

## *Retraction*

# **Retracted: Data Management Platform of Forest Ecological Station Based on Internet of Things and Big Data Sensor**

### **Journal of Sensors**

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

Copyright © 2023 Journal of Sensors. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

### **References**

- [1] P. Sun, "Data Management Platform of Forest Ecological Station Based on Internet of Things and Big Data Sensor," *Journal of Sensors*, vol. 2022, Article ID 1207745, 8 pages, 2022.

## Research Article

# Data Management Platform of Forest Ecological Station Based on Internet of Things and Big Data Sensor

Ping'an Sun <sup>1,2</sup>

<sup>1</sup>College of Chemistry and Materials Engineering, Zhejiang A&F University, Hangzhou, Zhejiang 311300, China

<sup>2</sup>School of Mathematics and Computer Science, Wuyi University, Wuyishan, Fujian 354300, China

Correspondence should be addressed to Ping'an Sun; 2016122630@jou.edu.cn

Received 18 May 2022; Revised 4 June 2022; Accepted 15 June 2022; Published 29 June 2022

Academic Editor: C. Venkatesan

Copyright © 2022 Ping'an Sun. 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.

In order to solve the problems of low data storage efficiency and poor retrieval performance in forest ecological station, a method of a forest ecological station data management platform based on Internet of Things and big data sensor is proposed. The framework method designs the prepartition algorithm to ensure the uniform distribution of data in the cluster. According to the characteristics of ecological data, the RowKey is scientifically designed to realize the rapid retrieval of ecological data. The Elasticsearch index fragment placement strategy based on index data and server performance evaluation is designed, and the packaging and merging strategy based on data site and time correlation is proposed to improve the storage efficiency. The results are as follows: when the scale of structured data is  $10^8$ , the retrieval time of the system is 1.045 s, which is 3.99 times faster than that of the original HBase. When the scale of unstructured data is  $10^7$ , the packaging small image strategy based on data site and time correlation is 1.15 times higher than that of the sequence file and 1.79 times higher than that of the original HBase. In the case of  $10^4$  concurrent users, the number of queries per second after optimization is 1.88 times higher than the original, the throughput per second is 1.74 times higher than that before optimization, and the system response time is 69.5% lower than that before optimization. The results show that the proposed scheme has significantly improved the performance in the aspects of cluster load balancing and massive structured and unstructured data retrieval efficiency and system throughput, and provide the necessary theoretical basis and technical implementation for the storage and management of forest ecological data.

## 1. Introduction

In recent years, the rapid development of forests in China has caused a large number of ecological and environmental problems. Soil change, land occupation, climate change, air pollution, and water pollution are becoming more and more serious, resulting in forest ecosystems being affected by human activities, frequent urban disasters, and more serious environmental pollution [1]. The overall balance of the forest ecosystem is broken, which will affect the development of cities and reduce the ability to resist external interference, which will directly affect the sustainable development of urban society and economy. With the emergence of ecological and environmental problems, people began to pay attention to and

monitor the change and development of the environment. Ecological environment monitoring is a trade-off measurement method based on time and space. This measurement method can use the professional methods in the field of forest ecology to study the overall structure and function of the system at different levels [2]. Connect the monitored data and information with modern technology, measure and judge, and analyze the feedback effect of different levels and types of objects in the forest ecosystem on the changes of ecology, nature itself, and human activities. By synthesizing different types of feedback information, we can analyze and measure the impact and harm of these interferences from different factors on the environment. At the same time, we can also summarize the trend of their development and change, which

provides a strong basis for evaluating environmental quality and measuring ecological services.

In order to better evaluate the service function of the forest ecosystem, China has established a large number of forest ecological positioning observation stations around the country and conducted long-term continuous positioning observation for typical ecosystem types. So far, China has more than 180 national forest ecological positioning and observation stations (hereinafter referred to as ecological stations), which are distributed in different climatic regions and cover different types of ecosystems. The ecological observation station can carry out long-term and continuous observation and automatically perceive and obtain the ecological factor data of water, soil, gas, and biomass in the observation area. The accumulated data is massive and diverse. However, using a single site to store and manage ecological data cannot meet the storage and management needs of massive heterogeneous ecological data. Each ecological site is independent of each other and gradually forms an information island, which cannot meet the needs of multistation joint analysis, data mining, real-time retrieval, and highly concurrent access for ecological service function evaluation. Moreover, the massive ecological data is diverse and complex and cannot be reconstructed. It is easy to have problems such as heavy computing burden and slow retrieval time in the process of ecological data processing [3]. Therefore, it is very necessary to study the storage and index model of forest ecological big data and establish a massive ecological data management platform based on it. Figure 1 shows a data processing system of Forest Ecological Station Based on Internet of Things technology.

