

Research Article

Construction of an Intelligent APP for Dance Training Mobile Information Management Platform Based on Edge Computing

Yan Gao^{1,2} and Dazhi Xu^{3,4} 

¹College of Art, Hunan University of Arts and Science, Changde 415000, Hunan, China

²School of Management, Shinawatra University, Thani, Bangkok 12160, Thailand

³College of Economics and Management, Hunan University of Arts and Science, Changde 415000, Hunan, China

⁴Public Administration, Central South University, Changsha 410083, Hunan, China

Correspondence should be addressed to Dazhi Xu; xudazhi@huas.edu.cn

Received 11 May 2021; Revised 4 June 2021; Accepted 19 June 2021; Published 28 June 2021

Academic Editor: Sang-Bing Tsai

Copyright © 2021 Yan Gao and Dazhi Xu. 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 recent years, with the rapid development of modern technology and the continuous promotion of information technology, information technology has been widely used in modern performing arts. Information management has become the most practical and effective method and means in performing arts training management, but as the amount of various data grows exponentially, the requirements for computing processing power and speed for massive amounts of data and information are also increasing day by day. This article aims to study the use of edge computing to solve the problems of high latency and high cost when traditional cloud computing centers provide services. In response to these problems, this paper proposes a data acquisition and processing system architecture based on edge computing, which uses edge computing to mine the computing power of edge terminals in the network, performs partial or all calculations at the edge terminals, processes private data, and reduces cloud computing. The center's computing, transmission bandwidth load, and energy consumption, combined with cloud computing, provide data acquisition, processing, and analysis solutions with low latency and high processing capabilities. This article details how to optimize edge server development to minimize access latency and consider network reliability when requesting access to edge servers. This paper uses the proposed edge server deployment algorithm and system load optimization, which can effectively reduce the network delay and system load of the edge server, and the experimental results show that the system performance is improved by 23.5% after effective optimization.

1. Introduction

With the continuous advancement of science and technology, as well as the development of the art of dance and the education of dance itself, the scientific training of dance has gradually gained the attention of all dance educators. In modern performing arts, information technology has widely penetrated into training management, daily management, health management, and other aspects, making dance training more efficient and effective. In recent years, various sports smart APPs have continued to rise. Users only need to open the APP to perform various sports such as running, fitness, dancing, and learning martial arts through the APP.

In addition, during the exercise, the sports APP can accurately record the user's exercise time and calories burned and then perform effective evaluation, give targeted exercise suggestions, and specifically provide users with various sports service needs. Design exercises such as fitness, running, and professional training are according to the needs of users. Therefore, designing a scientific and effective dance training information management system combined with a smart APP can make the management of dance training more informationized, systematic, and efficient, thereby changing the status quo of dance training and playing a major role in further improving the level of modern performing arts. The data information generated by the mobile

information management system for dance training is sent to each cloud platform through the network, and the analysis results are obtained through the central function. However, due to the widespread use of distributed intelligent devices, a large amount of data transmission blocks the transmission channel. There are many difficulties in using cloud computing technology to solve this problem, and the latest computing technology provides an effective method to solve this problem. Information technology and scientific information management software can be used to enable functions such as submitting online training plans and remotely monitoring training effects. It emphasizes the role of smart APP in dance training and achieves the best training management effect with minimal investment.

Services based on the Internet of Things benefit from cloud services. Yang L. uses edge computing to study computing offloading, delivering computing to a new paradigm at the edge of popular networks near mobile users. In order to stimulate cloud service operators and local edge server owners to participate in computing offloading, he formulates the interaction between cloud service operators and edge server owners as a Stackelberg game to maximize cloud service operations by obtaining the best payment and fees. The utility of the business and edge server owners: however, his research cannot satisfy more complex applications, and further research is needed [1]. Mobile edge computing (MEC) can be freed from computationally intensive workloads by offloading mobile devices to nearby MEC servers. In order for offloading to be effective, radio and computing resources need to be dynamically managed to cope with time-varying computing requirements and wireless fading channels. Mao has developed an online joint radio and computing resource management algorithm for multiuser MEC systems. Its goal is to minimize the long-term average weighting and power consumption of mobile devices and MEC servers under the effect of task buffers. Stability constraints: specifically, in each time slot, the optimal CPU cycle frequency of the mobile device is obtained in a closed form, and the optimal transmit power and bandwidth allocation for calculating offloading are determined by the Gauss–Seidel method. For the MEC server, the optimal frequency of the CPU core and the optimal MEC server scheduling decision are derived in a closed form. But it did not consider that this algorithm is too complicated, which may cause a higher load on the equipment during operation [2]. The advent of the Internet of Things and 5G applications requires the integration of centralized cloud computing and emerging mobile edge computing (MEC) with existing network infrastructure to enhance storage, processing, and caching functions not only in a centralized manner but also in a distributed manner. Achieve the following goals: support both delayed and mission-critical applications at the same time. Rimal studied the performance improvement of centralized cloud and MEC-enabled integrated optical fiber wireless (FiWi) access network. A novel unified resource management solution is proposed, which combines centralized cloud computing and MEC computing offloading activities into the underlying FiWi dynamic bandwidth allocation process. By using time

division multiple access, both MEC and cloud traffic are arranged outside the transmission time slot of FiWi traffic. An analysis framework was developed to model the packet delay, response time efficiency, gain-offload cost ratio, and communication-to-computation ratio of cloud and broadband access traffic. But his research is not supported by actual data, and a lot of experiments are needed to verify the feasibility of the theory [3].

