

Research Article

Application of Mobile Edge Computing Technology in Civil Aviation Express Marketing

Ying Yu 

Loughborough University (Bachelor in Shanghai Lixin University of Accounting and Finance), Zhaotong, 657000 Yunnan, China

Correspondence should be addressed to Ying Yu; 21027043@zju.edu.cn

Received 16 March 2021; Revised 6 May 2021; Accepted 18 May 2021; Published 1 June 2021

Academic Editor: Wenqing Wu

Copyright © 2021 Ying Yu. 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.

With the popularization of mobile terminals and the rapid development of mobile communication technology, many PC-based services have placed high demands on data processing and storage functions. Cloud laptops that transfer data processing tasks to the cloud cannot meet the needs of users due to low latency and high-quality services. In view of this, some researchers have proposed the concept of mobile edge computing. Mobile edge computing (MEC) is based on the 5G evolution architecture. By deploying multiple service servers on the base station side near the edge of the user's mobile core network, it provides nearby computing and processing services for user business. This article is aimed at studying the use of caching and MEC processing functions to design an effective caching and distribution mechanism across the network edge and apply it to civil aviation express marketing. This paper proposes to focus on mobile edge computing technology, combining it with data warehouse technology, clustering algorithm, and other methods to build an experimental model of MEC-based caching mechanism applied to civil aviation express marketing. The experimental results in this paper show that when the cache space and the number of service contents are constant, the LECC mechanism among the five cache mechanisms is more effective than LENC, LRU, and RR in cache hit rate, average content transmission delay, and transmission overhead. For example, with the same cache space, ATC under the LECC mechanism is about 4%~9%, 8%~13%, and 18%~22% lower than that of LENC, LRU, and RR, respectively.

1. Introduction

In recent years, the ability of humans to use computer technology to generate and collect information has been greatly improved. Large-scale database systems have been widely used in the management of research companies, and their development momentum is very strong. This raises a new question for managers, how can we effectively manage and apply a large amount of data? In particular, in today's highly competitive society, data should be "used" effectively and has been put on the agenda.

With the in-depth development of domestic and foreign markets, the direction of civil aviation express marketing has gradually shifted from "product driven" to "market driven" and "customer driven," which requires civil aviation express to adopt a marketing strategy centered on market demand. After gradually realizing the role of historical data resources in improving the competitiveness of enterprises and increasing economic benefits, they tried to obtain their

own sales data and sales data of competing companies to assist in the formulation of sales decision information and timely adjustment of sales strategies and focus on sales and improve the level of revenue. How to obtain useful data faster to help companies analyze the actual needs of customers? By providing a variety of layered and personalized service solutions, the civil aviation industry must give priority to increasing sales revenue and profits and improving customer satisfaction and loyalty. In the marketing process, in order to make accurate and timely business decisions, relevant information must be fully obtained and used to assist the decision-making process.

Mobile edge computing can provide cloud and IT services near mobile subscribers and allow cloud servers to be used inside or near base stations. Therefore, using the MEC platform can reduce the terminal delay perceived by mobile users; application developers can use real-time wireless access network information from MEC to provide context-aware services; MEC can also implement the execution of

computationally intensive applications. Yu introduced the MEC platform's architecture and described and realized the key functions of the above-mentioned functions, then investigated the relevant latest research results, and finally discussed and determined the MEC open research challenges [1]. The downside is that the detailed analysis of this article around MEC just stays at the technology itself, without extending it. Tran and Pompili consider an MEC-enabled multicell wireless network, where each base station (BS) is equipped with an MEC server, which can help mobile users perform computationally intensive tasks through task offloading. In order to maximize the user's task offloading revenue, the problem of joint task offloading and resource allocation is studied [2]. However, the problem under consideration involves joint optimization of task offloading decisions. Due to the combined nature of this problem, it is difficult and impractical to find the best solution for large-scale networks. Ahmed and Rehmani introduced and discussed the definition of MEC given by researchers, the opportunities that MEC brings, and the research challenges in the MEC environment. In addition, the motivation of MEC was highlighted by discussing various applications [3]. However, the discussions conducted by the institute are all theoretical knowledge, not combined with practice, and are rather vague.

The innovation of this article is (1) the introduction of big data technology into mobile edge caching, and by bringing cloud computing and cloud storage closer to the edge of the network, it can better cope with the impact of the rapid growth of data traffic on the network and provide a high-quality network service and user experience. (2) This paper proposes a collaborative caching mechanism based on machine learning under the distributed MEC service system. The mechanism uses the local caches of several MEC servers to form a cooperative cache domain as the overall structure and uses the migration learning method to predict the popularity of the content. Based on this prediction, an MEC with an optimization goal of minimizing the average content transmission cost is established.

2. Application Methods in Civil Aviation Express Marketing

2.1. Mobile Edge Computing Technology. As a new architecture to improve the efficiency of computing offload, mobile edge computing provides computing resources in the access network close to smart mobile devices [4, 5]. Mobile edge computing is the foundation of cloud computing. The quasi-cloud computing is located close to the local network where the data source is located, and the data will not be sent to the cloud as much as possible, so as to reduce transmission delay and network bandwidth costs as much as possible. Currently, mobile edge computing is widely used in various fields. In the communications field, network operators combine telecommunications network services with edge computing and provide services to users by arranging MEC servers (MEC servers) at access points (access point (AP)) or base stations (base station (BS)).

Mobile edge computing allows mobile terminals to transfer computing tasks to network edge nodes, such as wireless access points and base stations. The basic framework diagram of the mobile edge computing system is shown in Figure 1. Its design idea meets the needs of mobile terminals to expand computer functions and at the same time compensates for the long-term transmission delay of the platform [6].

