



OPEN ACCESS

EDITED BY
Hui-Jia Li,
Nankai University, China

REVIEWED BY
Chaima Aouiche,
University of Tébessa, Algeria
Chaker Abdelaziz Kerrache,
University of Ghardaia, Algeria

*CORRESPONDENCE
Maojin Sun,
✉ sunmaojin@ldy.edu.rs

RECEIVED 21 August 2024
ACCEPTED 20 September 2024
PUBLISHED 24 October 2024

CITATION
Sun M and Sun L (2024) Optimization of artificial
intelligence in localized big data real-time query
processing task scheduling algorithm.
Front. Phys. 12:1484115.
doi: 10.3389/fphy.2024.1484115

COPYRIGHT
© 2024 Sun and Sun. This is an open-access
article distributed under the terms of the
[Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/).
The use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in this
journal is cited, in accordance with accepted
academic practice. No use, distribution or
reproduction is permitted which does not
comply with these terms.

Optimization of artificial intelligence in localized big data real-time query processing task scheduling algorithm

Maojin Sun* and Luyi Sun

CEICloud Data Storage Technology (Beijing) Co., Ltd., Beijing, China

Introduction: The development of science and technology has driven rapid changes in the social environment, especially the rise of the big data environment, which has greatly increased the speed at which people obtain information. However, in the process of big data processing, the allocation of information resources is often unreasonable, leading to a decrease in efficiency. Therefore, optimizing task scheduling algorithms has become an urgent problem to be solved.

Methods: The study optimized task scheduling algorithms using artificial intelligence (AI) methods. A task scheduling algorithm optimization model was designed using support vector machine (SVM) and K-nearest neighbor (KNN) combined with fuzzy comprehensive evaluation. In this process, the performance differences of different nodes were considered to improve the rationality of resource allocation.

Results and Discussion: By comparing the task processing time before and after optimization with the total cost, the results showed that the optimized model significantly reduced task processing time and total cost. The maximum reduction in task processing time is 2935 milliseconds. In addition, the analysis of query time before and after optimization shows that the query time of the optimized model has also been reduced. The experimental results demonstrate that the proposed optimization model is practical in handling task scheduling problems and provides an effective solution for resource management in big data environments. This research not only improves the efficiency of task processing, but also provides new ideas for optimizing future scheduling algorithms.

KEYWORDS

task scheduling algorithm, artificial intelligence (AI), support vector machines (SVM), big data, optimization model

1 Introduction

With the rapid development and change of the scientific environment, information technology has also been greatly improved, which has brought about the explosive growth of information data, and the surge in data volume has also led to the emergence of various data analysis methods [1]. In the era of big data, the information is mixed, the sources of data information are extensive, and there are many types of information. These large-capacity data information also has a lot of invalid information when used, which leads to a tedious process of finding target information and a long calculation time when mining big data.

Therefore, how to improve the user's processing accuracy when searching for big data, that is, how to schedule tasks in a timely and efficient manner, has become a research hotspot for researchers [2]. In particular, the scale of today's Internet users is gradually increasing, and the number of users who use such tools is also increasing. At the same time, the increase of users means that various demands are also constantly changing, which also makes the system have higher requirements for various performances in terms of task completion. For example, the user experience and cost consumption of the user are all directions that need to be paid attention to at present.

Under the background of the era of big data, many information resources are connected through the network, and the way of user information search has become simple. However, with the continuous development of society, various products are no longer just aiming to meet basic needs, but begin to pursue the quality of services. The same is true in the field of big data [3]. Many researchers have begun to consider how to make the search method simpler and faster, and the focus of completing tasks is to allocate and schedule tasks. Therefore, reasonable optimization of task scheduling can make information search tasks more convenient and accurate. This paper optimizes task scheduling through artificial intelligence (AI) technology. It provides direction for the improvement of task scheduling algorithm, and also provides theoretical reference for the optimization of task scheduling algorithm, which has certain theoretical significance.

AI technology has become a hot topic in recent years. Many disciplines have conducted theoretical and practical research on AI technology, which has confirmed its effectiveness in data analysis [4]. This paper uses AI technology to optimize and analyze the task scheduling algorithm for localized big data real-time query processing, and confirms the feasibility of applying AI technology to task scheduling algorithms through experiments. The AI technology developed from the field of computer science is applied to the task scheduling algorithm processed by the computer system, and its effects are verified from different angles by designing experiments. This is the innovation of this paper. The contribution of this study lies in the combination of AI technology and task scheduling optimization, proposing Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and fuzzy comprehensive evaluation methods, which are conducive to improving the intelligence and adaptability of algorithms, and promoting the application of big data technology in social and economic development.

2 Related work

The processing of big data has high requirements on the system, so many researchers have studied the scheduling of task processing. Gruzlikov A. M. conducted research on scheduling methods in the computing process of distributed real-time systems [5]. Kaur A. conducted related research on cloud computing, and believed that the cost of data analysis and execution was relatively high. Therefore, a variety of task scheduling algorithms had been studied to solve such problems [6]. Lu P. conducted optimization research on the scheduling method in grid data transmission, and further solved the problems existing in the optimization through experiments [7].

Sarathambekai S. optimized the task scheduling of the computing operation process in the distributed system through the method of particle swarm optimization [8]. Panda S. K. optimized the scheduling method based on various methods through the analysis of cloud scheduling work, and designed experiments to compare and analyze the optimization effect under different methods [9]. At present, there are many researches on scheduling algorithms in the field of cloud computing, but less research on big data, especially localized big data. Therefore, it is necessary to optimize and analyze task scheduling algorithms from the perspective of localized big data.

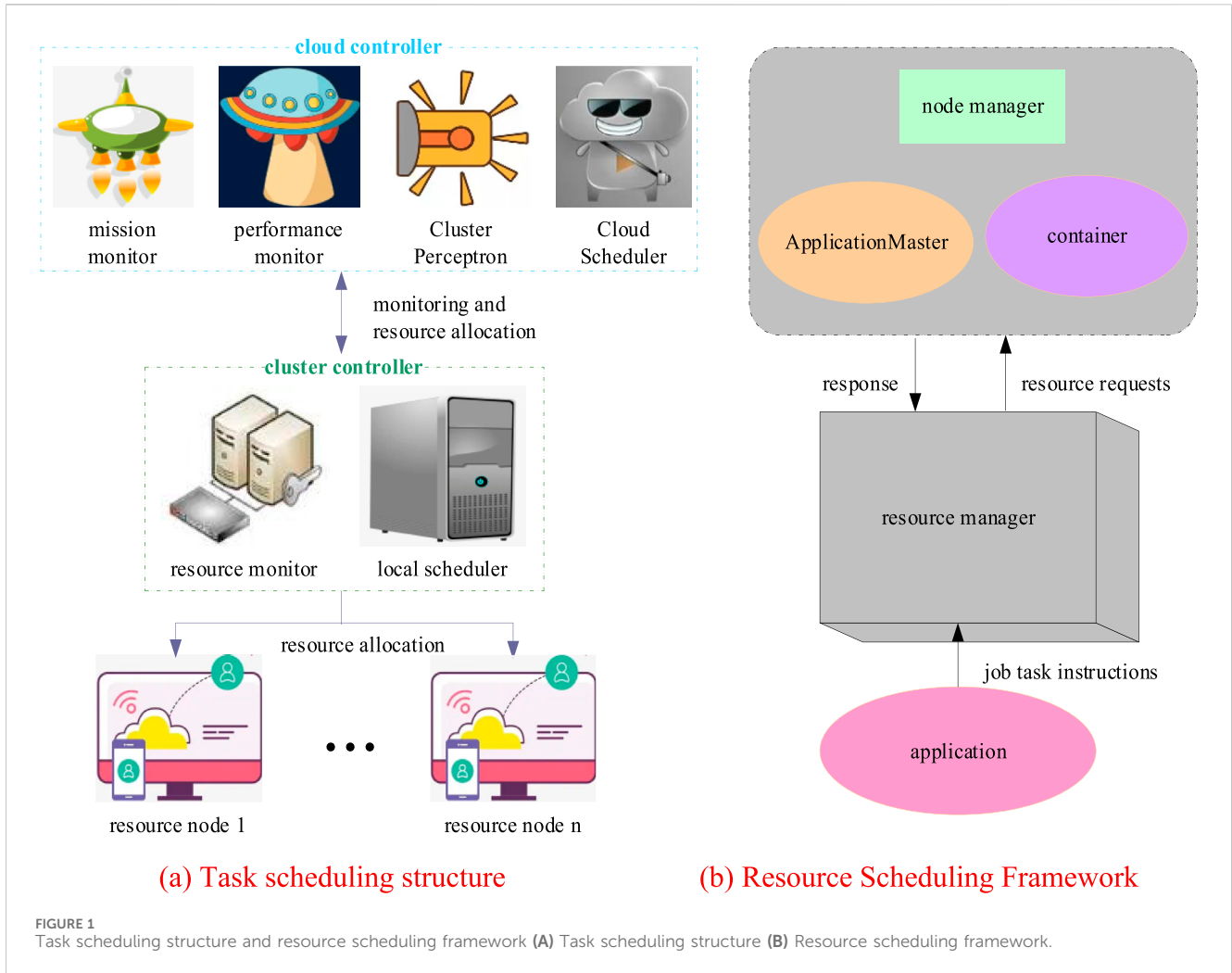
The rapid development of computer science has made AI technology increasingly multiplied, and there are many research areas currently used in AI technology. Rongpeng, Li studied the relationship between the 5G field and AI technology, and believed that the lack of complete AI functions in the cellular network in 5G was difficult to meet complex configuration problems and new business requirements [10]. Labovitz D. L. applied AI technology in the monitoring of mixed medication and intake [11]. Youssef A. briefly summarized the effective application of AI technology in photovoltaic systems, and believed that AI technology had made a great contribution in this field [12]. Syam N. applied AI technology to the field of marketing, and studied the impact of using AI technology in sales management by observing the sales practice process [13]. Ehteram M. used AI technology for the operation of reservoirs and power plants [14]. AI is used in a wide range of fields, but the use of AI technology in big data is mostly used in result classification. There is less research on the task work of real-time systems.