At present, the storage architecture of most ecological stations is mainly a relational database represented by MySQL. Some scholars proposed to realize the application of a digital forest ecological station through MySQL Server Replication Technology and then call the Rest Service API provided by the cloud platform to realize the construction of the data management system of the West Tianshan forest ecological station, which can ensure the consistency of transactions, but it cannot meet the storage and management of massive heterogeneous data in terms of expansibility, fault tolerance, and availability [4]. Aiming at the disadvantages of a traditional relational database, the Hadoop distributed platform and HBase distributed NoSQL database are adopted, which have great advantages in scalability, fault tolerance, and availability. In order to solve the massive GIS data, the researchers designed a Hadoop-based GIS platform and built a forest resource information platform through Hadoop to provide more effective, scientific, and accurate data reference for departments at all levels [5]. The birth of Hadoop and HBase technology has laid an important technical foundation for solving the problem of efficient storage and rapid indexing of ecological big data. However, native Hadoop cannot handle the problem of small files well, and native HBase only supports primary indexing by default [6]. There are two main problems in the above platforms: (1) In terms of storage, native Hadoop does not solve the storage problem of massive small images and does not partition massive data. (2) Native HBase does not provide a good

fast retrieval scheme in the face of a massive data multidimensional query [7, 8].

A large number of small files generated by the ecological station will lead to memory bottleneck and low retrieval performance of NameNode. Hardballing technology is proposed to package small files into large files through a packaging technology, but the preprocessing takes a long time [9]. It was also proposed to merge small files of the same type into large files and establish the index relationship from small files to merged large files. The index relationship is stored in HashMap. If this method fails to hit the cache, the reading performance is not high. The SequenceFile technology is used to realize the massive Internet of Things image packaging and merging strategy. This method can solve the problem of excessive memory in NameNode but does not consider the relationship between images, which is not conducive to association query. For GIS data, some researchers try to use the traditional relational database to store, but this architecture will have the problems of poor expansibility and low access efficiency when facing the massive GIS data of an ecological station. In addition, for the application scenario of massive ecological data, if the reasonable prepartition is not achieved, the data will be tilted. In extreme cases, the distributed storage will become a single node storage [10].

Based on this research, this paper proposes a data management method of the forest ecological station platform based on the Internet of Things and big data sensor. In view of the urgent needs of ecological big data storage and fast index, as well as the disadvantages of the existing forest ecological data schemes, this paper scientifically designs RowKey in storage and then designs a prepartition algorithm to ensure consistent data distribution. For the processing of massive small files, a Redis-based station time cooperation (RBSTC) storage method based on data site and time correlation is proposed. Aiming at the problem of GIS data storage, HBase and GeoTools storage methods are proposed. In terms of index, RowKey is designed according to the characteristics of ecological sites, and the Elasticsearch fragmentation algorithm based on index data and server performance evaluation is designed to optimize the secondary index failure of multicondition retrieval, in order to meet the storage and efficient retrieval of massive heterogeneous ecological data and provide technical support for efficient storage, management, and analysis of ecological monitoring data.

## 2. Research Methods

*2.1. Overall Architecture of Forest Ecological Big Data Platform.* Starting from the characteristics of forest ecological data, a forest ecological big data platform based on Hadoop is proposed, which can be used for data management of ecological stations distributed all over the country [11]. The platform deeply integrates big data, Internet of Things, artificial intelligence, and other technologies to provide users with rapid retrieval, processing, and visual analysis of forest ecological data [12]. Its overall architecture is

