chemotherapy*, vol. 27, no. 2, pp. 161–164, 2021.
- [27] R. Qiu, M. Hadzikadic, S. Yu, and L. Yao, "Estimating disease burden using Internet data," *Health Informatics Journal*, vol. 25, no. 4, pp. 1863–1877, 2019.
- [28] M. P. Jacob, D. Thomas Js, R. Bunch Dustin et al., "Health care and precision medicine research: analysis of a scalable data science platform," *Journal of Medical Internet Research*, vol. 21, no. 4, 2019.
- [29] J. Neto, A. M. Moreira, and G. Vargas-Solar, "TRANSMUT-SPARK: Transformation Mutation for Apache Spark," *Software Engineering*, 2021.
- [30] V. S. Sharma, A. Afthanorhan, N. C. Barwar, S. Singh, and H. Malik, "A dynamic repository approach for small file management with fast access time on Hadoop cluster: hash based extended Hadoop archive," *IEEE Access*, vol. 10, pp. 36856–36867, 2022.
- [31] X. Wen and J. Hu, "SWEclat: a frequent itemset mining algorithm over streaming data using Spark Streaming[J]," *The Journal of Supercomputing*, vol. 76, no. 10, 2020.
- [32] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: Recent achievements and new challenges," *Information Fusion*, vol. 28, pp. 45–59, 2016.
- [33] X. Cai, F. Li, P. Li, L. Ju, and Z. Jia, "SLA-aware energy-efficient scheduling scheme for Hadoop YARN," *The Journal of Supercomputing*, vol. 73, no. 8, pp. 3526–3546, 2017.
- [34] B. Dong, Q. Zheng, F. Tian et al., "Performance models and dynamic characteristics analysis for HDFS write and read operations: a systematic view," *Journal of Systems and Software*, vol. 93, pp. 132–151, 2014.
- [35] N. Papadakis, P. Kefalas, and M. Stilianakakis, "A tool for access to relational databases in natural language," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7894–7900, 2011.
- [36] A. S. Alammary, "Arabic questions classification using modified TF-IDF," *IEEE Access*, vol. 9, pp. 95109–95122, 2021.
- [37] S. Mai, B. Mahmoud, E. G. Sally, A. Reham, and E. Ali, "An optimized FP-growth algorithm for discovery of association rules," *The Journal of Supercomputing*, vol. 78, no. 4, 2021.
- [38] K. Li, S. Martin, C. R. Rojas, S. Chatterjee, and M. Jansson, "Alternating Strategies with Internal ADMM for Low-Rank Matrix reconstruction," *Signal Processing*, vol. 121, pp. 153–159, 2016.
- [39] D. Yang, G. Li, and G. Cheng, "On the efficiency of chaos optimization algorithms for global optimization," *Chaos, Solitons & Fractals*, vol. 34, no. 4, pp. 1366–1375, 2007.
- [40] N. Bhushana Samyuel and B. A. Shimray, "Securing IoT Device Communication against Network Flow Attacks with Recursive Internetworking Architecture (RINA)," *ICT Express*, vol. 7, no. 1, pp. 110–114, 2020.
- [41] M. S. Wiewiórka, D. P. Wysakowicz, M. J. Okoniewski, and T. Gambin, "Benchmarking distributed data warehouse solutions for storing genomic variant information," *Database*, vol. 2017, Article ID bax049, 2017.
- [42] A. Boron, "Front-end circuit for energetic signal data acquisition," *Przegląd Elektrotechniczny*, vol. 845, pp. 119–121, 2008.