In summary, the current research on scheduling algorithms in the field of cloud computing is relatively extensive, involving multiple aspects such as distributed systems, grid data transmission, and particle swarm optimization. However, there is a clear research gap in task scheduling algorithms for handling localized big data using existing methods. Most existing research focuses on result classification and task scheduling optimization in cloud computing environments, with relatively insufficient in-depth exploration of the specific challenges and requirements of localized big data. To this end, an AI-based approach was adopted to optimize the task scheduling algorithm. A task scheduling algorithm optimization model based on SVM, KNN, and fuzzy comprehensive evaluation was designed to effectively meet the localization requirements in big data processing.

3 Overview of artificial intelligence and task scheduling algorithms

3.1 Localized big data real-time query processing task scheduling algorithm

Data localization means that both the data and the calculation process are carried out on the same node. More precisely, it is the execution process of the calculation in the node where the data itself is located, that is, the calculation operation is moved without changing the position of the data [15]. Real-time query is the process of executing computing tasks within a certain period of time and transmitting the results of task processing to the outside



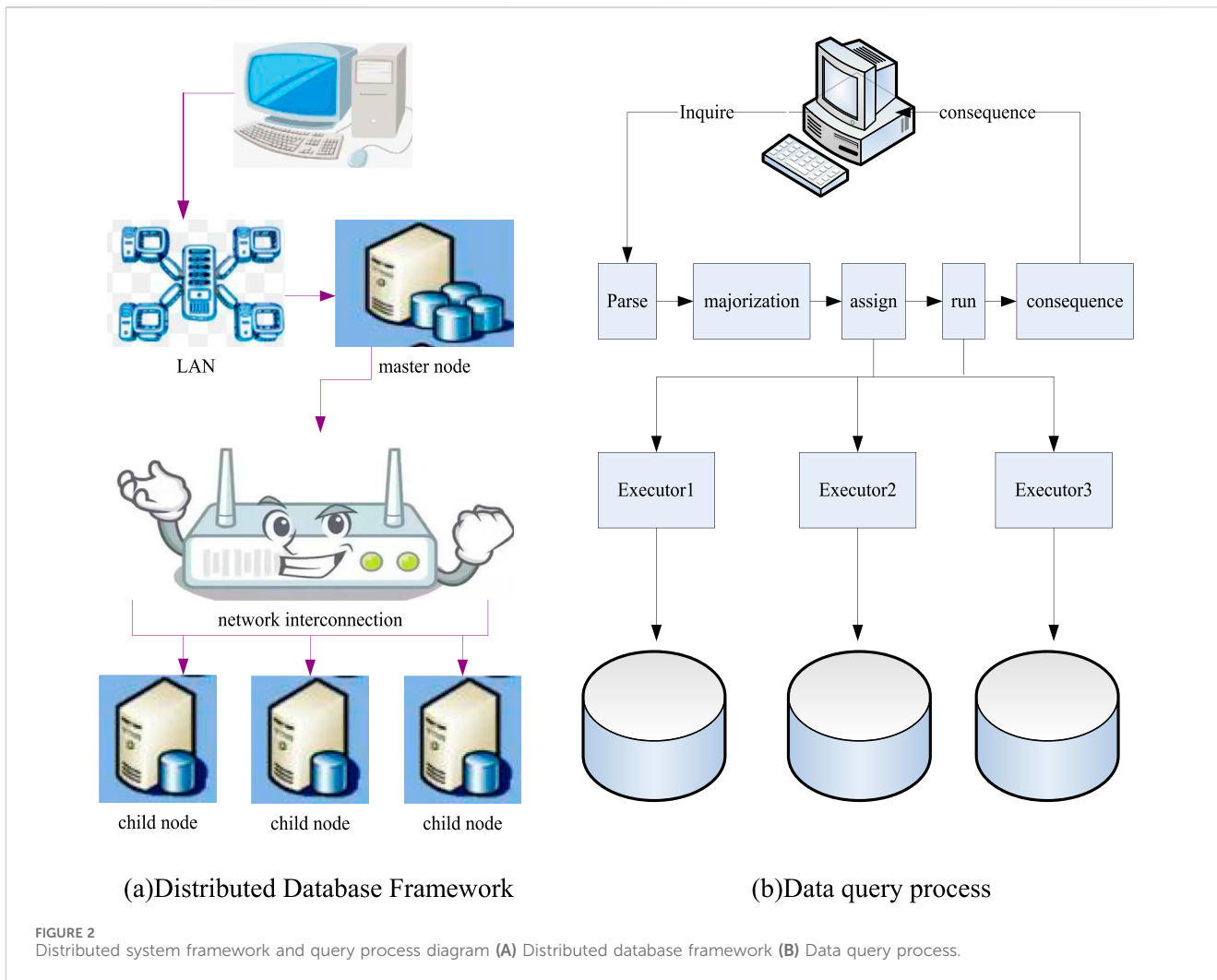
world. In this process, the calculation of the system involves the assignment of tasks, which is related to task scheduling. Task scheduling is to allocate resources reasonably according to the actual task processing attributes and resource conditions during task processing of data, and distribute them to big data tasks for data execution. In this process, how to carry out scientific and reasonable resource allocation is the key [16, 17]. The structure of task scheduling and the basic framework of resource scheduling are shown in Figure 1.

In Figure 1, in the task scheduling structure, tasks and resources are monitored by the controller and the cluster controller, and resources are allocated through the cluster controller. In a big data processing system, there are multiple cluster controllers and resource nodes. In task scheduling, resource scheduling is extremely important, and resource scheduling is to allocate resources through the resource scheduler [18]. The application program sends job task instructions to the resource scheduling manager, the node manager and resource nodes send resource requests to the resource scheduling manager, and the resource scheduling manager responds to the request [19].