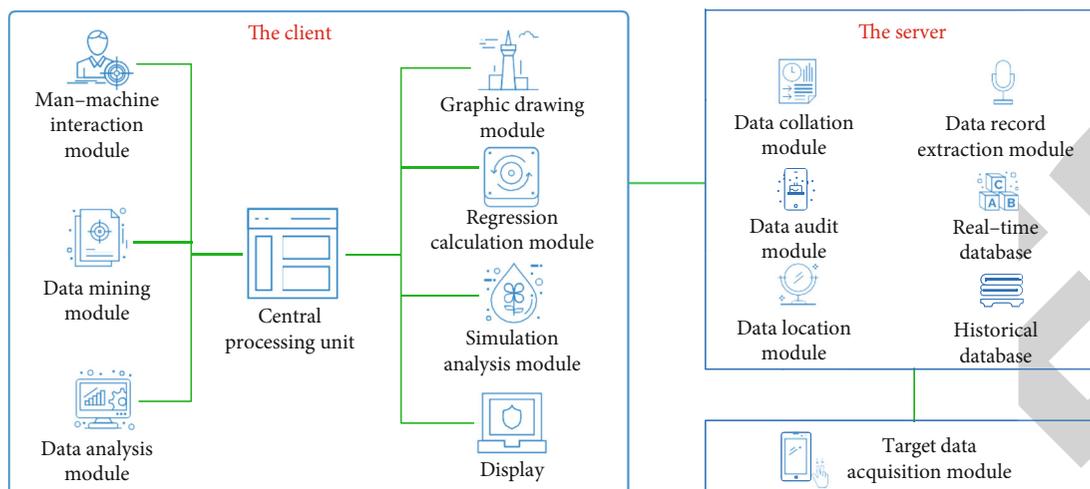


FIGURE 1: A data processing system of Forest Ecological Station Based on Internet of Things technology.

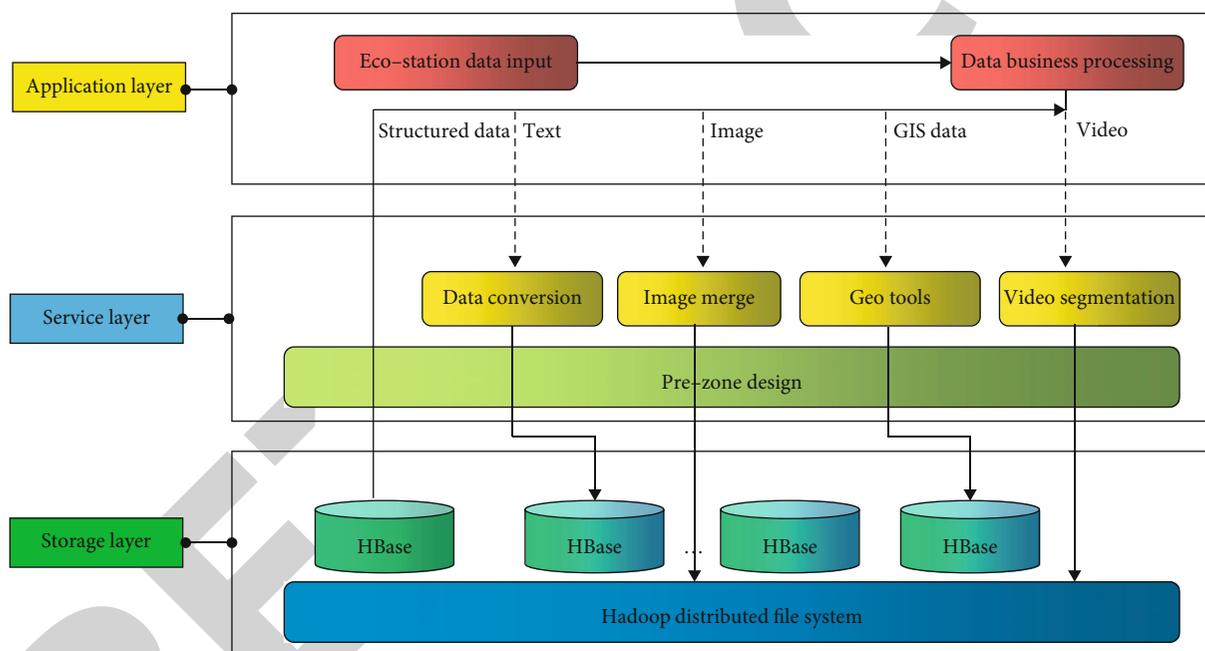


FIGURE 2: Overall architecture.

shown in Figure 2, mainly including the application layer, service layer, and storage layer.

The storage layer is the most important part of the big data model of forest ecology, and its main function is to use it for persistent storage. This layer includes distributed HDFS and the column-oriented HBase database. HDFS stores video, image, and other unstructured data in forest ecological data; the HBase database is used to store structured data generated in forest ecology [13, 14].

The service layer mainly includes data conversion, image merging, GeoTools, video segmentation, prepartition design, and HBase secondary index [15]. Data conversion is mainly because the access data is complex and massive. At the same time, the models and types of sensors are different, and the

data transmission format is not unified. Using the data conversion module to continue to unify the data is conducive to the later expansion and performance enhancement of the system. The image merging module mainly solves the problem of insufficient performance when Hadoop processes massive small files. Small files are merged based on data site and time correlation algorithm and stored on HDFS after reaching the threshold. GeoTools mainly analyzes GIS data and stores it into the HBase database. Video segmentation is to segment large files according to the size of the HDFS data block and store them directly on HDFS. the HBase secondary index module mainly solves the problem of failure of the HBase data index outside the query primary key and can realize efficient data retrieval [16].