The innovation of this paper is as follows: (1) reducing the data transmission delay of the edge network through the designed algorithm, using the abnormal data recognition algorithm to identify abnormal data, improving the computing efficiency of edge computing, and reducing the cost of edge server deployment; (2) adopting an open platform integrating network, computing, storage, and application core capabilities, services provided at close distances, and their applications starting from the edge, making network services more responsive and satisfying real-time processing, smart applications, and basic industry needs such as security and privacy.

2. Edge Server Deployment Algorithm Design Experiment

2.1. Edge Computing. Edge computing refers to an open platform that uses the core capabilities of network, computing, storage, and applications on the side close to the source of things or data. Services are provided at close range, and its applications start from the edge, making network services more responsive, and meet the basic industry needs of real-time processing, smart applications, security, and privacy [4]. Mobile edge computing technology provides mobile users with the wireless network computing and cloud functions, localizes application services that require a lot of computing resources, and deploys them to nearby mobile users, thereby reducing wireless network delays and high bandwidth transmission capabilities. In addition, reducing backhaul bandwidth requirements can greatly reduce operating costs. Edge computing is located between physical entities and industrial connections, or on top of physical entities [5, 6]. In addition, cloud computing can continue to access edge data on edge computers [7, 8]. Since the introduction of cloud computing, although it has brought convenience to people's lives and work, the development of the Internet of Things and the increase in data volume have revealed the drawbacks of cloud computing. People began to explore new data processing methods to solve the disadvantages of cloud computing. Mobile Cloud Computing (MCC) was proposed [9, 10]. The definition of cloud computing is that a mobile device delivers computing unit storage, operating platforms, other resources, or information service resources, or information service delivery communication networks from the cloud center over the mobile device on demand. This is the combined product of the mobile communication network [11]. Mobile devices connect to the cloud hub over a wireless network and send their own tasks to the cloud hub. There is no need to perform data management calculations or other processing [12], just provide data input and output in the cloud center. This

feature allows users to download and install applications, store mobile storage memory, view data on a variety of devices and platforms, improve application compatibility, and cover mobile devices [13, 14]. Edge computing refers to the “sinking” of cloud computing functions at the network edge, which is a new type of computer model used with network edge devices. The “end” of edge computing is the data source generated by the end of the network terminal [15, 16]. For the resources between the data paths in the cloud data center, the basic idea is to perform computer work on the computer resources near the data source. Edge computing and cloud computing are complementary to each other [17, 18]. The portable computing chip is a computing chip. The traditional wireless access network is located between the wireless access point and the wired network, which can provide end users with higher bandwidth and waiting for data service, which can save time and reduce bandwidth, data service requirements for network search [19]. Although mobile edge computing technology is very convenient, it faces many challenges when applied to real network scenarios. First, a large number of network access points (APs) need to be developed to allow almost all smart electronic devices to access the Internet [20]. However, compared to these APs, the number of edge servers is very limited considering the deployment cost. How to optimize these low-tech servers to develop a modern network with a large number of APs is a major challenge. Mobile edge computing technology provides mobile users with a wealth of application services by migrating service resources from remote cloud computing centers to edge networks. This can significantly eliminate the network during the data transmission process provided by the Remote Cloud Computing Center. Time delay: the innovation of this paper is to reduce the data transmission delay of the edge network through the designed algorithm, use the abnormal data recognition algorithm to identify abnormal data, improve the computing efficiency of edge computing, and reduce the cost of edge server deployment.

2.2. Calculation of Transmission Delay of Edge Network.

In order to ensure that a large number of mobile devices can access the edge network to obtain services at any time during the movement, a large number of APs need to be deployed. The transmission delay of requesting data from these APs to the edge server will greatly affect the edge server. Quality of service: if the data load needs to be transmitted on the switch through which the data transmission path passes is too large, it will cause a very serious delay in the edge network [21, 22]. For different types of application services, the request data for resource requirements are also diversified.

The variable $R_{t_i}^n$ represents the average transmission delay of the request data of the application service n on the link t_i . The variable $E_{l,k}^{t_i}$ indicates whether the link t_i is on the path when these request data are transmitted. The overall network delay generated when the edge server k provides application services n to the mobile devices within the coverage of the edge server l is

$$R_{t_i}^n = \omega_{l,k}^n \cdot S_l^n \cdot \sum_{t_i \in T} R_{t_i}^n E_{l,k}^{t_i}, \quad (1)$$

where S_l^n represents the number of requests for application n within the service range of edge server l , namely,

$$S_l^n = m_l \cdot \alpha_l^n \cdot \lambda, \quad (2)$$

where α_l^n represents the request rate of mobile devices for service application n within the coverage of edge server l .