It can be seen from Figure 1 that on the MEC platform, the edge computing server is installed on the side of the wireless access network, which greatly shortens the distance between it and the user equipment. In traditional cloud computing, data must pass through the wireless access network and the main network to reach the remote cloud server. After the cloud server processes the data, it returns the result to the user through a long and visible link [7]. It takes a lot of time to use the cloud platform to process this task. Due to the shorter transmission distance, MEC no longer needs to pass through large mobile links and the main network, thereby reducing delay costs. On the other hand, because the processing power of the server edge is much higher than that of the mobile device, the processing time of the task is greatly reduced.

The MEC technology and hierarchical microcloud (Cloudlet) technology that provide services such as computing and caching for users within the access range are both considered to be the natural evolution of mobile cloud technology. The MEC server can provide cloud services to mobile users within the coverage of its Wi-Fi access points. Currently, applications deployed on terminal devices have increasing demands for computing resources and storage resources; making offloading workloads deployed on mobile terminals to the cloud has become the most effective solution to the insufficient performance of terminal devices [8]. The MEC server brings traditional core cloud-centric computing closer to the edge of the network on the user side, accelerating content, services, and applications, thereby improving responsiveness from the edge. While providing users with a highly distributed computing environment, the MEC server can also perform special tasks that cannot be completed in the traditional Internet architecture, such as real-time big data analysis, perception performance optimization, and distributed cloud computing. For many emerging applications, such as intensive video coding and local streaming services, mobile operators can use MEC servers to work with core cloud systems. When the local MEC server cannot meet the needs of users or the computing power of the MEC server is insufficient to support when computing intensive tasks are performed by the mobile terminal, the MEC server can request computing resources from the core cloud and collaborate to complete the work [9, 10]. This MEC server and the core cloud data center cooperate to complete tasks, which can make up for the core cloud's long distance to users, long transmission delay, and small cloud Wi-Fi coverage. The MEC server is used to complete seamless task offloading and execution.

At present, the MEC technology has been recognized by the 5G PPP organization as one of the three key emerging technologies for 5G networks, including network

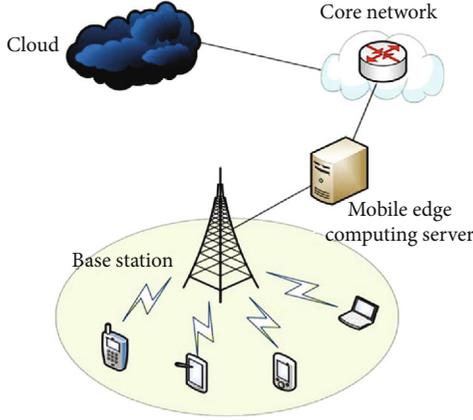


FIGURE 1: The basic framework of mobile edge computing system. (This picture is borrowed from Baidu Encyclopedia.)

function virtualization (NFV) and software-defined networking (SDN).

2.2. Data Warehouse Technology. Data warehouse is a new data processing architecture based on relational database technology to solve integrated data problems. The currently accepted concept of a data warehouse is the definition of the creator's data warehouse: a data warehouse is centralized, integrated, time-changing, and continuous data collection, supporting the execution of the decision-making process [11]. Regarding data warehouse, although domestic and foreign experts do not have a unified definition, their views all point out that data warehouse has the characteristics of subject-oriented, integrated, relatively stable, and historical.

2.2.1. Theme. Each topic corresponds to an analysis question raised by managers, which is an analysis field at the macro level. The theme refers to the abstract concept that aggregates and categorizes data in the macro field for mining, analysis, and utilization [12]. Since the data in the data warehouse is subject-oriented, there are two processes for the organization of the data: determining the subject and determining the content of the data that needs to be included in each subject. In a data warehouse, each topic corresponds to a relationship, that is, a data table, so the expression of the topic can be realized by using a data table in a relational database. Subject-oriented data organization has two characteristics: independence and completeness.

2.2.2. Granularity. Granularity refers to the level of detail and the level of the data warehouse in the data warehouse and is a very important concept in the data warehouse. The more detailed the description in the data warehouse, the lower the level of data analysis. On the other hand, the wider the description of the information, the higher the resolution. The data in the data warehouse usually has various resolution levels, including resolution level, resolution, openness, and advanced analysis. Analysis classification directly affects the amount of data in the repository and the adjustable data tables and numbers associated with the analysis process [13] as shown in Figure 2.

The data granularity of a data warehouse is related to the level of data collection organized in a specific topic. Data warehouse analysis must meet certain design principles to meet all levels of questionnaires and analysis requirements. Granular design principles are as follows: optimized storage structure, high query performance, space saving, and strive to eliminate data loss in the existing structure.

2.2.3. Metadata. Metadata is "data information" used to describe and identify the source of data elements, the activities of data elements in the process, the data warehouse, and the description of data and operations (input, calculation, and output). In order to effectively organize a data warehouse, metadata with clear description capabilities and rich content must be designed.

2.2.4. Data Segmentation. Data segmentation refers to a way to reasonably separate all data into small physical units suitable for independent storage management [14]. The main purpose of data segmentation is to improve the efficiency of organization and data analysis. There are many standards for data segmentation such as date, characteristics, region, and business area, or it can be a combination of two or more reference standards. Under normal circumstances, the date item should be regarded as the basic segmentation standard in the segmentation process. It can naturally segment the data according to people's understanding, and the segmentation is even. Data segmentation can be divided into two types: system level and application level [15]. System-level segmentation is done by most systems, such as database management and operating systems. Application-level segmentation is performed directly by password repository administrators or developers, and this level of segmentation is relatively more flexible.

2.3. Clustering Algorithm. Clustering algorithm is the most studied and widely used algorithm in unsupervised learning. Its basic function is to divide a specified data set into multiple subsets through a certain algorithm, that is, clusters [16, 17]. It can be said that clustering is a process of dividing samples in a data set into disjoint subsets through algorithms. The purpose of the clustering algorithm is to classify a group of data to form data clusters, and meet the following two conditions. (1) The tasks after the same cluster are as similar as possible. (2) The difference between tasks in different clusters is as large as possible.