The application layer is mainly for all kinds of users to conduct unified data processing for various businesses in the forest ecological big data platform. Users can query, analyze, manage, and download forest ecological data.

*2.2. Platform Business Design Process.* At present, the data collected by forest ecological sites are mainly divided into five categories: image, video, GIS data, unstructured data of text data, and structured data. Text data mainly includes Excel files and txt files. The business processing module needs a unified data access interface to judge the data types to be stored and adopts different storage strategies for different data types.

When storing images, the images in ecological monitoring are mainly massive small files (each image is usually within a few megabytes), while the default capacity of the HDFS data block is 128 MB. When an image is stored in a data block, although it will not occupy the whole data block, and a large number of small files will not put pressure on the hard disk storage, it will increase the memory consumption of the NameNode in HDFS, and reading small-size images will waste a lot of time [17]. In general, the metadata capacity of NameNode is 250 B. by default, the metadata capacity of two new replicas is 368 B. When a small image is stored in three copies on HDFS, NameNode's memory consumption  $M$  increases as  $n$  increases, as shown in the following equation.

$$M = a + 250n + (368 + b) \sum \frac{F}{B}, \quad (1)$$

where  $a$  is the memory capacity occupied by the NameNode when there is no data in HDFS,  $b$  is the memory consumption of each data block in the NameNode,  $B$  is the capacity of the HDFS data block, and  $F$  is the memory capacity of  $n$  images stored in HDFS.

Therefore, when the data is an image, the strategy of merging and storing the image based on data site and time correlation is adopted; that is, first write the image to an image queue, judge whether the value of the image queue is greater than the capacity of a block, and continue to write if it is not enough. If it is larger than the capacity of a block, it will be directly stored in HDFS and indexed, and the image meta information will be stored in HBase. When the written data is a video, first judge whether it exceeds the capacity of a block. If not, it will be processed directly according to the capacity of a block. If it is greater than the capacity of a block, it will be segmented directly according to the HDFS blocking strategy, an index will be established, and the video metadata information will be stored in HBase [18]. When the written data is GIS data, it is analyzed via GeoTools tool and stored in the HBase database. When the written data is text data, the text data is transformed into structured data through corresponding service layer analysis and stored in the HBase database. When the written data is structured data, the data is directly stored in the HBase database.

*2.3. Prezoning Design.* By default, when HBase creates a data table, it will create a region without start and end, and the

TABLE 1: Server configuration.

Equipment	Parameter
CPU	Inter (R) Xeon (R) Silver 4110 CPU @ 2.10 GHz
Memory capacity (CB)	32 (server configuration)
Hard disk capacity (CB)	500
Network type	Gigabit LAN
Operating system	Centos Linux server 7.4
Hadoop	3.1.2
HBase	2.1.0
Elasticsearch	6.7.0
Zookeeper	3.4.10
Solr	6.0.0
Phoenix	5.0.0

data will be written to the region in ascending order according to the dictionary of key value pairs. If the region of HBase reaches the threshold, it will frequently trigger split operation, which will cause hot spot tilt, and the value range is 0-60. Assuming that it is to be divided into  $k$  partitions ( $k$  is an integer), start the data SplitKey in the range of 0-60 from 1 and preliminarily prepartition according to the HBase prepartition algorithm. Finally, the SplitKey is obtained according to the prepartition algorithm, and the prepartition table is created to avoid hot spot tilt [19, 20].

*2.4. RowKey Design.* The HBase database is mainly composed of a row key, column family, column family qualifier, and timestamp. On the premise of meeting the length principle, hash principle, and uniqueness principle, the line key can improve memory utilization. Due to the frequent use of an ecological site query, this field is added in the design. At the same time, in order to record the data generation time and data version control, the time is also placed in the primary key RowKey. Therefore, the RowKey format designed by this system is as follows: site + time.

Forest ecological structured data is divided into 4 column families according to its characteristics, which are divided into the soil column family, meteorological column family, biodiversity column family, and hydrological column family, including 742 elements, such as temperature, relative humidity, wind speed, and precipitation. The identification ID of a site at a specific time stores a RowKey. Each RowKey will have multiple column qualifiers, representing the element values at different times [21].