When the request of application n reaches the edge server that provides the service, the edge server will process the resources it needs. Assuming that each mobile device generates an average rate of requests for application service n , the variable C_l^n is used to represent the average rate of application n reaching edge server l ; then

$$C_l^n = \sum_{k \in L} J_l^n \cdot \omega_{k,l}^n \cdot m_k \cdot \lambda_m \cdot \alpha_l^n. \quad (3)$$

The edge server 1 is used to provide service time for the application, and its average service time is $1/\nu_l$, where ν_l represents the average service difference of the edge server 1. Therefore, it can be deduced that the processing delay of all service requests in the edge server 1 is

$$S_l^t = \nu_l - \sum_{n \in N} C_l^n. \quad (4)$$

In order to ensure that the service rate is greater than the requested arrival rate, it is necessary to meet

$$\nu_l - \sum_{n \in N} C_l^n > 0. \quad (5)$$

In order to limit the sum of the number of mobile devices covered by each edge server to be equal to M and to ensure that no requests sent by mobile devices are repeatedly counted, then

$$\sum_{k \in L} m_k = m, \ni m_l \in G. \quad (6)$$

2.3. Abnormal Data Recognition Algorithm. When collecting data, the monitored object sometimes performs some more intense behaviors [23, 24]. The effect of these behaviors and actions on the data is the rapid change of the acceleration or angular velocity. Therefore, this type of data is often affected. Think of it as abnormal data [25, 26]. For the identification and processing of abnormal data, taking into account the timeliness of environmental data and the limitation of processor performance, when running filling data on edge devices, it is simple and accurate to use the latest hot card filling and then perform according to the known node data. According to the filling result, insert the function $s(t)$, and then use the interpolation function to quickly obtain the initial running data of the corresponding position. The mathematical expression is to take the number of interpolation nodes on the original function $m(n)$ as $x+1$, which are $(m_0, n_0), (m_1, n_1), (m_2, n_2), \dots, (m_x, n_x)$ and substitute the following formula:

$$y = a_0 + a_1 n + a_2 n^2 + a_3 n^3 + \dots + a_{x-1} n^{x-1}. \quad (7)$$

The following is also available:

$$y = s(t) = \sum_{i=0}^x y_i \prod_{j=0, i \neq j}^x \frac{n - n_j}{n_i - n_j}. \quad (8)$$

After calculating the nearest points from the missing data through the above formula and substituting them into the above expression, the corresponding polynomial can be obtained. The point N corresponding to the specified value of the function is then replaced by the interpolating polynomial to get an approximation of the missing value.

2.4. Edge Network Task Load Calculation. On the MEC server side, the computing resources of the server are also limited [27]; that is, the MEC server can only meet the computing requirements of certain data at the same time. In this article, we assume that the MEC server can only perform one computing task at a time. After the MEC server completes the calculation task, it returns the calculation result to the mobile user.

Assuming that the SV unit can only upload the data of one calculation task to the MEC server at the same time and the transmission power of the SV unit is expressed as G_{sv} , then the transmission rate can be expressed as

$$K(G_{sv}) = \mu \log_2 \left(\frac{t_0 (k_0/k)^x G_{sv}}{X_0 \mu} \right). \quad (9)$$

Among them, t_0 is the energy loss constant, x is the energy loss index, k is the distance between the mobile device and the MEC server, and μ represents the bandwidth of the system.

According to the energy consumption and time required for the execution of computing tasks on the local CPU and MEC server, these data can be used by combining the two factors of energy consumption and delay into one factor, and the load can meet the individual needs of different users and flexibly adjust related influencing factors [28].

Calculate the time s required for all computers to execute:

$$s = \max\{s^{m_1}, s^{m_2}\}, m_1 + m_2 = m. \quad (10)$$

The energy t required to perform all tasks is

$$t = t^{m_1} + t^{m_2}, m_1 + m_2 = m. \quad (11)$$

Suppose the overall load of the system is expressed as X :

$$X = \lambda^s s + \lambda^t t. \quad (12)$$

Among them, the coefficients λ^s and λ^t , respectively, represent the weight of the delay and energy consumption in the calculation task. The two coefficients satisfy the following relationship:

$$\begin{cases} \lambda^s + \lambda^t = 1, \\ \lambda^s \geq 0, \\ \lambda^t \geq 0. \end{cases} \quad (13)$$

When λ^s is larger, it means that the delay of the calculation task is larger, and when λ^t is larger, it means that the

user's energy consumption is too high at this time. In order to reduce the energy consumption of the calculation task, a weight coefficient can be selected.

In this way, our goal is to optimize load X :

$$G_1: \min X. \quad (14)$$

2.5. Service Configuration Cost Calculation. The cost of application service configuration refers to the resource requirements for configuring VRC on the edge server to control the state synchronization between VRCs of the same application service [29, 30].

This cost is defined as a quadratic function $\phi \cdot x \cdot |n|$ containing the total number of VRC services and the number of all application services, where ϕ is an indicator coefficient between 0 and 1. Define a linear function $\varphi \cdot S_{cl}(n, x)$ that divides the average request response delay cost from the average response time obtained; then, the total cost of each application service standard supported by n VRCs can be expressed as follows.

3. Design of the Dance Training Mobile Information Management System Based on Edge Computing

3.1. Architecture Design Based on the Edge Computing System. According to the characteristics of edge computing, it is necessary to use edge devices and cloud computing centers to coordinate the realization of the overall system functions. Traditional cloud computing requires data to reach remote cloud servers through wireless access and a central network. After processing the data, the cloud server returns the result to the user through a long-distance backhaul connection. It can be seen that this method of operation is very time-consuming. As one of the basic infrastructures of today's 5G network technology, the edge computing network is mainly developed through distributed portable service nodes to provide these nodes with computing and storage resources for terminals to process user requests and make full use of them to make the network have low latency and high scalability.