The most commonly used criterion to determine the degree of similarity between points in the clustering algorithm is to use the Euclidean distance and Mahalanobis distance between two points [18, 19]. Generally, the Euclidean distance in the r -dimensional feature space can be expressed as

$$D_{m,n} = \sqrt{\sum_{i=1}^r (m_i - n_i)^2}. \quad (1)$$

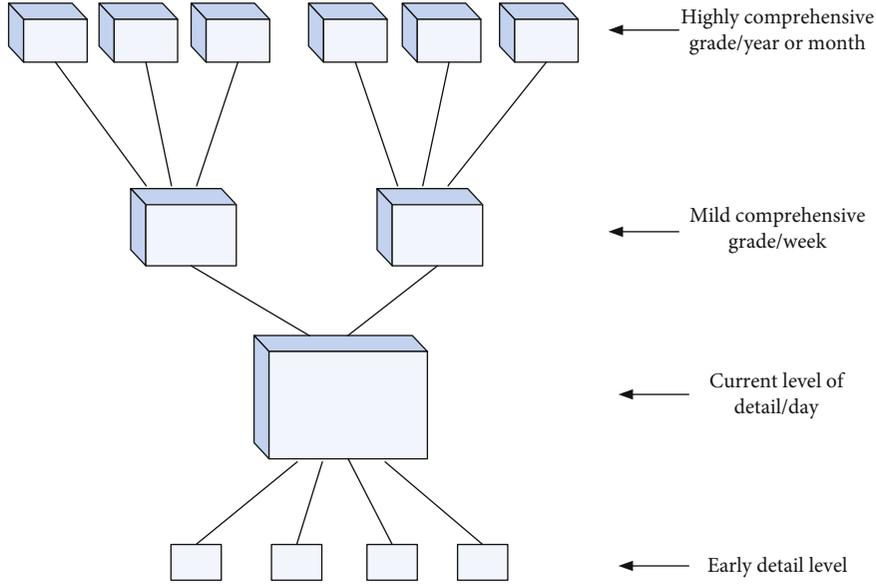


FIGURE 2: Data granularity in the data warehouse.

TABLE 1: Cache hit rate of MEC server mechanism in each cache space.

Cache space size	Caching mechanism	Cache hit rate (%)
20	GT	21.6
	LECC	20.3
	LENC	18.6
	LRU	11.8
	RR	7.5
40	GT	33.1
	LECC	31.5
	LENC	27.8
	LRU	21.7
	RR	15.1
60	GT	43.3
	LECC	39.8
	LENC	34.2
	LRU	28.1
	RR	21.9
80	GT	50.8
	LECC	47.6
	LENC	40.0
	LRU	36.7
	RR	28.8
100	GT	57.5
	LECC	54.8
	LENC	44.3
	LRU	42.1
	RR	34.9

The Mahalanobis distance can be expressed as

$$D_{(m,n)} = \sum_{i=1}^r |m_i - n_i|. \quad (2)$$

Ming's distance is a generalization of Mahalanobis distance, which can be expressed as

$$D_{(m,n)} = \left(\sum_{i=1}^r |m_i - n_i|^\lambda \right)^{1/\lambda}. \quad (3)$$

It can be seen that for Ming's distance, when $\lambda = 1$, it corresponds to the Mahalanobis distance, and when $\lambda = 2$, it corresponds to the Euclidean distance.

For the clustering problem in the MEC system, we need to determine the similarity metric before classification. Suppose there are currently X tasks, numbered $\{A_1, A_2, A_3, \dots, A_X\}$, and the total amount of calculation required to complete each task is $\{B_1, B_2, B_3, \dots, B_X\}$, respectively. Assuming that the sensitivity of all tasks to delay can be divided into Y categories, numbered $\{1, 2, 3, \dots, Y\}$, when there are X tasks. The extension sensitivity can be expressed as $\{E_1, E_2, E_3, \dots, E_X\}$, where $E_i \in \{1, 2, 3, \dots, X\}$, $i = \{A_1, A_2, A_3, \dots, A_X\}$. We define the classified cluster as $\{C_1, C_2, C_3, \dots, C_j\}$ and hope that the maximum value of the standard deviation of all clusters obtained after classification is as small as possible.

Assume that at a certain moment the task in the controller is $\{A_1^*, A_2^*, \dots, A_x^*\}$, and the corresponding delay sensitivity is $\{E_1', E_2', \dots, E_x'\}$. After normalizing the delay sensitivity, the delay sensitivity obtained is $\{E_1^*, E_2^*, \dots, E_x^*\}$, where

$$E_i^* = \frac{E_i'}{\sum_k^x E_k'}. \quad (4)$$

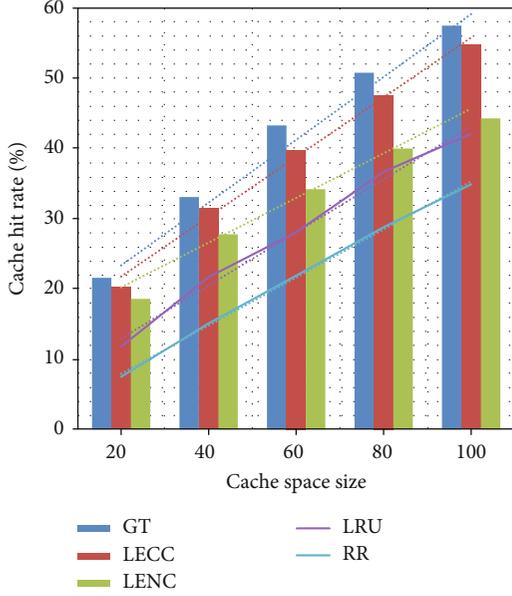


FIGURE 3: The trend of the cache hit rate with the size of the MEC cache space under each cache mechanism.