The real-time query of localized big data also depends on the construction of the system database. In the development of big data, centralized databases and distributed databases have emerged. In

real-time query tasks, distributed database systems are more suitable than centralized database systems. This is because the data nodes in the distributed database are different. Although these data logically belong to the same system, each kind of data has its own site [20]. This way of data storage makes the data independent. Relatively speaking, the data information is more transparent and easier to query [21]. In the process of task processing, task scheduling is a key factor affecting the maximization of task completion efficiency. The framework and query process of a distributed database system are shown in Figure 2.

As shown in Figure 2, the distributed database system first selects a master node to schedule the system. Moreover, this work is carried out in the form of parallelization. That is to say, multiple nodes can be scheduled at the same time. A complete query process is firstly that the master node receives the request instructions sent by the user, and parses the instructions through related programs. Secondly, through computational analysis, the analysis is optimized and selected, and an optimal plan for the instruction query is obtained. Thirdly, according to the obtained query, the most planned task is scheduled and allocated, and the result of the scheduling and allocation is sent to the child nodes. The child nodes perform the query operation and feed back the running results to the master node. The master node receives the calculation results



after each node executes the command, summarizes the results and then feeds them back to the client [22, 23].

Commonly used scheduling tools include CRONTAB, AZKABAN, Oozie and so on. Among them, Oozie is a scheduling tool which comes with Hadoop. Its operating architecture and workflow are shown in Figure 3.

As shown in Figure 3, Oozie workflow first determines the nature of the task, and then performs attribute processing on the task. Finally, it is scheduled according to the task properties. In this process, it is mainly to complete the control of the process and the definition of nodes. The specific operation is to control workflow nodes through a Directed Acyclic Graph (DAG), submit jobs to Hadoop through the Oozie interactor, and finally perform operations based on the tasks that need to be started and executed in the workflow.

3.2 Artificial intelligence

AI technology is a technical science that learns based on human intelligent behavior and theoretical technology through computer systems [24]. The field of AI contains knowledge of multiple disciplines. The application of AI requires practitioners to have a

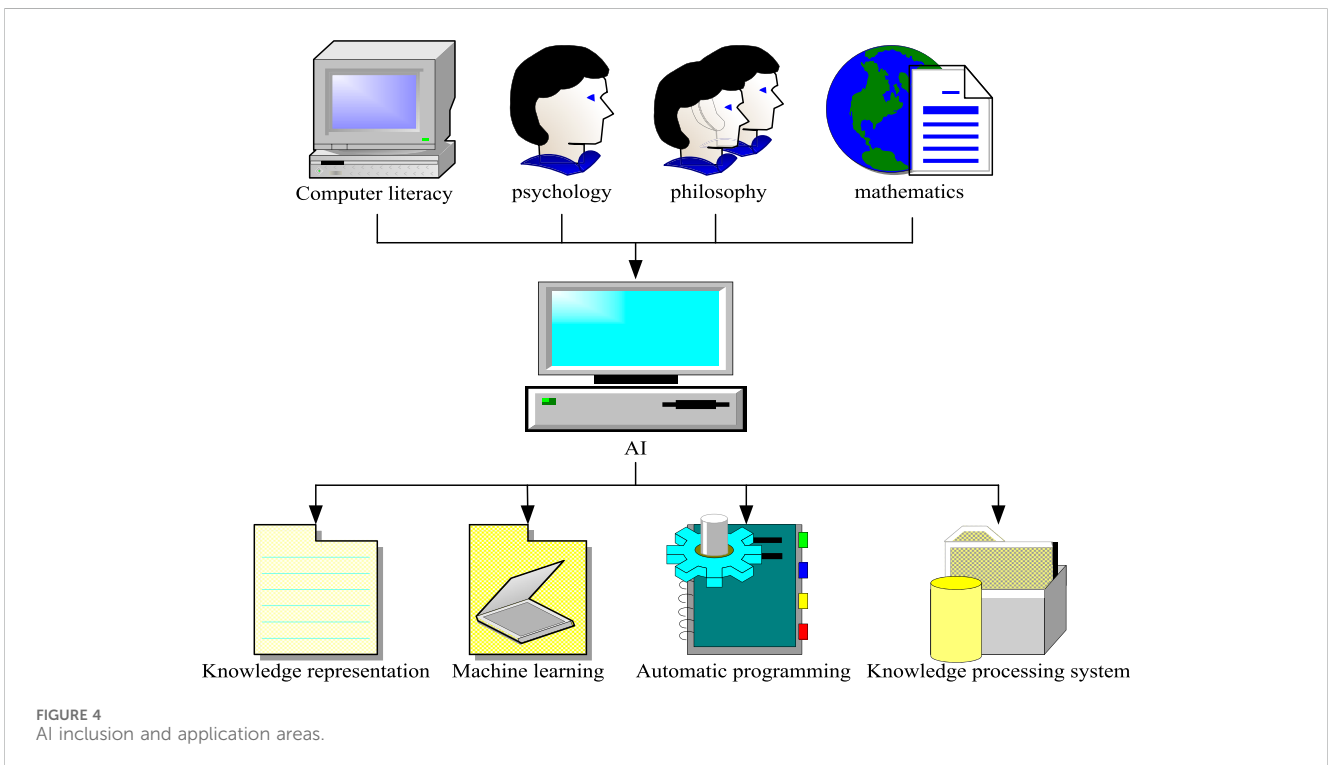
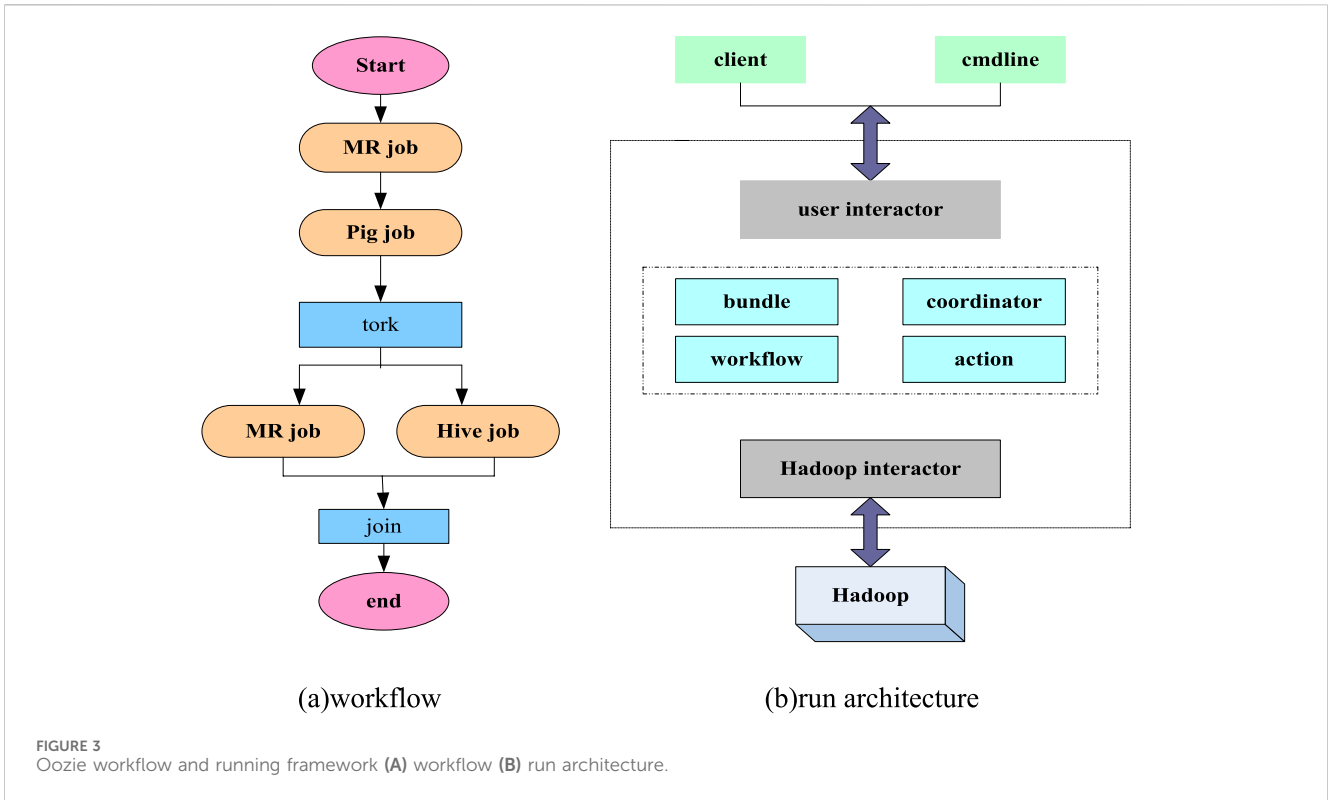
deep understanding of computer knowledge, as well as psychological and philosophical ideas and must be able to use them. The research direction of AI is also relatively extensive, as shown in Figure 4.