Figure 1 shows the basic schematic diagram of a mobile edge computing system. On the MEC platform, the edge server is located on the side of the wireless access network, which greatly shortens the distance from the user equipment. Due to the shortened transmission distance, the MEC task no longer requires long-distance backhaul connections and core networks, which can generally reduce delay. On the other hand, the computing functions of edge servers are much more powerful than mobile devices, which significantly reduces task processing time. The design idea of the portable computing system can not only meet the needs of portable terminal equipment to expand the computer capacity but also make up for the long transmission delay of the cloud computing platform. Mobile edge computing service nodes are the core of the entire network, and developers develop many computing and storage resources on these nodes. Real-time services can be provided when a

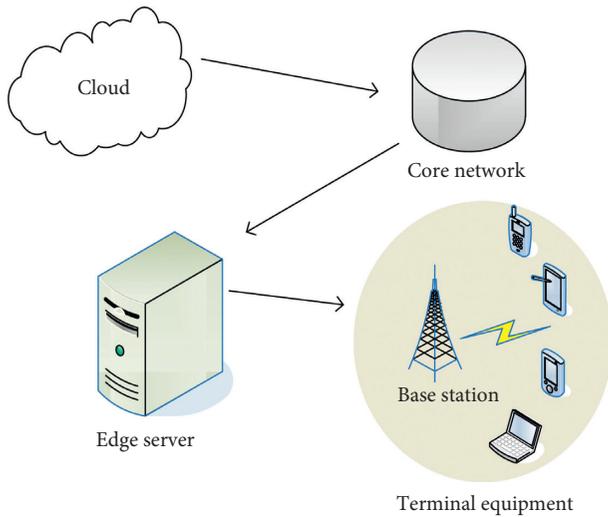


FIGURE 1: Basic framework diagram of mobile edge computing system.

request is received, thereby saving basic computing network resources.

3.2. Design of the Dance Training Mobile Information Management System

3.2.1. System Structure Design. The dance training management information system is a comprehensive management system for multilevel users. It has high requirements for the relevance and real-time performance of each link. Therefore, it is necessary to use a computer network system as a platform for construction. The main body of sports training management is information. This information includes all the information generated by the interaction or interrelation of each link in sports training management; they are the bond of each link. The transmission of information in various links is the key to information management, and the role of the management information system is to make the transmission of information more timely and accurate. Therefore, using the management information system as a tool to conduct sports training management is the development direction of modern sports management. Through demand analysis, the dance training management information system is divided into four modules: user management module, training plan management module, training effect management module, and body information management module as shown in Figure 2.

3.3. MySQL-Based Data Storage Module. Data management is the process of using computer hardware and software technology to effectively collect, store, process, and apply data. In the process of system operation, a large amount of data needs to be processed, and the database can access and process data reasonably and efficiently. Among them, MySQL is popular among small- and medium-sized websites due to its small size, fast speed, and low cost. This module will also choose it as a data storage tool. When storing data,

because the system uses multiple edge devices to transmit data to the cloud computing center at the same time, the data transmission speed will be very fast in the actual operation process. When requesting, the phenomenon of task jam will inevitably occur. This article calls the database connection pool under the MySQL software system, stores the requests for data storage tasks in the connection pool in a certain order, and calls them sequentially, thereby alleviating the phenomenon of task blocking.

After the database connection is established through Figure 3, when the database server receives the database execution message uploaded by the WEB, the message is analyzed. Since the database execution message sent by WEB is in string format and the data type is text, it is stored in the byte array after being received through the data interface. First, the byte array is converted into a string in UTF-8 encoding format through the conversion function. Get the index value through the keyword SQL and the quotation mark delimiter, thereby obtaining the SQL execution statement, instantiate the execution statement through the SqlCommand class, call the built-in ExecuteNonQuery function to execute the operation statement, and complete the database operation.

As shown in Figure 4, the basic idea of the connection group is to store the database connection as an object in memory during system initialization. When the database receives a connection request, it will not create a new connection but will delete established connection inactive objects from the connection group. After use, reconnect the connection to log in to the next access request.

4. Edge Server Deployment Algorithm

4.1. Transmission Delay Analysis of Edge Network. Under this system architecture, for the transmission delay of end-to-end data transmission, the submission time is generally used to verify the time interval from one device to another for submitting data. As shown in the mobile edge network architecture based on SDN technology, this paper uses a Python program to develop a simulation environment to simulate the impact of different configurations of VRC supporting multiple application services in the edge network on the service request response delay of each application. This article analyzes the network transmission delay by comparing the four algorithms: EOESPA, RNOESPA, KMCA, and RESPA. In order to further study how the limited edge server access delay affects the maximum reliability of the SDN-based edge network, the edge network service deployment plan is represented by four algorithms, as shown in Table 1.