*2.5. Image Merging Index Algorithm.* Since most of the images stored in the ecological station are small images, there will be a NameNode bottleneck and low retrieval performance. The existing algorithms are generally used to solve the problem of high memory occupation by merging small files into large files and then storing them into large files but do not take into account the problems between images. For example, taking an ecological feature data image may be scattered in different large files, resulting in low storage efficiency. Therefore, this paper proposes a consolidated

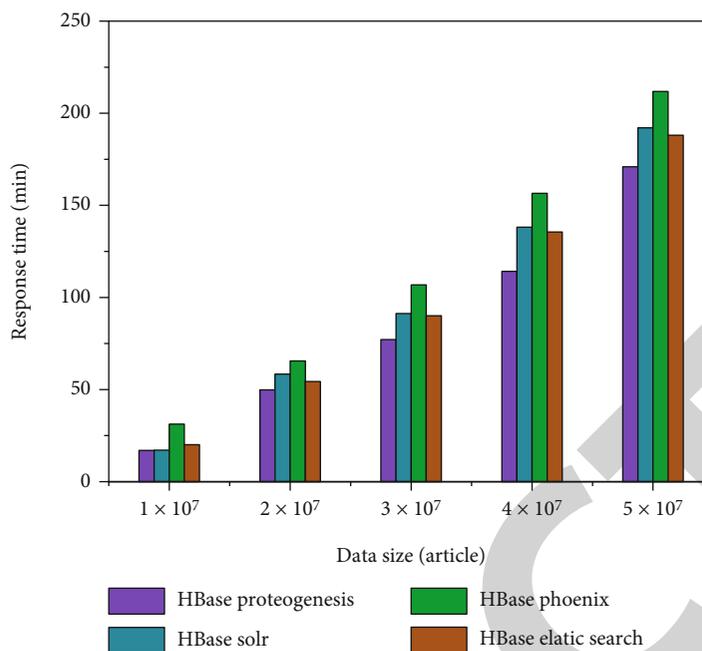


FIGURE 3: Comparison of response time for inserting data.

storage method RBSTC based on data site and time correlation. First, create a temporary queue and store the initial image. Then, judge whether the images to be stored come from the same site and on the same day. If so, merge the stored images into the temporary queue. Otherwise, create a new queue, and the images to be stored will be used as the initial images in the new queue. Repeat the above operations until all images to be uploaded are transmitted to HDFS.

**2.6. GIS Data Storage Design.** GeoTools is a GIS toolkit developed in Java language. Based on the standard GIS interface, it supports the access of multiple GIS data sources. GIS data is generally composed of coordinate data, attribute data, and topological relationship data. According to the characteristics of vector data, a vector data storage model suitable for HBase is designed. In the design of GIS data RowKey, RowKey is site + timestamp, which is divided into three column families: the spatial information column family, attribute information column family, and topological information column family [22, 23].

### 3. Result Analysis

**3.1. System Insertion Performance Comparison.** In order to evaluate the performance of the technical scheme in this paper, the performance of the storage model, prepartition, RowKey design, secondary retrieval scheme, and image merging strategy is tested based on the data of different data levels. This paper configures the server-related environment and builds the Hadoop cluster, HBase cluster, and Zookeeper cluster. In order to conduct the secondary index comparison experiment, the Elasticsearch cluster, Solr cluster,

and Phoenix cluster are also built. The server configuration is shown in Table 1.

In the experiment, four clients simultaneously insert data into the HBase table, and we count the put time of each 107 data on the four clients. After repeating the experiment for 10 times, take the average value and test the put time of the HBase native, Elasticsearch-based index, Solr index, and Phoenix index under the same conditions. The results are shown in Figure 3.

It can be seen from Figure 3 that put is the most efficient way to build indexes without additional resource allocation. It can be seen that when the same 10 pieces of data are added, the insertion time is longer and longer, because with the increase of the amount of data, there are more and more index data, making it difficult to insert the index. At the same time, because the Phoenix bottom layer needs to build an appropriate storage index structure in the coprocessor, it consumes additional computing resources. Based on Elasticsearch and Solr, you only need to build indexes in your own cluster without additional computing resources. Therefore, the Phoenix secondary index has the greatest loss on insertion performance [24, 25].

**3.2. Comparison of Single Condition Query Performance of Different Secondary Indexes.** The retrieval data still adopts the above data, and its retrieval performance comparison is shown in Figure 4.