As shown in Figure 4, AI includes fields such as computer science, psychology, and philosophy mentioned above, as well as mathematics and other disciplines, while the application areas include knowledge representation and processing systems, machine learning, and automatic programming.

3.2.1 BP neural network

Backpropagation (BP) neural network is a kind of neural network model that can reversely propagate the error. Different from the general neural network, the BP neural network can complete the transmission of data information along the information transmission direction of the training [25]. At the same time, as the information is passed forward and trained, the computed error can be fed back to the previous layer. The BP neural network training process and structural model are shown in Figure 5.

As shown in Figure 5, the BP neural network mainly reduces the error value to the expected error by continuously updating the threshold value of the error value generated between the target output and the actual output, and finally obtains the functional relationship between the input and the output.



In the process of transmitting information along the training direction, it is assumed that the input value of the d -th node of the hidden layer is D_d , and the input signal at this time is $D_d \cdot \omega_{de}$ represents the connection weight between the d -th node and the e -th

node of the hidden layer, then the input value obtained according to the input signal and the connection weight is shown in Equation 1:

$$D_d = \sum_{e=0}^Q \omega_{de} \cdot \epsilon \tag{1}$$

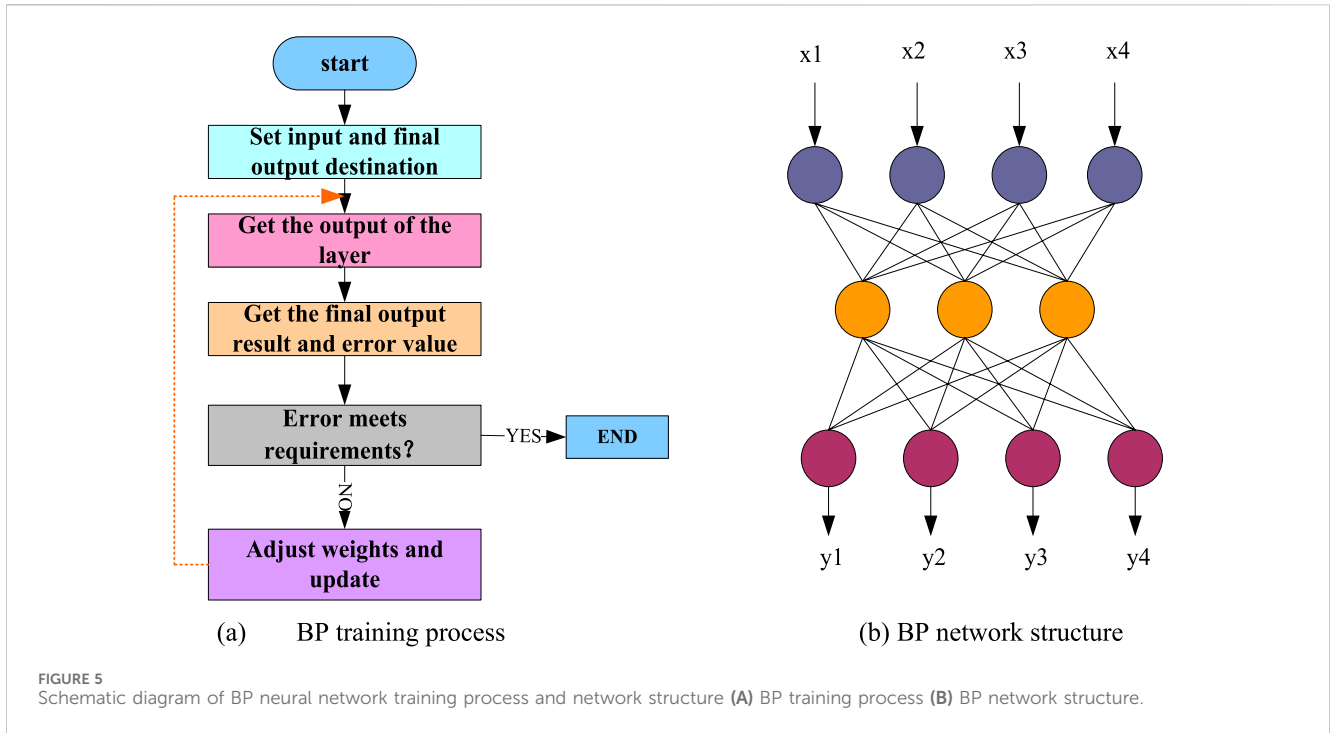


FIGURE 5 Schematic diagram of BP neural network training process and network structure (A) BP training process (B) BP network structure.

The output of the hidden layer obtained from the output is shown in Equation 2:

$$O_d = \alpha_1 \left(\sum_{e=0}^Q \omega_{de} \cdot \varepsilon \right) \quad (2)$$

It is supposed that the input of the f-th node of the output layer is F_f , and the output of this layer is O_f . According to the same calculation principle as before, the calculation formula to obtain this input value is shown in Equation 3:

$$F_f = \sum_{d=0}^T \omega_{fd} \cdot O_d \quad (3)$$

The calculated output value is shown in Equation 4:

$$O_f = \alpha_2 \left(\sum_{d=0}^T \omega_{fd} \cdot O_d \right) \quad (4)$$

In the process of forward transmission of errors in the training process, an appropriate error function is firstly selected for error calculation, and the error values of each training layer are obtained and compared with the expected error values. If it is not within the expected error range, the error value needs to be updated continuously until the expected error is met. In this process, the gradient descent method needs to be used to adjust the error value to minimize the error value. The sample numbers for the samples in the training set are set, which are 1, 2, 3, H, respectively. That is, the total number of samples in the training set is H, and the training process of the training sample numbered H is taken as an example. When H samples enter the training process, first it to set a target value of M_f^H for H. If the actual output obtained by the sample is O_f^H , the calculation error of the error for the H-th sample is shown in Equation 5:

$$e_H = \frac{\sum_{f=1}^H (M_f^H - O_f^H)^2}{2} \quad (5)$$

According to the error of the H-th sample, the error of each sample is calculated by using the same principle, and the total error of the entire training set is obtained as shown in Equation 6:

$$e = \sum_{H=1}^H e_H \quad (6)$$

When using the gradient descent method to adjust and calculate the parameters in the training process, the bias of the output layer is assumed to be δ_f , and the calculation formula of the adjustment amount required for the output layer bias calculated according to the gradient descent method is shown in Equation 7:

$$\Delta \delta_f = -\lambda \frac{\partial e}{\partial \delta_f} \quad (7)$$

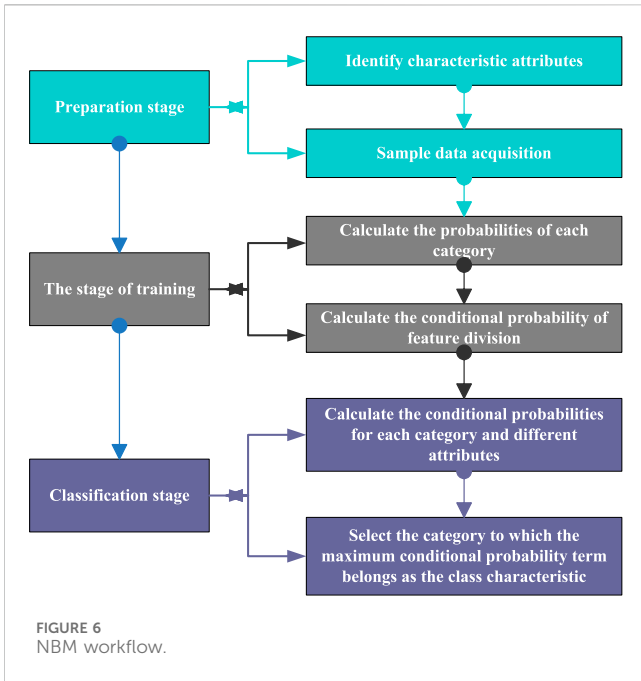
The calculation formula of the adjustment required for the calculated output layer weights is shown in Equation 8:

$$\Delta \omega_{fd} = -\lambda \frac{\partial e}{\partial \omega_{fd}} \quad (8)$$

It can be seen that the adjustment amount is actually seeking a partial derivative.

By continuing to calculate the partial derivative process of the above two equations, the final value of the output layer weight adjustment is obtained as shown in Equation 9:

$$\Delta \omega_{fd} = \lambda \sum_{H=1}^H \sum_{f=1}^H (M_f^H - O_f^H) \cdot \alpha_2'(H_f) \cdot O_d \quad (9)$$



The final value of the output layer bias adjustment obtained by the same calculation principle is shown in Equation 10:

$$\Delta\delta_f = \lambda \sum_{H=1}^H \sum_{f=1}^f (M_f^H - O_f^H) \cdot \alpha_2'(H_f) \quad (10)$$

The weights and biases of the hidden layer also need to be updated during the training process. According to the calculation method of calculating the two variables of the output layer, the final weight update adjustment of the hidden layer is obtained as shown in Equation 11:

$$\Delta\omega_{de} = \lambda \sum_{H=1}^H \sum_{f=1}^f (M_f^H - O_f^H) \cdot \alpha_2'(H_f) \cdot \omega_{fd} \cdot \alpha_1'(D_d) \cdot \varepsilon \quad (11)$$

The resulting hidden layer final bias update adjustment is shown in Equation 12:

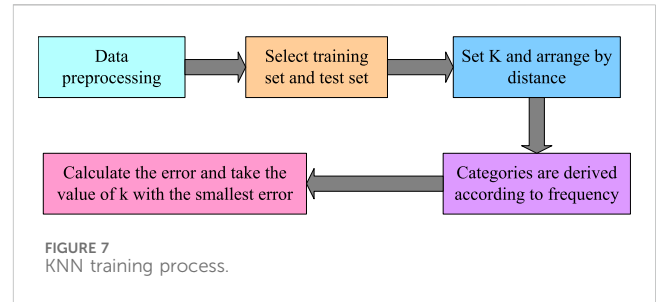
$$\Delta\delta_d = \lambda \sum_{H=1}^H \sum_{f=1}^f (M_f^H - O_f^H) \cdot \alpha_2'(H_f) \cdot \omega_{fd} \cdot \alpha_1'(D_d) \quad (12)$$

3.2.2 Naive Bayes

Naive Bayes (NBM) is a learning method based on probability theory, developed by Bayesian [26]. The premise of this learning method is to ensure that each attribute is independent of each other. The workflow of NBM is shown in Figure 6.

As shown in Figure 6, the NBM training process can be divided into three stages, namely the preparation stage, the training stage and the classification stage. Each stage performs its own duties, and finally outputs the results.

The assumption of Bayesian conditional probability is that there are two conditions C and D. If it needs to calculate the probability that condition C occurs under condition D, the prior probability of the two conditions needs to be calculated. The mathematical expression is shown in Equation 13:



$$p(C|D) = \frac{p(CD)}{p(D)} = \frac{p(D|C) \cdot p(C)}{p(D)} \quad (13)$$

The condition of classification is to find the item with the highest probability and classify the sample into this category is shown in Equation 14:

$$\max(p(C|D)) = \max(p(D|C) \cdot p(C)/p(D)) \quad (14)$$

According to the Bayesian principle, in NBM, it is assumed that an attribute in the sample may be classified as K, and the class in the sample is always M. Then the conditional probability is shown in Equation 15:

$$p(M|K) = \frac{p(M)}{p(K)} \prod_{x=1}^d p(K_x|M) \quad (15)$$

The final classification result is shown in Equation 16:

$$U_{NB}(K) = \operatorname{argmax} p(M) \prod_{x=1}^d p(K_x|M) \quad (16)$$

3.2.3 K-nearest neighbor

KNN simply means that each sample in training can be represented by its K nearest neighbor samples [27]. Its classification process is shown in Figure 7.

As shown in Figure 7, KNN first preprocesses the data. In the preprocessing process, all training instances and parameters in the training process need to be set. Then, the sample set is divided into two groups, namely the training set and the test set. Through the test on the test set, the best K value is obtained, and the sample category is classified according to the K value.

3.2.4 Support vector machine

SVM is a supervised learning algorithm. The principle of its training is to project the sample space into a higher-dimensional space, and the feature relationship among the sample data in the higher-dimensional space can be more clearly represented [28]. V is assumed to be the number of samples, then the linear classification surface according to this mapping is shown in Equation 17:

$$H(V) = \omega \cdot V \quad (17)$$

In the Formula 17, ω represents the weight.

The final classification formula is shown in Equation 18:

$$g(V) = \operatorname{sign}\{(\omega \cdot V) + \delta'\} \quad (18)$$

Among them, δ' represents the threshold partial derivative preset by SVM.

3.3 Fuzzy comprehensive evaluation method

The fuzzy comprehensive evaluation method is an evaluation method based on the principle of fuzzy mathematics [29]. When analyzing data, this kind of method can change the qualitative representation of the data into a quantitative representation, so that the evaluation becomes clearer and simpler. In practical data processing, the data to be processed is often complex, covering multiple types, each with different categories and attributes, which makes it difficult to track relevant patterns. It is difficult to define and select the characteristics of this kind of data when it is analyzed by means of a model, and the fuzzy comprehensive evaluation can appropriately reduce the adverse effects of complex data. The localized big data real-time query task processing process studied in this paper is also complex, and it will be affected by multiple factors in the task scheduling, resulting in the scheduling results not meeting expectations. In the task scheduling of big data, the factors affecting the scheduling results cannot be completely determined, and it can also be said to be vague. Therefore, the fuzzy comprehensive evaluation can be used for task classification.

The fuzzy comprehensive evaluation method must first determine the object to be evaluated in the process of operation, and set the evaluation index according to the object to be evaluated and the goal to be achieved. To better judge the final result according to the evaluation index, the evaluation index is grouped according to the level. That is to say, each evaluation index is graded. Since in real data, results or factors have their own focus, the set evaluation indicators also need to be prioritized. In this algorithm, this priority arrangement is mainly carried out by assigning weights to each evaluation index. The formula for determining the weights is shown in Equation 19:

$$S = -\sum_{j=1}^N p_j \cdot \log p_j \quad (19)$$

Among them, p_j represents the output probability of a random variable.

The conditions for the weights of all evaluation indicators to be satisfied are shown in Equation 20:

$$\sum_{j=1}^N \omega_j = 1 (\omega_j > 0) \quad (20)$$

After determining the weights of the evaluation indicators, a fuzzy relationship matrix is first established, which is composed of the fuzzy evaluation results of each single factor variable. Single factor fuzzy evaluation is a quantitative process in which each evaluated object is classified and sorted based on preset indicators to generate a relationship matrix. Next, this relationship matrix is used to comprehensively evaluate each evaluation indicator and obtain preliminary results. Finally, based on the comprehensive evaluation results, the appropriate fuzzy synthesis operator is selected, the fuzzy weights are combined with the fuzzy relationship matrix, and the final result vector is obtained.