The total amount of calculation required to complete the task is $\{B_1', B_2', \dots, B_x'\}$, respectively, and the total calculation amount obtained after normalizing the calculation amount is $\{B_1^*, B_2^*, \dots, B_x^*\}$, where

$$B_i^* = \frac{B_i'}{\sum_{k=1}^x B_k'} \quad (5)$$

We define that the Euclidean distance between tasks A_i^* and A_k^* in the same cluster can be expressed as

$$D_{i,k} = \sqrt{(E_i^* - E_k^*)^2 + (B_i^* - B_k^*)^2} \quad (6)$$

The average of the distances in the same cluster can be expressed as

$$\bar{D} = \frac{\sum_{i=1}^x \sum_{k=1, k \neq i}^x D_{i,k}}{C_x^2} \quad (7)$$

Define the standard deviation corresponding to tasks in the same cluster as W , when there is task $\{A_1^*, A_2^*, \dots, A_x^*\}$ in cluster C_i ; the corresponding standard deviation is

$$W_i = \sqrt{\frac{\sum_{i=1}^{x'} \sum_{k=1, k \neq i}^{x'} (D_{i,k} - \bar{D})^2}{C_{x'}^2}} \quad (8)$$

We believe that in the same cluster, the smaller the standard deviation of all tasks, the higher the similarity of the tasks. Therefore, we hope that in the results obtained by clustering, the maximum value of the standard deviations in all

TABLE 2: The average content transmission delay of the MEC server mechanism in each cache space.

Cache space size	Caching mechanism	Average content transmission delay
20	GT	78.1
	LECC	79.7
	LENC	81.5
	LRU	88.6
	RR	92.2
40	GT	66.7
	LECC	69.3
	LENC	73.2
	LRU	79.1
	RR	85.2
60	GT	57.0
	LECC	60.8
	LENC	66.8
	LRU	72.1
	RR	79.2
80	GT	49.3
	LECC	53.9
	LENC	60.0
	LRU	64.1
	RR	72.2
100	GT	42.8
	LECC	45.7
	LENC	56.4
	LRU	58.9
	RR	66.7

clusters is as small as possible. That is, when all tasks can be divided into j clusters, we hope to get

$$\min \{\max W_i\}, \quad i = 1, 2, 3, \dots, j. \quad (9)$$

Assuming that there are a total of f' tasks in the current i -th cluster, then the average of the distances of all tasks in this cluster is

$$\bar{D}_i = \frac{\sum_{i=1}^{f'} \sum_{k=i}^{f'} D_{i,k}}{C_{f'}^2}, \quad (10)$$

where $C_{f'}^2$ means

$$C_{f'}^2 = \frac{f'!}{f'! \cdot (f' - 2)!} \quad (11)$$

The weight vector w_i of the currently selected sample i , where $w_k = (w_{k,1}, w_{k,2})$ and w_k represent the weight vector between the current input sample in the output layer.

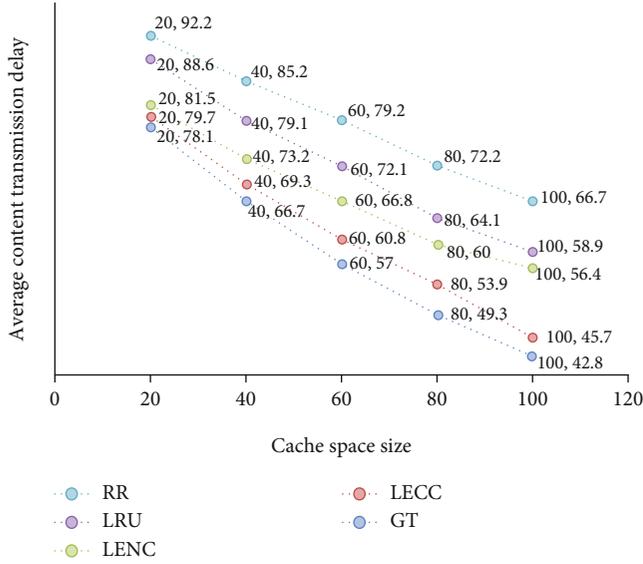


FIGURE 4: The average content transmission delay under each caching mechanism varies with the size of the MEC cache space.

TABLE 3: The average content transmission overhead of the MEC server mechanism in each cache space.

Cache space size	Caching mechanism	Average content transfer overhead
20	GT	376
	LECC	392
	LENC	409
	LRU	442
	RR	467
40	GT	334
	LECC	347
	LENC	365
	LRU	387
	RR	425
60	GT	281
	LECC	302
	LENC	327
	LRU	359
	RR	390
80	GT	244
	LECC	261
	LENC	301
	LRU	322
	RR	358
100	GT	213
	LECC	236
	LENC	278
	LRU	290
	RR	327

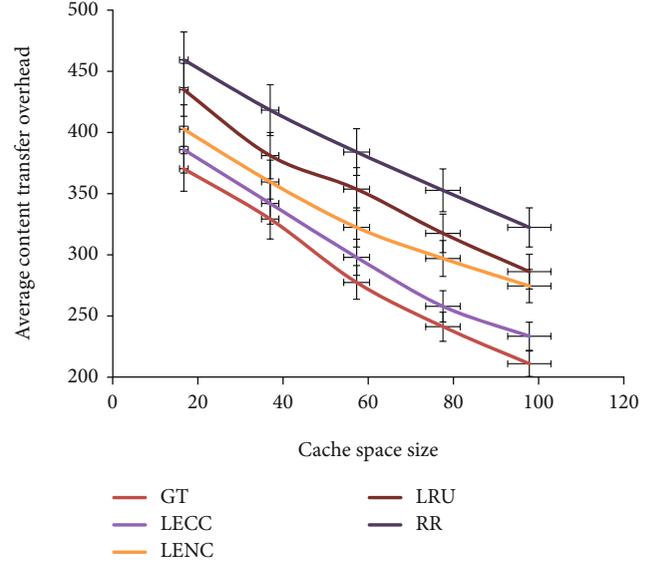


FIGURE 5: The trend of cache hit rate with the size of MEC cache space under each cache mechanism.