It can be seen from Figure 4 that the response speed of the native HBase decreases obviously. When the amount of data reaches  $1 \times 10^8$ , the response time is greater than 5 s. Theoretically, HBase is a column-oriented database. Its bottom layer is to establish a B+ tree index based on RowKey, which can retrieve data efficiently, but the system will scan the whole table corresponding to the index of the nonrow

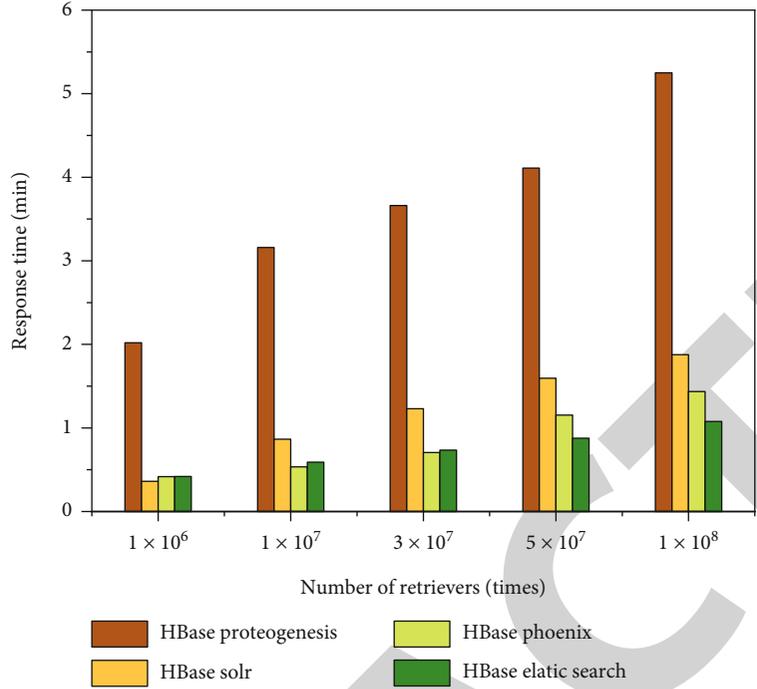


FIGURE 4: Comparison of retrieval performance.

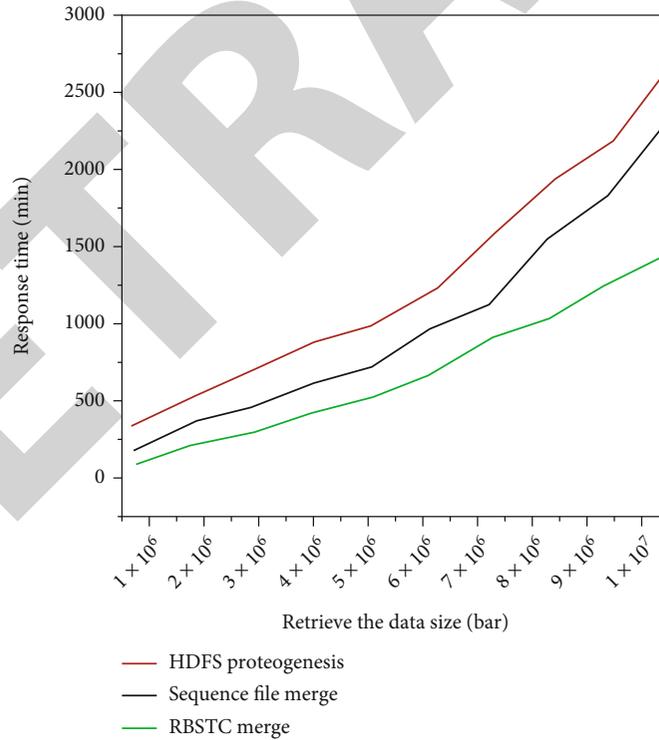


FIGURE 5: Comparison of reading and writing performance.

key, resulting in low efficiency of overall data retrieval. For the comparison of secondary indexes, the bottom layers of Solr and Elasticsearch are based on Lucene, but the framework design of Elasticsearch is further optimized, and the data retrieval efficiency also has better performance. There-

fore, when the amount of data reaches  $1 \times 10^8$ , the efficiency of Elasticsearch is 1.72 times that of Solr. The retrieval efficiency of Phoenix is close to that of Elasticsearch, but Phoenix has strong coupling, so Elasticsearch is finally selected as the secondary index.

TABLE 2: Concurrent stress test.

Parameter	HBase	HBase secondary index
Queries per second	3475	6520
Throughput per second (MB)	42	73
Response time (ms)	852	260

**3.3. Performance Verification of Unstructured Data Storage Scheme.** In the data storage verification part, the performance of the image merging strategy is verified; that is, the default sequence file merging of HDFS is verified and compared with the RBSTC algorithm used in this experiment. A total of  $10^7$  images in the eco station system are used, occupying 100~500 kb of storage space. The images are tested, respectively, via HDFS default sequence file merging and RBSTC merging index. 10 read-write experiments are carried out, and the average value of 10 reads and writes is taken as the final time consumption. The average read-write verification is shown in Figure 5.