As shown in Figure 5, it can be observed that, for each average service request data size setting, the average access delay achieved by the deployment scheme proposed by RNOESPA is very close to the best performance that EOESPA can achieve and is even worse than the classic KMCA. Moreover, regardless of the amount of data transmitted, the average delivery time is below 2 ms, and with the increase of data, the delivery time will not change

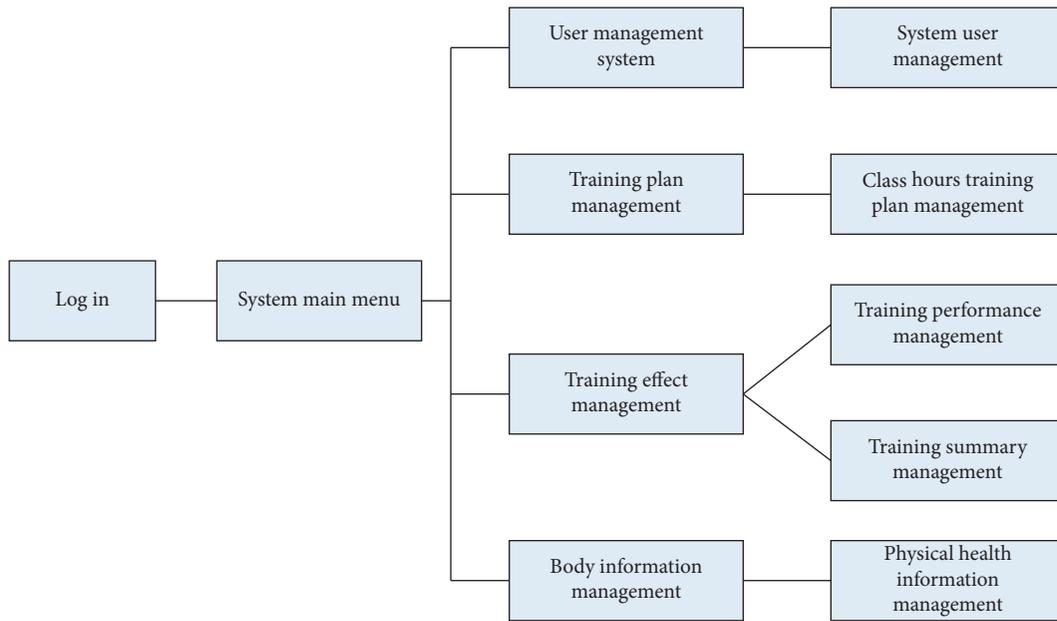


FIGURE 2: Framework diagram of dance training mobile information management system.

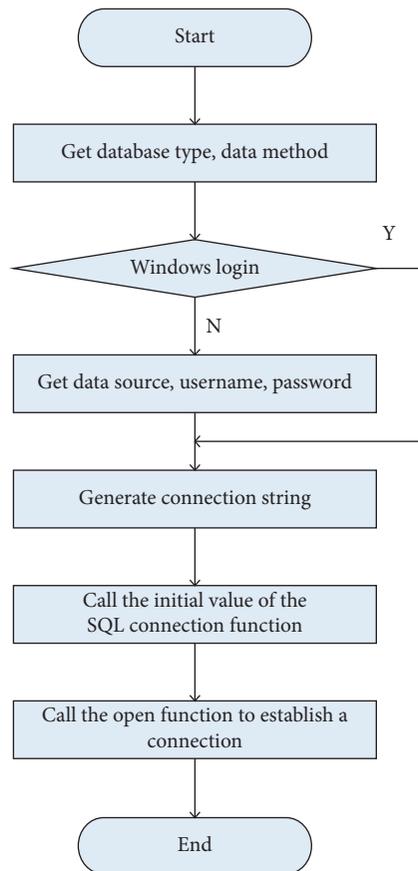


FIGURE 3: Database connection flow chart.

much, which can better ensure the real-time nature of data transmission.

As shown in Figure 6, as the number of mobile devices in the edge network increases, the edge server deployment

solution proposed by RNOESPA can also achieve better performance than KMCA. In addition, from the experimental results shown in Figures 5 and 6, it can be seen that the edge server access delay caused by the deployment

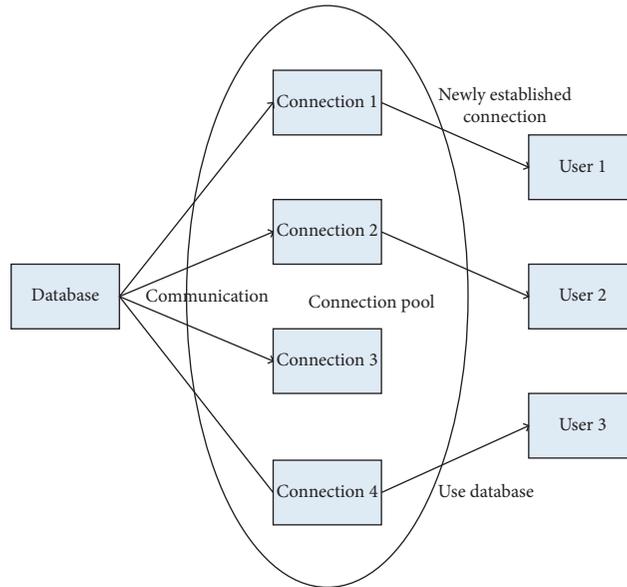


FIGURE 4: Schematic diagram of MySQL connection pool.

TABLE 1: Edge network server deployment plan.