Therefore, the Euclidean distance vector D_i between the weight vector and the input M_i can be obtained by the following formula:

$$D_{i,k} = \|M_i - w_k\| = \left[\sum_{\lambda=1}^2 (m_{i,\lambda} - w_{k,\lambda})^2 \right]^{1/2}, \quad k = 1, 2, \dots, j. \quad (12)$$

Determine the winning neuron h by the smallest distance between the output layer neuron and the task sample i :

$$D_{i,h} = \min \{D_{i,k}\}, \quad k = 1, 2, \dots, j. \quad (13)$$

Adjust the connection weight: for the winning neuron h and all the neurons in the output layer and input layer in its neighborhood $N_h(n)$, the connection weight vector of the algorithm executed at the $n+1$ th time is in accordance with the formula shown below, make corrections

$$w_{i,k}(n+1) = w_{i,k}(n) + \eta(n) \times (M_i - w_{i,k}(n)), \quad k = h, \quad (14)$$

$$w_{i,k}(n+1) = w_{i,k}(n) + \frac{\eta(n)}{2} \times (M_i - w_{i,k}(n)), \quad k \in N_h(n), \quad (15)$$

$$w_{i,k}(n+1) = w_{i,k}(n), \quad k \notin N_h(n). \quad (16)$$

The connection weight of the input neuron i and the output neuron k is also a two-dimensional vector, and h represents the winning neuron.

Update the learning rate $\eta(n)$ and the neighborhood function $N_h(n)$: the general learning rate and neighborhood

TABLE 4: Cache hit rate of MEC server mechanism under different service contents.

Number of service contents Cache hit rate (%) Caching mechanism	GT	LECC	LENC	LRU	RR
1000	37.2	35.6	30.4	24.9	19.3
2000	26.3	24.8	19.8	14.9	9.8
3000	21.2	18.7	16.2	10.5	7.2
4000	18.5	15.8	12.1	7.9	5.1

function will gradually decrease with the increase of the number of iterations, which can be expressed as

$$\eta(n+1) = \eta(0) \times \left(1 - \frac{n}{N}\right), \quad (17)$$

$$N_h(n+1) = N_h(0) \times \left(1 - \frac{n}{N}\right). \quad (18)$$

The clustering algorithm is a strategy for clustering mobile terminal tasks in the MEC system. For the results obtained after clustering, it is hoped that the maximum distance standard deviation in all clusters is as small as possible [20]. It can not only ensure the quality of service and improve the user experience but also reduce the queuing delay of subsequent tasks, which is conducive to reducing system overhead.

3. Application Experiment Based on Mobile Edge Computing Technology in Civil Aviation Express Marketing

3.1. Experimental Program Based on MEC Architecture. In order to verify the effectiveness of the prediction of content popularity to design the collaborative caching mechanism under the MEC architecture, this article compares the caching scheme with some known content caching schemes, including the least recently used (LRU) strategy and randomized cache strategy (randomized replacement (RR)). The least recently used (LRU) replacement strategy is a caching strategy for estimating future requested content by observing the content being accessed by users in the near future. When the amount of cached data exceeds the threshold, the least recently accessed content is deleted [21, 22]. The RR caching strategy randomly selects data content for caching.

In addition, in order to more intuitively and effectively illustrate the advantages of the content popularity prediction scheme based on migration learning proposed in this paper and the improvement of network performance by the MEC collaborative caching mechanism, this research also combines the caching scheme with LENC (noncooperative learning). Based caching scheme and GT (popularity-aware greedy strategy) scheme are compared. Specifically, the LENC scheme means that each MEC directly caches content with high popularity based on the prediction results of the content popularity based on the migration learning method in this article, until the cache space is full, that is, no collabo-

rative caching is performed between MEC servers; the GT scheme is when the true value of content popularity is known in advance; the MEC collaborative greedy algorithm proposed in this paper is used for cache deployment [23]. In fact, the GT algorithm provides the performance upper limit (not the real performance upper limit) under the greedy cooperative caching mechanism of this article for the other four algorithms (LECC, LENC, LRU, and RR).

3.2. Performance Indicators. The comparison is made from three performance index parameters, namely, the cache hit rate (hit rate (HR)), the average content transmission delay (average delivery latency (ADL)), and the average content transmission cost (average transmission cost (ATC)).

3.2.1. Cache Hit Rate (HR). In the MEC service system, the processing methods of user requests can be divided into the following two situations: one is that if the requested content has a backup in the MEC cooperative cache domain; the content is sent to the user, which we call a cache hit. The other is that if the content requested by the user is not cached in the local cooperative cache domain; the content needs to be obtained from the remote central server and then pushed to the user. This is called a miss [24–26].

3.2.2. Average Content Transmission Delay (ADL). In the MEC service system, the average delay for users to obtain content is an important performance indicator to measure the quality of user experience. The smaller the average content delivery delay, the more user requests can be satisfied by the local MEC, and the higher the quality of user experience [27, 28].

3.2.3. Average Content Transmission Cost (ATC). The average content transmission cost (ATC) index is the value of the objective function of the optimization problem. In addition, in the definition of the objective function of the optimization problem, we set the unit data content to be transmitted from C_m to C_n ($m \neq n$), and the transmission cost through single-hop routing is γ . The unit data content is transmitted from the remote cache server to C_m , and the transmission cost of the single-hop route is γ_0 . Since the exact values of γ and γ_0 cannot be known in the simulation of real test scenarios, we use simulation to illustrate the impact of γ and γ_0 value settings on performance. We define the cost factor μ :

$$\mu \stackrel{\text{def}}{=} \frac{\gamma_0}{\gamma}. \quad (19)$$

3.3. Experimental Parameter Settings. Consider the scenario where video content services are dominant in mobile Internet applications. The simulation parameters are as follows. There are 3 randomly distributed BSs in the cooperative buffer domain, and the number of mobile users in each cell is 400. Assume that the content provider publishes a total of 800 video files whose popularity obeys the ZipF distribution model, and the skewness coefficient is $\alpha = 0.52$. This parameter describes the degree of skewness of the content popularity distribution. For the cooperative cache domain model, we consider 3 MEC servers in a domain, and each server receives

TABLE 5: The average content transmission delay of the MEC server mechanism under different service content quantities.