As can be seen from Figure 5, with the increasing scale of image data, the advantages of the merging method based on data site and time correlation become more and more obvious. When the number of images is  $10^7$ , the merging of small files based on data site and time correlation is 1.79 times that of original reading and writing and 1.15 times that of the sequence file. Therefore, the merging strategy based on data site and time correlation is more suitable for the storage scenario of massive images in ecological stations.

**3.4. System Pressure Test.** In order to verify the stability of the system, the postman tool is used to conduct pressure test on the big data platform. The query rate per second (QPS), throughput (TPS), and response time (RT) are selected as the main parameters. The concurrency is  $1 \times 10^4$  times and the test time is 3 min. The average value of 10 experiments is taken. The concurrency test is shown in Table 2.

Through the test results, it can be found that in the case of  $10^4$  concurrent users, the number of queries per second after optimization is 1.88 times that of the original, the throughput per second is 1.74 times that of before optimization, and the system response time is 69.5% lower than that before optimization, indicating that the system can also operate stably under the condition of high concurrency.

## 4. Conclusion

The subject of this study involves relevant knowledge in the fields of forestry and ecology. In the process of understanding and studying this subject, I have carefully studied the basic knowledge in relevant fields. In the process of developing the data management platform of forest ecological monitoring network, it not only deepened the understanding of ecology but also had a certain research conclusion on the significance of building a forest ecological monitoring network. This paper designs the big data storage framework of the Forest Ecological Station Based on Hadoop and HBase. The results are as follows:

- (1) Facing the demand of massive ecological data storage and rapid retrieval, the traditional model architecture cannot guarantee the data processing performance of the ecological data platform. The distributed big data technology is used to build the ecological big data platform. Through the scientific design of RowKey, Hadoop and HBase are used as the data storage layer to realize the storage of massive data. A prepartition algorithm is proposed and designed to ensure consistent data distribution and avoid the problem of hot data skew
- (2) According to the storage requirements of massive image data, an association merging storage method based on data site and time correlation is proposed. When the unstructured data is  $10^7$ , the packaging small image strategy based on data site and time correlation is 1.15 times the merging efficiency based on SequenceFile and 1.79 times that of native HBase

The research and development of the forest ecological monitoring network data management platform is still in its infancy. Although the analysis and processing of forest ecological data is currently written, there are still many areas that need to be improved and further studied. In terms of data cleaning, although outliers have been removed from the granularity of months and years, the cleaning efforts are still not enough. It is still necessary to learn more about the field of forest ecology and be able to determine different types of indicators for different ecosystems. For its numerical range, we optimize the calculation method proposed for outliers; in terms of data filling, it is necessary to understand more models and methods in the field, and through the comparison of different filling methods, the data can be filled more accurately.

## Data Availability

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

## Conflicts of Interest

The author declares that they have no conflicts of interest.

## References

- [1] M. E. Spicer, H. Radhamoni, M. C. Duguid, S. A. Queenborough, and L. S. Comita, "Herbaceous plant diversity in forest ecosystems: patterns, mechanisms, and threats," *Plant Ecology*, vol. 223, no. 2, pp. 117–129, 2022.
- [2] A. Reuber, C. Flávio, L. Maurice, and D. Wesley, "Discovery-defense strategy as a mechanism of social foraging of ants in tropical rainforest canopies," *Behavioral Ecology*, vol. 91, no. 5, pp. 5–11, 2021.
- [3] S. Estrada-Villegas, J. S. Hall, M. Breugel, and S. A. Schnitzer, "Lianas reduce biomass accumulation in early successional tropical forests," *Ecology*, vol. 101, no. 5, pp. 7–10, 2020.
- [4] X. Tong, M. Brandt, Y. Yue, P. Ciaias, and R. Fensholt, "Forest management in southern China generates short term extensive