Common fuzzy synthesis operators and characteristics are shown in Table 1.

Table 1 shows the common fuzzy synthesis operators. From the content of Table 1, the weighted average operator is more suitable for the use of multiple evaluation indicators in big data tasks.

TABLE 1 Common fuzzy synthesis operators and characteristics.

Type	Reflection of the function of weights	The degree of the comprehensive
Zadeh	Not obvious	Weak
Main factor protrusion type	Obvious	Weak
Weighted average type	Obvious	Strong
Equilibrium mean type	Not obvious	Strong

4 Optimization of localized big data real-time query processing task scheduling algorithm based on artificial intelligence

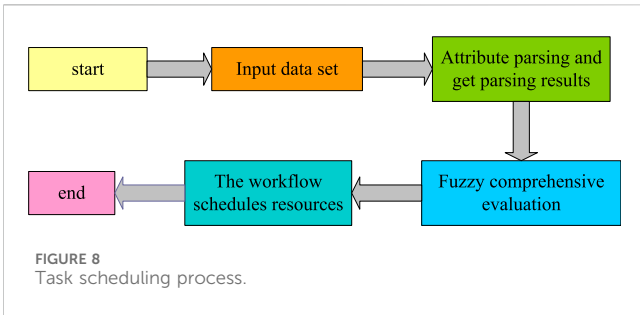
4.1 Optimization design of localized big data real-time query processing task scheduling algorithm based on artificial intelligence

According to the characteristics of the query task, SVM-KNN was used in the system to classify the input instances, and then fuzzy comprehensive evaluation was performed. The classification model was initially completed through preprocessing of training data, establishment and training of SVM and KNN models, and performance evaluation. SVM was used to create decision boundaries, KNN was used for neighboring classification, and the results of both were further optimized through fuzzy comprehensive evaluation. The task scheduling execution process using this model is shown in Figure 8.

Figure 8 shows the task scheduling process after adding SVM-KNN and fuzzy comprehensive evaluation. The optimized task scheduling is mainly to classify the data set more accurately. The process of using SVM, KNN, and fuzzy comprehensive evaluation on the Hadoop platform was to first initialize the system and load the dataset to be processed from the Hadoop platform. Then feature extraction and attribute parsing were performed on the tasks in the input dataset. SVM was used to classify data and determine the category of each task. Simultaneously, KNN was used for further classification or weighted scoring of tasks. Subsequently, based on the results of SVM and KNN, a fuzzy comprehensive evaluation was conducted to achieve a comprehensive assessment of task characteristics. Next, based on the results of fuzzy comprehensive evaluation, resources would be allocated for tasks of different categories. Resource allocation was configured and optimized through Hadoop's resource manager. Tasks to be executed on the Hadoop cluster were arranged based on their priority and resource requirements. Finally, the task execution status were recorded and corresponding reports were generated.

According to the foregoing distributed database, the query task process was optimized, and the optimized query process is shown in Figure 9.

As shown in Figure 9, the optimized query process mainly lied in the parallelized calculation process. In the system, fuzzy comprehensive evaluation was added to classify the information when the query process is executed.



4.2 Optimization experiment of localized big data real-time query processing task scheduling algorithm based on artificial intelligence

The experiment in this paper was based on the Oozie of the Hadoop platform to conduct the task scheduling experiment. The experiment selected 4 data sets and divided them into test set and training set. The dataset included Mashroom, Chess,

Sonar, and Wine datasets. The Mushroom dataset belongs to the UCI machine learning library and contains descriptive features about mushrooms for classification of toxicity. Each sample has 22 attributes, with a total sample size of 8,124, of which 6,513 samples are used for training and 1,611 samples are used for testing. The Chess dataset contains data from thousands of matches, including player ratings and chess piece movement sequences, with approximately 20,000 samples. The Sonar dataset is a dataset used for sonar signal classification tasks, which can be divided into two categories: ore and rock, containing 208 samples. The Wine dataset contains chemical analysis results of three grape varieties, with 178 samples and 13 chemical composition characteristics per sample. All datasets are publicly available. According to the results of the test set, the weight change table of the evaluation index was generated. The data sets used in the experiments are listed in Table 2.

Table 2 is the data set information of the experiment. Training will be carried out according to the data set in Table 2, and task resources will be scheduled according to the results of the comprehensive evaluation of fuzziness.

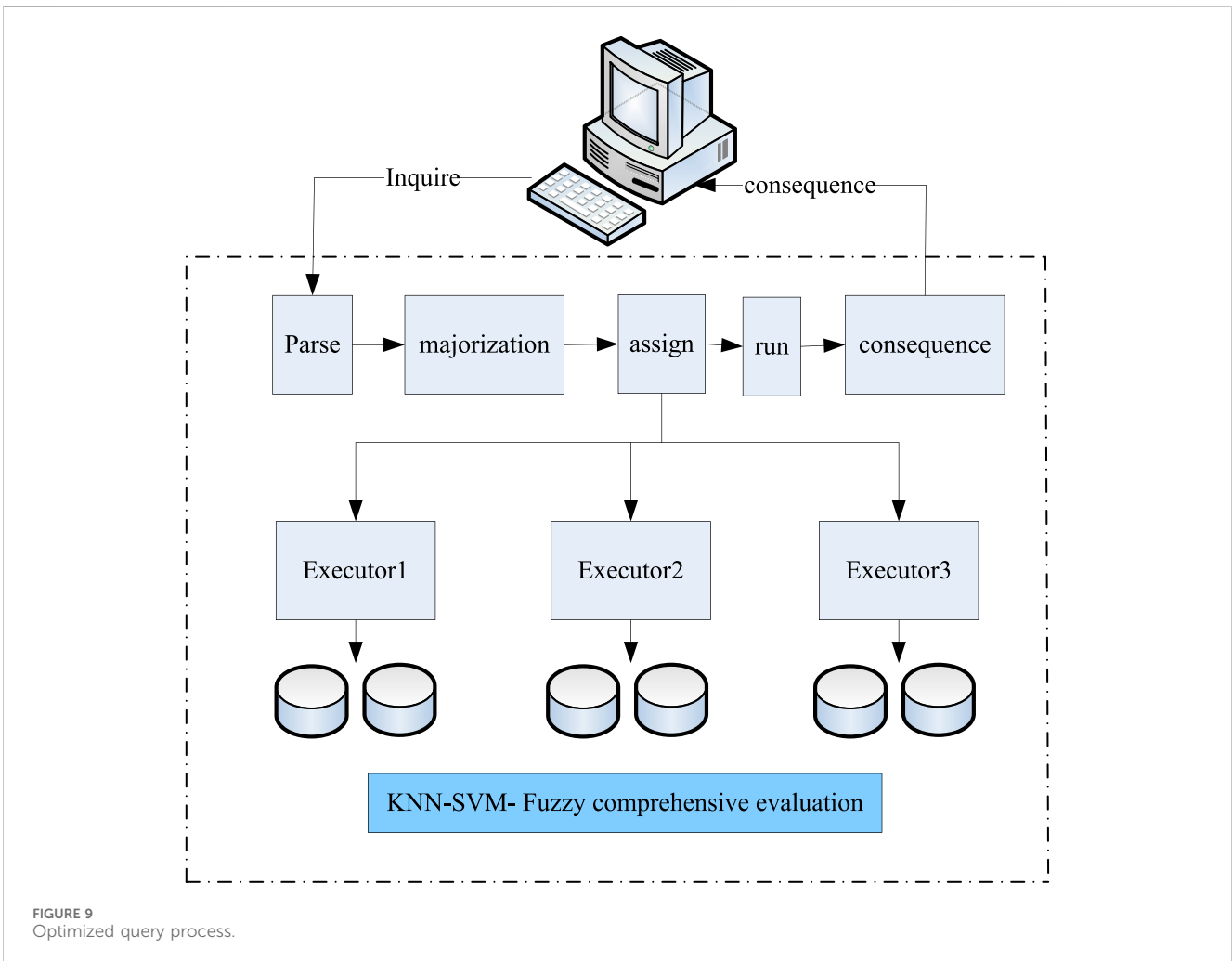


TABLE 2 Experimental dataset.