Number of service contents	GT	LECC	LENC	LRU	RR
Average content transmission delay					
Caching mechanism					
1000	61.8	64.9	69.8	75.7	82.1
2000	74.8	77.2	80.6	85.6	90.3
3000	79.7	82.8	85.1	89.1	94.0
4000	82.3	84.9	88.7	92.5	95.1

TABLE 6: The average content transmission overhead of the MEC server mechanism under different service content quantities.

Number of service contents	GT	LECC	LENC	LRU	RR
Average content transfer overhead					
Caching mechanism					
1000	312	325	349	375	413
2000	371	383	402	425	451
3000	398	412	422	445	470
4000	415	423	440	462	476

an average of 9000 content requests per day. It is assumed that the MEC server in the domain has a uniform size of cache space. The update period Δt is 3 h. In addition, the transmission delay in the access network and core network refers to the 3GPP LTE standard. In the simulation experiment, we study the impact of system parameters on cache performance, including the size of cache space, the number of service contents, the skewness coefficient of the popularity distribution, and the cost coefficient [29, 30].

4. Application Experiment Analysis Based on Mobile Edge Computing Technology in Civil Aviation Express Marketing

4.1. Performance Comparison of Five Mechanisms in a Certain Cache Space. First, the five caching mechanisms under the change of the MEC cache space are described and the performance comparison in three aspects: cache hit rate, average content transmission delay, and transmission overhead. The value of each MEC buffer space size ranges from 10 to 100, the overhead coefficient γ is 10, the skew coefficient α is 0.52, and the number of service contents in the system is 800. Table 1 records the variation of the cache hit rate with the size of the MEC cache space under each cache mechanism.

According to the variation of the cache hit rate with the size of the MEC cache space under each cache mechanism recorded in Table 1, the trend change graph of the cache hit rate with the size of the MEC cache space under each cache mechanism in Figure 3 can be obtained.

It can be seen from Figure 3 that the cache hit rate under all strategies increases as the cache space increases. Experimental data shows that the LECC mechanism proposed in this paper is superior to LENC, LRU, and RR. Specifically, when the cache space is 60, the cache hit rate under the LECC mechanism is about 6%, 11%, and 18% higher than that of

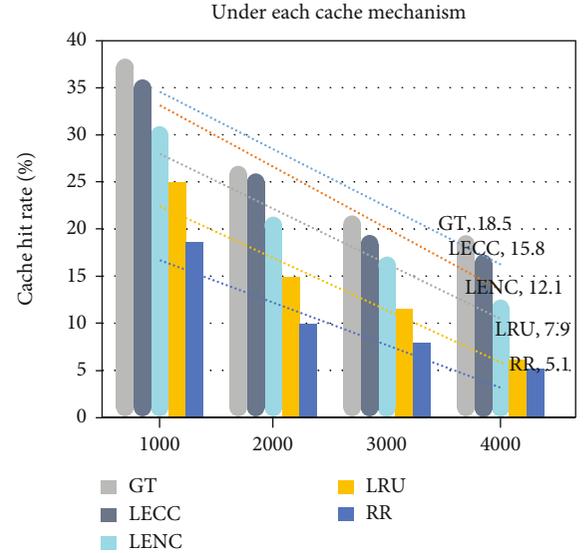


FIGURE 6: Cache hit rate of MEC server mechanism under different service contents.

LENC, LRU, and RR, respectively. In addition, when the size of the cache space changes from 20 to 100, the cache hit rate under the LECC mechanism differs from the upper limit of the cache hit rate obtained by GT by only 3%-5%. This proves the effectiveness of the caching mechanism based on intelligent prediction of content popularity in this paper.

Table 2 records the variation of the average content transmission delay with the size of the MEC cache space under each caching mechanism.

According to the variation of the average content transmission delay under each cache mechanism with the size of the MEC cache space recorded in Table 2, the trend change graph of the cache hit rate under each cache mechanism with the size of the MEC cache space can be obtained.

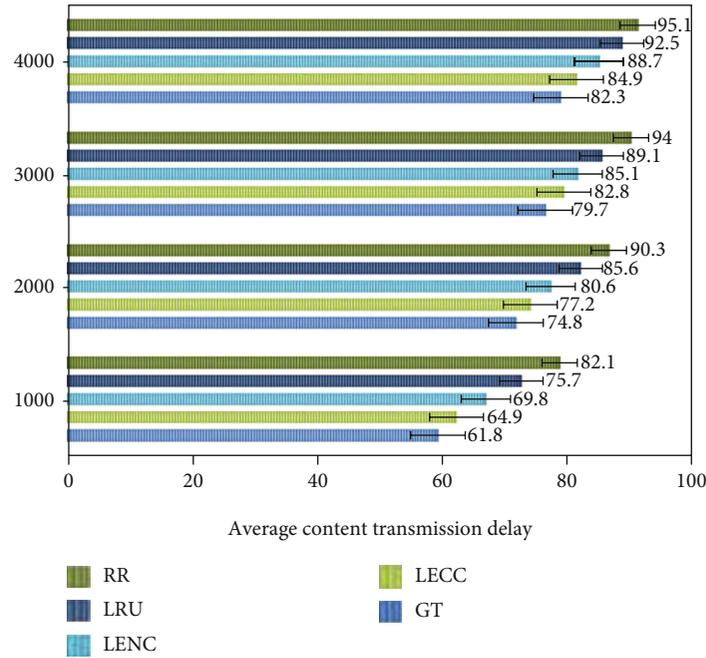


FIGURE 7: The average content transmission delay of the MEC server mechanism under different service content quantities.