Algorithm	Computation complexity
EOESPA	$O(K^2 \cdot V \cdot C^K)$
RNOESPA	$O(V ^2 + E \cdot V)$
KMCA	$O(K^3 \cdot V ^2 \cdot t)$
RESPA	$O(1)$

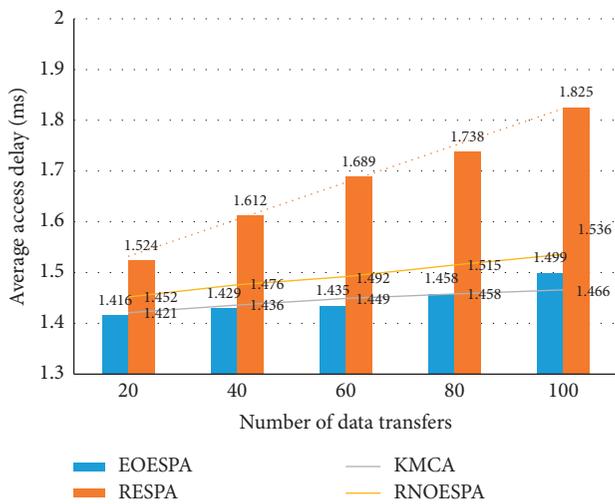


FIGURE 5: The impact of the average request data volume on the minimum access delay of edge servers.

schemes proposed by all four algorithms increases with the size of the service request data and the growth of the number of mobile devices. At the same time, the experimental results also show that the increase in the number of mobile devices has less impact on the access latency of the edge server than the increase in the size of the service request data.

In order to avoid data transmission disorder when the system architecture is sending data, the stability of PAG data

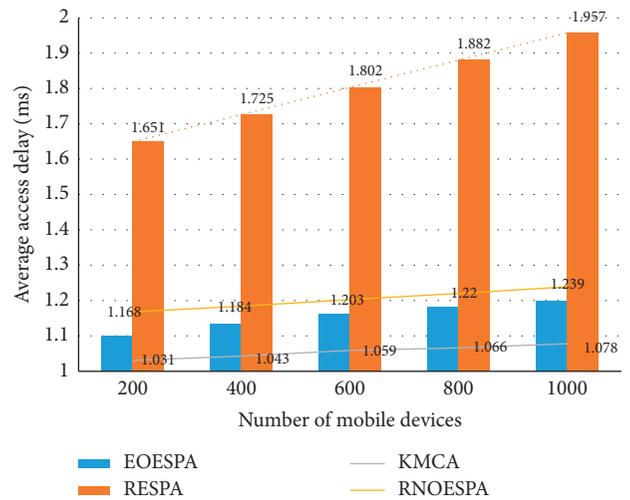


FIGURE 6: The impact of the number of mobile devices on the minimum latency of edge servers.

upload and the stability of PAG received data are analyzed. As shown in Tables 2 and 3, the delay number is that the response is not received immediately but received after a few responses. It can be seen from the results that with the increase of data sent per second, there is no packet loss in data upload, and the stability of data upload is better. At the same time, although there is a delay in receiving data when data is received, there is no packet loss, which can be determined. The stability of data acquisition is good.

4.2. Identification and Analysis of Abnormal Data. Data transmission due to various reasons, abnormal data will be generated during the transmission process, the abnormal data is identified and analyzed, the data is crawled by Python, and the algorithm is used to identify the abnormal data.

TABLE 2: Data upload result.

Type of data	Number of packets sent per second	Delay number	Packet loss rate
Data upload	10	0	0
	50	0	0
	100	0	0
	500	0	0

TABLE 3: Data reception result.

Type of data	Number of packets sent per second	Delay number	Packet loss rate
Data reception	100	0	0
	500	4	0
	1000	6	0
	1500	9	0

The recognition results are shown in Table 4: the recognition of abnormal data is basically accurate. For abnormal data processing, the found abnormal data will be completely deleted, and then the missing data will be filled together.

4.3. Edge Network Task Load Calculation. Figure 7 shows the overall load of the system in different scenarios when the number of users is different. It can be seen from the figure that the local execution method has the largest overall system load, while the other solutions reduce the overall load of the system compared to the local execution method, indicating that the execution of tasks on the cloud can obviously bring benefits to users. Among the remaining methods, the system load optimization method proposed in this chapter has the lowest overall system load, and the performance improvement effect is obvious. Compared with execution on the cloud, because execution on the cloud transmits all tasks to the cloud for execution without taking into account the mutual influence of users, the performance cannot be effectively improved.

Figure 8 shows the change in task load of each method under different task numbers. Likewise, the local execution method has the highest load. Like the previous analysis, when the number of tasks is less than a certain amount, the cloud execution method and the edge execution method have similar loads. However, as the number of tasks continues to increase, the edge execution method has a higher load than the cloud execution method due to the impact of delay, which also illustrates the important role of system load optimization in mobile edge computing.

4.4. Calculation and Analysis of Total Cost of Service Configuration. A good application configuration scheme not only ensures a low average response delay for service requests but also needs to consider the cost of providing service configuration in the edge network. This chapter further analyzes the optimization performance of the

TABLE 4: Abnormal data recognition rate.