- carbon sequestration,” *Nature Communications*, vol. 11, no. 1, pp. 129–132, 2020.
- [5] J. L. R. D. Lima, K. Jobim, J. G. Jardim, and B. T. Goto, “Highlighting the species richness of arbuscular mycorrhizal fungi (Glomeromycota) in the Brazilian semi-arid,” *Nova Hedwigia*, vol. 114, no. 1-2, pp. 107–139, 2021.
- [6] M. Villagra, C. P. Trentini, D. D. Francescantonio et al., “Disturbance as a driver of trait assembly in liana communities in a semi-deciduous Atlantic forest,” *Plant Ecology*, vol. 222, no. 7, pp. 773–790, 2021.
- [7] L. Motta, A. Vitali, G. C. Amico, D. García, and M. A. Rodriguez-Cabal, “Post-dispersal seed predation in Patagonia temperate forest depends on habitat patchiness and seed species,” *Plant Ecology*, vol. 222, no. 7, pp. 819–827, 2021.
- [8] H. Xiao, S. Ali, Z. Zhang, M. S. Sarfraz, and M. Faisal, “Big data, extracting insights, comprehension, and analytics in cardiology: an overview,” *Journal of Healthcare Engineering*, vol. 2021, Article ID 6635463, 14 pages, 2021.
- [9] J. Wang and Y. Gao, “Suspect multifocus image fusion based on sparse denoising autoencoder neural network for police multimodal big data analysis,” *Scientific Programming*, vol. 2021, Article ID 6614873, 12 pages, 2021.
- [10] P. Ni, B. Liu, and G. He, “An online optimization strategy for a fluid catalytic cracking process using a case-based reasoning method based on big data technology,” *RSC Advances*, vol. 11, no. 46, pp. 28557–28564, 2021.
- [11] I. Awajan, M. Mohamad, and A. Al-Quran, “Sentiment analysis technique and neutrosophic set theory for mining and ranking big data from online reviews,” *IEEE Access*, vol. 9, no. 99, pp. 47338–47353, 2021.
- [12] J. Guo, “Deep learning approach to text analysis for human emotion detection from big data,” *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 113–126, 2022.
- [13] F. Tian, “Immersive 5G virtual reality visualization display system based on big-data digital city technology,” *Mathematical Problems in Engineering*, vol. 2021, Article ID 6627631, 9 pages, 2021.
- [14] B. Khan, R. Naseem, M. A. Shah, K. Wakil, and M. Mahmoud, “Software defect prediction for healthcare big data: an empirical evaluation of machine learning techniques,” *Journal of Healthcare Engineering*, vol. 2021, Article ID 8899263, 16 pages, 2021.
- [15] B. A. Jnr, S. A. Petersen, D. Ahlers, and J. Krogstie, “Big data driven multi-tier architecture for electric mobility as a service in smart cities,” *International Journal of Energy Sector Management*, vol. 14, no. 5, pp. 1023–1047, 2020.
- [16] G. Liang, Y. Wang, Z. Huang, F. Mo, and C. Zhi, “Initiating hexagonal MoO<sub>3</sub> for superb-stable and fast NH<sub>4</sub><sup>+</sup> storage based on hydrogen bond chemistry,” *Advanced Materials*, vol. 32, article 1907802, 2020.
- [17] S. Xiao, Z. Li, J. Liu, Y. Song, and Q. Yan, “Se—C bonding promoting fast and durable Na<sup>+</sup> storage in yolk-shell SnSe<sub>2</sub>@Se—C,” vol. 16, no. 41, Article ID e2002486, 2020.
- [18] H. Duan, Z. Hu, and Y. Song, “Bidding strategies in energy and reserve markets for an aggregator of multiple EV fast charging stations with battery storage,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 471–482, 2021.
- [19] H. Sun, L. Chen, X. Hao, C. Liu, and M. Ni, “An energy-efficient and fast scheme for hybrid storage class memory in an AIoT terminal system,” *Electronics*, vol. 9, no. 6, pp. 1013–1017, 2020.
- [20] H. Song, J. Li, M. Luo, Q. Zhao, and F. Liu, “Ultra-thin mesoporous LiV<sub>3</sub>O<sub>8</sub> nanosheet with exceptionally large specific area for fast and reversible Li storage in lithium-ion battery cathode,” *Journal of the Electrochemical Society*, vol. 168, no. 5, article 050515, 2021.
- [21] Q. Zhang, “Relay vibration protection simulation experimental platform based on signal reconstruction of MATLAB software,” *Nonlinear Engineering*, vol. 10, no. 1, pp. 461–468, 2021.
- [22] R. Huang, S. Zhang, W. Zhang, and X. Yang, “Progress of zinc oxide-based nanocomposites in the textile industry,” *IET Collaborative Intelligent Manufacturing*, vol. 3, no. 3, pp. 281–289, 2021.
- [23] L. Xin, M. Chengyu, and Y. Chongyang, “Power station flue gas desulfurization system based on automatic online monitoring platform,” *Journal of Digital Information Management*, vol. 13, no. 6, pp. 480–488, 2015.
- [24] N. Balakrishnan, A. Rajendran, and P. Ajay, “Deep embedded median clustering for routing misbehaviour and attacks detection in ad-hoc networks,” *Ad Hoc Networks*, vol. 126, article 102757, 2021.
- [25] M. Fan and A. Sharma, “Design and implementation of construction cost prediction model based on SVM and LSSVM in industries 4.0,” *International Journal of Intelligent Computing and Cybernetics*, vol. 14, no. 2, pp. 145–157, 2021.