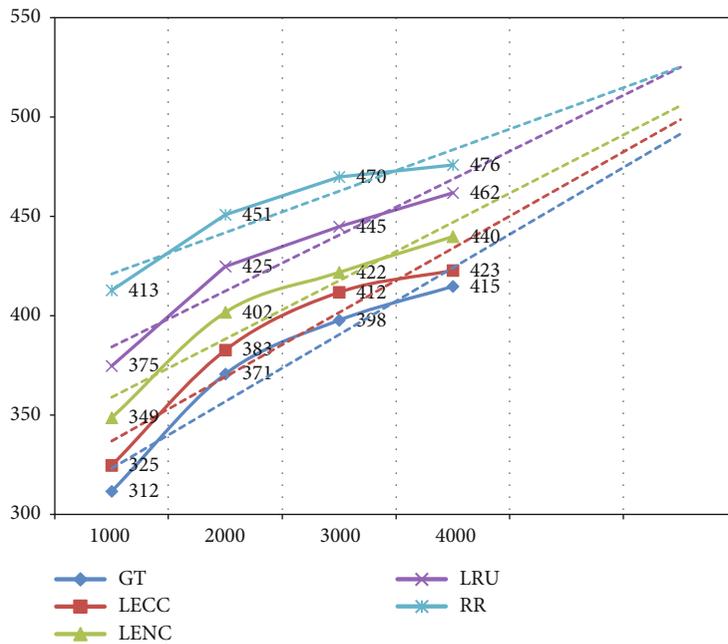


FIGURE 8: The average content transmission overhead of the MEC server mechanism under different service content quantities.

It can be seen from Figure 4 that the average content transmission delay under all strategies decreases as the cache space increases. This is because as the cache space increases, more content can be cached locally so that more user requests can be directly served by the local cooperative cache domain. Experimental data shows that the LECC mechanism has the lowest ADL. For example, when the storage space ranges from 60 to 100, the ADL under the LECC mechanism is about 6% to 10%, 10% to 15%, and 18% to 22% lower than that of LENC, LRU, and RR, respectively.

Table 3 records the variation of the average content transmission overhead with the size of the MEC cache space under each caching mechanism [31, 32].

According to the changes in the average content transmission overhead of each cache mechanism with the size of the MEC cache space recorded in Table 3, the trend graph of the cache hit rate with the size of the MEC cache space under each cache mechanism in Figure 5 can be obtained.

It can be seen from Figure 5 that the average content transmission overhead under all strategies decreases as the

cache space increases. Experimental data shows that the LECC mechanism has the lowest ATC. For example, when the storage space ranges from 60 to 100, the ATC under the LECC mechanism is about 4%-9%, 8%-13%, and 18%-22% lower than that of LENC, LRU, and RR, respectively. The significant improvement in cache performance brought about by the LECC mechanism mainly comes from the content popularity prediction based on migration learning proposed in this paper and the MEC collaborative cache [33].

4.2. Performance Comparison of Five Mechanisms under a Certain Amount of Service Content. Tables 4–6 record the cache hit rate, average content transmission delay, and transmission overhead under the five caching mechanisms as the number of different service contents changes. The value range of the number of service contents is 500 to 4000, the cost coefficient γ is fixed at 10, the skew coefficient α is taken at 0.52, and the size of the cache space of each MEC server is fixed at 60.

According to the changes in the cache hit rate of each cache mechanism with the number of different MEC service contents recorded in Table 4, the trend change graph of the cache hit rate with the size of the MEC cache space under each cache mechanism in Figure 6 can be obtained.

According to the variation of the average content transmission delay under each caching mechanism with the amount of different MEC service content recorded in Table 5, the trend change graph of the average content transmission delay under each caching mechanism in Figure 7 with the size of the MEC cache space can be obtained.

According to the variation of the average content transmission cost under each caching mechanism with the number of different MEC service contents recorded in Table 6, the trend graph of the average content transmission cost under each caching mechanism in Figure 8 with the size of the MEC cache space can be obtained.

Figures 6–8 describe the changes in the cache hit rate, average content transmission delay, and transmission overhead with the number of different service contents under the five caching mechanisms. Experimental data shows that the LECC mechanism proposed in this paper is superior to LENC, LRU, and RR in terms of HR, ADL, and ATC. It can be seen from Figure 6 that the cache hit rate under all strategies decreases as the number of service contents increases. The average content transmission delay and transmission overhead increase with the increase in the number of service contents. This is due to the limited cache space of the MEC server. With the continuous increase of service content, more and more user requests cannot be satisfied from the local cache. From the numerical results in Figure 6, when the number of service content changes from 1000 to 3000, the HR under the LECC mechanism is about 5%~9%, 11%~14%, and 16%~21% higher than that in LENC, LRU, and RR, respectively.

5. Conclusions

With the rapid development of the mobile Internet and the Internet of Things, mankind is about to usher in the 5G

era. 5G technology demands “large capacity, large bandwidth, low latency, and low power consumption,” and mobile edge computing is precisely the 5G network that improves user experience. The key technology is deployed at the edge of the network, close to the data source, and has rapid feedback. It can sink the computing content and capabilities and localize the business. This paper proposes a collaborative caching mechanism based on machine learning under the 5G MEC architecture. The mechanism uses the local caches of several MEC servers to form a cooperative cache domain as the overall structure and uses the migration learning method to predict the popularity of the content. Based on this prediction, an MEC is established that optimizes the average content transmission cost. The content cache model further proves the above optimization problem. We use algorithms to solve the content caching scheme and perform simulation verification. Experimental results show that the proposed LECC caching strategy can effectively improve the cache hit rate and reduce transmission overhead and content transmission delay.

The proposed mobile edge computing architecture provides storage and computing and network services for mobile terminal users and provides a good platform for task migration. Task migration is to transfer all or part of large-capacity PC applications to mobile devices with a large amount of resources on the platform to deal with insufficient processing problems and limited power. The short-term capabilities of mobile edge computing provide users with powerful computing capabilities and ultra-high-speed networks, which can be accessed anytime, anywhere, shorter migration path and lower power consumption.