Dataset	Number of samples	Number of categories	Characteristics of several
Mashroom	8,256	14	26
Chess	3,468	6	42
Sonar	261	2	71
Wine	189	4	11

TABLE 3 Prediction accuracy comparison experimental results.

Dataset	NBM	BP	Optimization model of this paper
Mashroom	80.26	81.98	84.57
Chess	83.81	88.15	89.64
Sonar	80.96	86.25	89.03
Wine	81.12	82.93	85.67

To better evaluate the performance of the optimized model, the time cost and total cost of completing the task were used as evaluation indicators.

First, the optimization algorithm in this paper was compared with NBM and BP neural network. The experimental results are shown in Table 3.

As shown in Table 3, the algorithm used in this paper was better in the classification accuracy of each dataset, with an average accuracy of 87.23%. The second was BP neural network, with an average classification accuracy of 84.83%. The classification effect of NBM was slightly worse than the first two, with an average classification accuracy of 81.53%, which showed that the optimization model constructed in this paper had a good effect on data classification.

By setting different number of tasks, comparing and analyzing the task time cost and total cost before and after optimization, the experimental results obtained are shown in Figure 10.

As shown in Figure 10, when different number of tasks were set, the completion time of the optimized model was shorter. According to the change trend graph, when the number of tasks was smaller, the difference between the two was not obvious. When the number of tasks was larger, the optimized model had more advantages in the consumption of time cost. When the number of tasks was set to 200, the time difference between the two was 2,935 ms. In the total cost of tasks, the trend was similar to the time cost trend. But in the total cost, when the number of tasks was small, the difference between the two was less obvious. The gap between the two did not widen until the number of tasks reached 200. But in general, the optimized model had more advantages in cost consumption. The results showed that the optimization method in this paper had certain feasibility in improving the cost consumption of task scheduling.

The comparative experiment on the processing query of the distributed database is mainly to judge the performance according to the change of the processing query statement time. The experimental results obtained are shown in Table 4.

As shown in Table 4, the optimized scheduling scheme significantly reduced the query time of the same statement. But overall, the reduction in time cost was not particularly large. On the one hand, the query time of some statements themselves was relatively short. The original query time of statement 3 was 1.96 s, which was not long. The optimized query time was 0.83 s, a reduction of 1.13 s, so the increase in time cost would not be obvious. But in fact, in a short-term statement query, the time cost itself was difficult to reduce. On the other hand, it may be related to the kernel of the system. The kernel of the system affected the speed of the system processing tasks, and the utilization of the kernel of the system was related to the scheduling of the kernel. However, the scheduling behavior itself consumes time, so when the core utilization was not obvious, the performance improvement was not obvious.

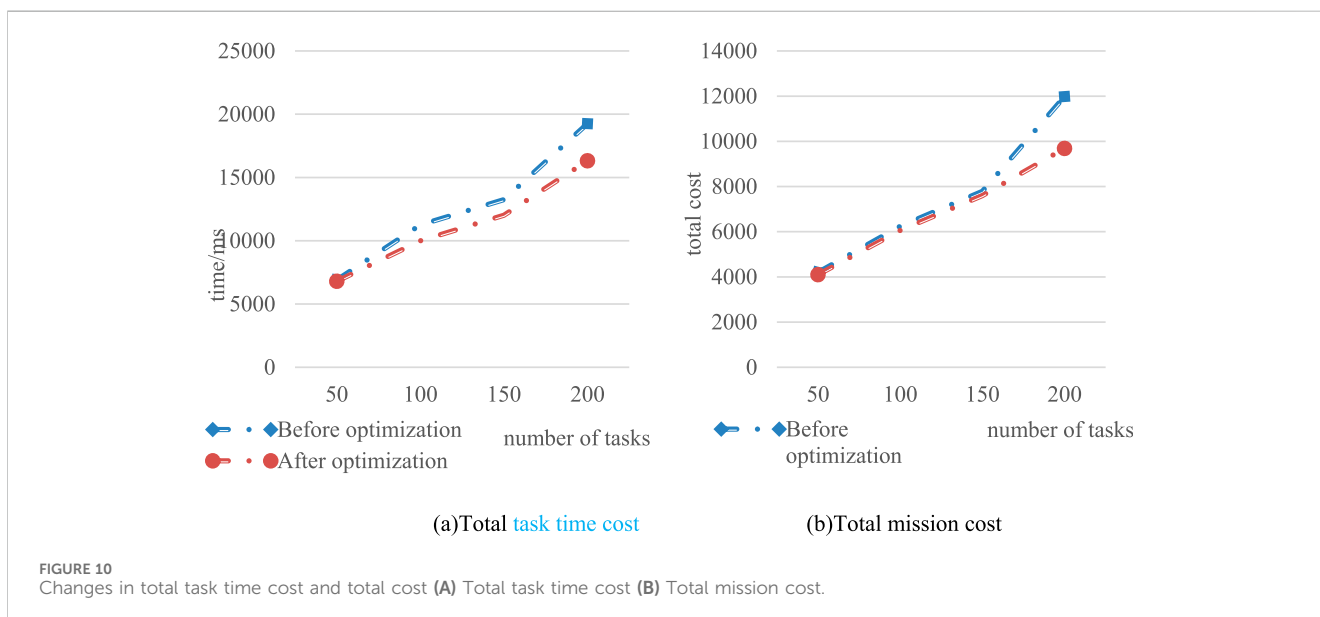


TABLE 4 Comparison of query time before and after optimization (Unit: s).

Statements	Before optimization	After optimization
1	36.98	35.17
2	15.69	12.26
3	1.96	0.83
4	86.45	83.97
5	9.33	7.21

5 Discussion

In response to the rational division and allocation of information and resources in big data, AI technology was used to optimize task scheduling algorithms. An optimization model for task scheduling algorithms was designed using SVM, KNN, and fuzzy comprehensive evaluation. The results showed that the task scheduling algorithm used in the study achieved an average classification accuracy of 87.23% on the Mashroom, Chess, Sonar, and Wine datasets. Mangalampalli S. et al. used a deep Q-learning network model, which achieved a classification accuracy of only 84.68% on the Mashaom dataset [30]. Compared with the classification algorithm used in the study, this method lacked efficient processing and comprehensive evaluation capabilities for complex data features, resulting in lower classification accuracy. In addition, the scheduling scheme used in the study had a minimum query time of only 0.83 s and had high response performance. Ghafari R. et al. adopted a task scheduling method based on metaheuristic, which took the shortest time to process query statements at 1.35 s, slightly inferior to the task resource scheduling scheme proposed in the study [31]. The task scheduling algorithm proposed by the research not only improved the accuracy of data classification, but also had significant advantages in response performance.