Project	Precision	Recall rate
Behavior data	0.994	0.989

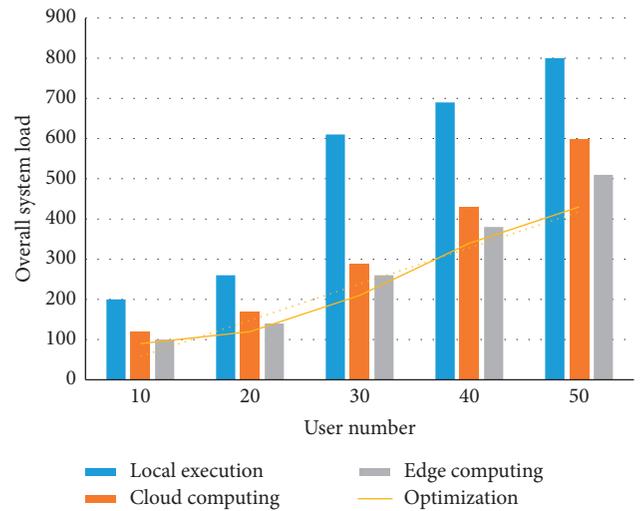


FIGURE 7: The average overall load of the system under different schemes.

application configuration schemes proposed by all algorithms in terms of service configuration cost.

Since the VRC configuration cost is a quadratic function of the number of VRCs supporting each application and the number of application types, it will increase rapidly as the number of application types increases. It can be seen from Figure 9 that when the number of VRCs is small, the total cost of the service configuration scheme proposed by all algorithms decreases with the increase of the number of VRCs. This is because when the number of VRCs is very small, the leading service configuration cost is the request response delay, cost.

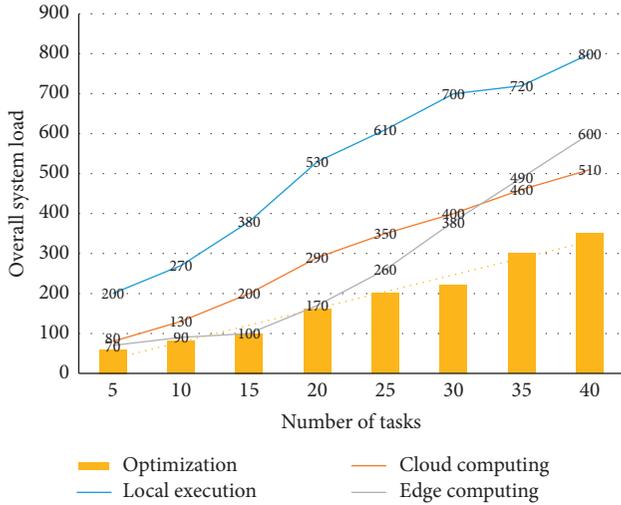


FIGURE 8: System task load changes under different task numbers.

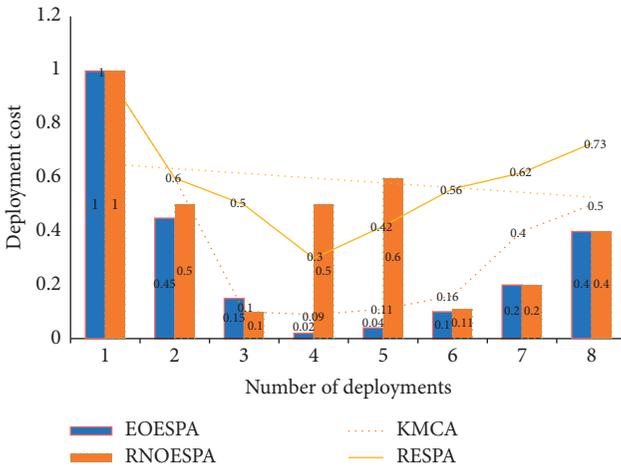


FIGURE 9: The impact of different quantities on the total cost of service configuration.

5. Conclusions

Based on the mobile edge computing architecture, this paper proposes an edge network service model that provides application services for mobile users. It not only analyzes the optimal deployment of edge servers and proposes corresponding solutions, but also studies a variety of modes in which edge network resources are limited. The problem of optimal configuration of application services in the edge network and corresponding solutions are proposed. A data collection and processing system based on edge computing is realized. Through the coordination and interaction between edge devices and cloud computing center the problems of the original system such as slow computing speed, large network delay, and large load are solved. The system fully integrates the latest information technology into the information collection of the dance training smart APP, so that users can use advanced methods such as the Internet to store ideas and promote the information and automation of

dance training management and its efficiency and efficiency, thereby helping to improve training level. The mobile information management system designed in this article still has shortcomings. It does not include all the content in the scope of dance training management. It only focuses on the selection of training plan management in training management as the core function of the system. Other aspects are only slightly involved and not in depth. To carry out a comprehensive functional design, further research is still needed.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Project of Hunan Provincial Social Science Achievement Review Committee, China (Grant no. XSP21YBC376), and the Excellent Youth Project of Department of Education of Hunan Province, China (Grant no. 20B395).

References

- [1] L. Yang, C. Xu, Y. Zhan et al., "Incentive mechanism for computation offloading using edge computing:A Stackelberg game approach," *Computer Networks*, vol. 129, no. 24, pp. 399–409, 2017.
- [2] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5994–6009, 2017.
- [3] B. P. Rimal, D. Pham Van, and M. Maier, "Mobile-edge computing versus centralized cloud computing over a converged FiWi access network," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 498–513, 2017.
- [4] C. Vallati, A. Viridis, E. Mingozzi, and G. Stea, "Mobile-Edge Computing Come Home Connecting things in future smart homes using LTE device-to-device communications," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 77–83, 2016.
- [5] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
- [6] Y. Wang, S. Min, X. Wang et al., "Mobile-edge computing: partial computation offloading using dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282, 2016.
- [7] Z. Ke, Y. Mao, S. Leng et al., "Mobile-edge computing for vehicular networks:A promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.
- [8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
- [9] N. Kryvinska and M. Gregus, "Agile information business," *Flexible Systems Management*, vol. 5, pp. 125–175, 2018.

- [10] Z. Yang, S. Deng, T. Gao et al., "Research on user needs for mobile information services in Chinese university libraries: Comparison between existing user and potential users," *Electronic Library*, vol. 34, no. 4, pp. 617–635, 2016.
- [11] Q. M. Ghyas and F. N. Kondo, "How do mobile information services improve quality of Life?The case of Japanese students," *International Journal of Innovation&Technology Management*, vol. 13, no. 6, pp. 64–70, 2016.
- [12] P. Yu, L. Feng, W. Li, and X. Qiu, "Energy-saving management mechanism based on hybrid energy supplies for LTE heterogeneous networks," *Mobile Information Systems*, vol. 2016, Article ID 3121538, 13 pages, 2016.
- [13] T. Arkadiusz, "The use of the mobile information and communication technologies in mass-casualty incident and disaster management-A medical triage system," *Prehospital and Disaster Medicine*, vol. 32, no. S1, pp. S239–S240, 2017.
- [14] M. R. Santos, L. J. d. Souza, E. W. Wildauer, S. B. D. Paula-Petroli, and F. E. Carrara-Marroni, "Mobile application proposal for mapping of multi-drug resistant bacteria information management," *International Journal for Innovation Education and Research*, vol. 8, no. 9, pp. 395–403, 2020.
- [15] R. W. Long, S. M. Yang, and B. M. Lee, "A mobile healthcare hospital service platform for personal health information management," *International Journal of Bio-Science and Bio-Technology*, vol. 8, no. 6, pp. 117–128, 2017.
- [16] T. Koç, A. H. Turan, and A. Okursoy, "Acceptance and usage of a mobile information system in higher education: an empirical study with structural equation modeling," *The International Journal of Management Education*, vol. 14, no. 3, pp. 286–300, 2016.
- [17] W. L. Shiau, C. M. Yan, and B. W. Lin, "Exploration into the intellectual structure of mobile information systems," *International Journal of Information Management*, vol. 47, pp. 241–251, 2019.
- [18] G. Wang, Z. Li, and Y. Li, "Design and implementation of mobile information system for electric power engineering environment evaluation water conservation," *Journal of Geomatics*, vol. 43, no. 5, pp. 100–103, 2018.
- [19] Y. Dong, M. Z. Hassan, J. Cheng, M. J. Hossain, and V. C. M. Leung, "An edge computing empowered radio access network with UAV-mounted FSO fronthaul and backhaul: key challenges and approaches," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 154–160, 2018.
- [20] L. Zhi, W. M. Wang, L. Guo et al., "Toward open manufacturing:A cross-enterprises knowledge and services exchange framework based on blockchain and edge computing," *Industrial Management & Data Systems*, vol. 118, no. 9, pp. 303–320, 2018.
- [21] S. Nastic, T. Rausch, O. Scekic et al., "A serverless real-time data analytics platform for edge computing," *IEEE Internet Computing*, vol. 21, no. 4, pp. 64–71, 2017.
- [22] M. Dias De Assunção, A. Da Silva Veith, and R. Buyya, "Distributed data stream processing and edge computing: a survey on resource elasticity and future directions," *Journal of Network and Computer Applications*, vol. 103, pp. 1–17, 2018.
- [23] G. Ananthanarayanan, P. Bahl, P. Bodík et al., "Real-time video analytics: the killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.
- [24] C. Peter and K. D. Soumya, "Mobile-edge computing and the Internet of things for consumers:extending cloud computing and services to the edge of the network," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, 2016.
- [25] X. Chen, Q. Shi, L. Yang, and J. Xu, "ThriftyEdge: resource-efficient edge computing for intelligent IoT applications," *IEEE Network*, vol. 32, no. 1, pp. 61–65, 2018.
- [26] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial Internet of things environment: software-defined-networks-based edge-cloud interplay," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 44–51, 2018.
- [27] E. Ahmed, A. Ahmed, I. Yaqoob et al., "Bringing computation closer toward the user network: is edge computing the solution?" *IEEE Communications Magazine*, vol. 55, no. 11, pp. 138–144, 2017.
- [28] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-centric edge computing based on virtualized D2D communication systems," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 32–38, 2018.
- [29] M. Gusev and S. Dustdar, "Going back to the roots-the evolution of edge computing, an IoT perspective," *IEEE Internet Computing*, vol. 22, no. 2, pp. 5–15, 2018.
- [30] F. Qiang and N. Ansari, "Application aware workload allocation for edge computing based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.