6 Conclusion

Through the theoretical understanding of localized big data and real-time query tasks, this paper revealed that one of the factors affecting the completion time of query tasks was the allocation and scheduling of resources, and the task scheduling method under big data was briefly introduced. In this content, a query process of a distributed database was proposed, and an optimization and improvement scheme based on SVM, KNN and fuzzy comprehensive evaluation was proposed through the characteristics of the running process of task scheduling. The classification accuracy of the classification model in this paper was first tested by setting the relevant experimental data set. Then through the comparison experiment with NBM and BP neural network, it is concluded that the optimization model proposed in this paper was more accurate in classification, thus confirming the effect of the proposed model in classification optimization. At the same time, it is proved that the proposed

model could be used for the optimization of task resource allocation. An experiment for task scheduling was set up based on the results of this experiment. By giving different number of tasks, the pre-optimized and optimized models were processed with related tasks. By comparing the time spent on tasks in the two groups and the trend of total cost, it is concluded that the proposed model significantly reduced the time spent in processing different tasks, and the total cost was also lower than the model before optimization. This verified that the proposed optimization model had obvious optimization effect in the task scheduling algorithm. Finally, through the statement query time experiment before and after the task scheduling optimization in the distributed database, it is concluded that the optimized task scheduling algorithm had a significant reduction in the actual statement task processing time. The results illustrates that the optimization of the task scheduling algorithm for real-time query processing in this paper is successful.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

MS: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Validation, Writing—original draft, Writing—review and editing. LS: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Supervision, Writing—original draft, Writing—review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

Authors MS and LS were employed by CEICloud Data Storage Technology (Beijing) Co., Ltd.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Fu X, Sun Y, Wang H, Li H. Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm. *Cluster Comput* (2023) 26(5):2479–88. doi:10.1007/s10586-020-03221-z
- Wang C, Yu X, Xu L, Wang W. Energy-efficient task scheduling based on traffic mapping in heterogeneous mobile-edge computing: a green IoT perspective. *IEEE Trans Green Commun Networking* (2022) 7(2):972–82. doi:10.1109/tgcn.2022.3186314
- Zhang Y, Zhai B, Wang G, Lin J. Pedestrian detection method based on two-stage fusion of visible light image and thermal infrared image. *Electronics* (2023) 12(14):3171. doi:10.3390/electronics12143171
- Chen J, Han P, Zhang Y, You T, Zheng P. Scheduling energy consumption-constrained workflows in heterogeneous multi-processor embedded systems. *J Syst Architecture* (2023) 142:102938. doi:10.1016/j.sysarc.2023.102938
- Gruzlikov AM, Kolesov NV, Skorodumov YM, Tolmacheva MV. Task scheduling in distributed real-time systems. *J Computer and Syst Sci Int* (2017) 56(2):236–44. doi:10.1134/s1064230717020101
- Kaur A, Sharma S. An analysis of task scheduling in cloud computing using evolutionary and swarm-based algorithms. *Int J Computer Appl* (2018) 89(2):11–8.
- Lu P, Zhu Z. Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: a comparison study. *J Lightwave Technology* (2017) 35(24):5335–46. doi:10.1109/jlt.2017.2777605
- Sarathambekai S, Umamaheswari K. Task scheduling in distributed systems using heap intelligent discrete particle swarm optimization. *Comput Intelligence* (2017) 33(4):737–70. doi:10.1111/coin.12113
- Paschen J, Paschen U, Pala E, Kietzmann J. Artificial intelligence (AI) and value co-creation in B2B sales: Activities, actors and resources. *Australasian Marketing Journal* (2021) 29(3):243–251. doi:10.1016/j.ausmj.2020.06.004
- Li R, Zhao Z, Zhou X, Ding G, Chen Y, Wang Z, et al. Intelligent 5G: when cellular networks meet artificial intelligence. *IEEE Wireless Commun* (2017) 24(5):175–83. doi:10.1109/MWC.2017.1600304WC
- Labovitz DL, Shafner L, Gil MR, Virmani D, Hanina A. Using artificial intelligence to reduce the risk of nonadherence in patients on anticoagulation therapy. *Stroke* (2017) 48(5):1416–9. doi:10.1161/STROKEAHA.116.016281
- Youssef A, El-Telbany M, Zekry A. The role of artificial intelligence in photovoltaic systems design and control: a review. *Renew and Sustainable Energy Rev* (2017) 78:72–9. doi:10.1016/j.rser.2017.04.046
- Syam N, Sharma A. Waiting for a sales renaissance in the fourth industrial revolution: machine learning and artificial intelligence in sales research and practice. *Ind Marketing Management* (2018) 69:135–46. doi:10.1016/j.indmarm.2017.12.019
- Ehteram M, Allawi MF, Karami H, Mousavi SF, Emami M, El-Shafie A, et al. Optimization of chain-reservoirs' operation with a new approach in artificial intelligence. *Water Resour Management* (2017) 31(7):2085–104. doi:10.1007/s11269-017-1625-6
- Bailey R, Parsheera S. Data localization in India: paradigms and processes. *CSI Trans ICT* (2021) 9(3):137–50. doi:10.1007/s40012-021-00337-4
- Huang W, Shi Z, Xiao Z, Chen C, Li K. A large-scale task scheduling algorithm based on clustering and duplication. *J Smart Environments Green Comput* (2021) 1(4):202–17. doi:10.20517/jsegc.2021.13
- Aziza H, Krichen S. Bi-objective decision support system for task-scheduling based on genetic algorithm in cloud computing. *Computing* (2018) 100(2):65–91. doi:10.1007/s00607-017-0566-5
- Krishnadoss P, Jacob P. OCSA: task scheduling algorithm in cloud computing environment. *Int J Intell Eng Syst* (2018) 11(3):271–9. doi:10.22266/ijies2018.0630.29
- Lin W, Wang W, Wu W, Pang X, Liu B, Zhang Y. A heuristic task scheduling algorithm based on server power efficiency model in cloud environments. *Sustainable Comput Inform Syst* (2017) 20:56–65. doi:10.1016/j.suscom.2017.10.007
- Sevinç E, Coşar A. An evolutionary genetic algorithm for optimization of distributed database queries. *Computer J* (2018) 54(5):717–25. doi:10.1093/comjnl/bxp130
- Jing C, Liu W, Gao J, Pei O. Research and implementation of HTAP for distributed database. *Xibei Gongye Daxue Xuebao/Journal Northwest Polytechnical Univ* (2021) 39(2):430–8. doi:10.1051/jnwpu/20213920430
- Yulianto AA. Extract transform load (ETL) process in distributed database academic data warehouse. *APTİKOM J Computer Sci Inf Tech* (2020) 4(2):61–8. doi:10.11591/aptikom.j.csit.36
- Torshiz MN, Esfaji AS, Amintoosi H. Enhanced schemes for data fragmentation, allocation, and replication in distributed database systems. *Computer Syst Sci Eng* (2020) 35(2):99–112. doi:10.32604/csse.2020.35.099
- Makridakis S. The forthcoming artificial intelligence (AI) revolution: its impact on society and firms. *Futures* (2017) 90:46–60. doi:10.1016/j.futures.2017.03.006
- Li X, Wang J, Yang C. Risk prediction in financial management of listed companies based on optimized BP neural network under digital economy. *Neural Comput Appl* (2023) 35(3):2045–58. doi:10.1007/s00521-022-07377-0
- Donnellan E, Aslan S, Fastrich GM, Murayama K. How are curiosity and interest different? Naive Bayes classification of people's beliefs. *Educ Psychol Rev* (2022) 34(1):73–105. doi:10.1007/s10648-021-09622-9
- Zhang S, Li J, Li Y. Reachable distance function for KNN classification. *IEEE Trans Knowledge Data Eng* (2022) 35(7):7382–96. doi:10.1109/TKDE.2022.3185149
- Kurani A, Doshi P, Vakharia A, Shah M. A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Ann Data Sci* (2023) 10(1):183–208. doi:10.1007/s40745-021-00344-x
- Zhu L. Research and application of AHP-fuzzy comprehensive evaluation model. *Evol Intelligence* (2022) 15(4):2403–9. doi:10.1007/s12065-020-00415-7
- Mangalampalli S, Karri GR, Kumar M, Khalaf OI, Romero CAT, Sahib GA. DRLBTA: deep reinforcement learning based task-scheduling algorithm in cloud computing. *Multimedia Tools Appl* (2024) 83(3):8359–87. doi:10.1007/s11042-023-16008-2
- Ghafari R, Kabutarkhani FH, Mansouri N. Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. *Cluster Comput* (2022) 25(2):1035–93. doi:10.1007/s10586-021-03512-z