Data Availability

No data were used to support this study.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Y. Yu, “Mobile edge computing towards 5G: vision, recent progress, and open challenges,” *China Communications*, vol. 13, Supplement2, pp. 89–99, 2016.
- [2] T. X. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.
- [3] E. Ahmed and M. H. Rehmani, “Mobile edge computing: opportunities, solutions, and challenges,” *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2016.
- [4] N. Ansari and X. Sun, “Mobile edge computing empowers Internet of things,” *IEICE Transactions on Communications*, vol. 101, no. 3, pp. 604–619, 2018.
- [5] W. Li, Z. Chen, X. Gao, W. Liu, and J. Wang, “Multimodel framework for indoor localization under mobile edge computing environment,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4844–4853, 2019.

- [6] S. Wang, Y. Zhao, L. Huang, J. Xu, and C. H. Hsu, "QoS prediction for service recommendations in mobile edge computing," *Journal of Parallel & Distributed Computing*, vol. 127, pp. 134–144, 2017.
- [7] J. Zeng, J. Sun, B. Wu, and X. Su, "Mobile edge communications, computing, and caching (MEC3) technology in the maritime communication network," *China Communications*, vol. 17, no. 5, pp. 223–234, 2020.
- [8] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, "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.
- [9] T. Wang, H. Luo, X. Zheng, and M. Xie, "Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, pp. 1–19, 2019.
- [10] Q. V. Pham, L. B. Le, S. H. Chung, and W. J. Hwang, "Mobile edge computing with wireless backhaul: joint task offloading and resource allocation," *IEEE Access*, vol. 7, no. 99, pp. 16444–16459, 2019.
- [11] J. C. Yim, S. H. Kim, and C. S. Keum, "Personalized service recommendation for mobile edge computing environment," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 42, no. 5, pp. 1009–1019, 2017.
- [12] 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.
- [13] Y. Jararweh, M. Al-Ayyoub, M. Al-Quraan, A. T. Lo'ai, and E. Benkhelifa, "Delay-aware power optimization model for mobile edge computing systems," *Personal & Ubiquitous Computing*, vol. 21, no. 6, pp. 1–11, 2017.
- [14] M. Zeng and V. Fodor, "Energy minimization for delay constrained mobile edge computing with orthogonal and non-orthogonal multiple access," *Ad Hoc Networks*, vol. 98, pp. 102060.1–102060.13, 2020.
- [15] Y. Zhai, T. Bao, L. Zhu, M. Shen, X. du, and M. Guizani, "Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 84–91, 2020.
- [16] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic task offloading and resource allocation for mobile edge computing in dense cloud RAN," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3282–3299, 2020.
- [17] J. Zhang, H. Guo, and J. Liu, "Adaptive task offloading in vehicular edge computing networks: a reinforcement learning based scheme," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1–10, 2020.
- [18] R. Fantacci and B. Picano, "Federated learning framework for mobile edge computing networks," *CAA Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 15–21, 2020.
- [19] N. Saranya, K. Geetha, and C. Rajan, "Data replication in mobile edge computing systems to reduce latency in Internet of things," *Wireless Personal Communications*, vol. 112, no. 4, pp. 2643–2662, 2020.
- [20] W. Wen, Y. Cui, T. Q. S. Quek, F. C. Zheng, and S. Jin, "Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7879–7894, 2020.
- [21] Y. Liu, Y. Li, Y. Niu, and D. Jin, "Joint optimization of path planning and resource allocation in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2129–2144, 2020.
- [22] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: a load-balancing solution," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2092–2104, 2020.
- [23] J. Shen, Y. Ren, J. Wan, and Y. Lan, "Hard disk drive failure prediction for mobile edge computing based on an LSTM recurrent neural network," *Mobile Information Systems*, vol. 2021, Article ID 8878364, 12 pages, 2021.
- [24] J. Ahn, J. Lee, S. Park, and H. S. Park, "Power efficient clustering scheme for 5G mobile edge computing environment," *Mobile Networks and Applications*, vol. 24, no. 2, pp. 643–652, 2019.
- [25] X. Meng, W. Wang, Y. Wang, V. K. N. Lau, and Z. Zhang, "Closed-form delay-optimal computation offloading in mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4653–4667, 2019.
- [26] Z. Lv and Q. Liang, "Optimization of collaborative resource allocation for mobile edge computing," *Computer Communications*, vol. 161, pp. 19–27, 2020.
- [27] J. Xue, Z. Wang, Y. Zhang, and L. Wang, "Task allocation optimization scheme based on queuing theory for mobile edge computing in 5G heterogeneous networks," *Mobile Information Systems*, vol. 2020, Article ID 1501403, 12 pages, 2020.
- [28] C. H. Wu and S. B. Tsai, "Using DEMATEL-based ANP model to measure the successful factors of E-commerce," *Journal of Global Information Management*, vol. 26, no. 1, pp. 120–135, 2018.
- [29] S. Wang, Q. Li, J. Hou, S. Meng, B. Zhang, and C. Zhou, "Active defense by mimic association transmission in edge computing," *Mobile networks & applications*, vol. 25, no. 2, pp. 725–742, 2020.
- [30] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3052–3056, 2019.
- [31] S. Li, D. Zhai, P. Du, and T. Han, "Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks," *Science China Information Sciences*, vol. 62, no. 2, pp. 1–3, 2019.
- [32] M. Abdel-Basset, G. Manogaran, and M. Mohamed, "A neutrosophic theory based security approach for fog and mobile-edge computing," *Computer Networks*, vol. 157, pp. 122–132, 2019.
- [33] B. Zhu, S. Ma, R. Xie, J. Chevallier, and Y. Wei, "Hilbert spectra and empirical mode decomposition: a multiscale event analysis method to detect the impact of economic crises on the European carbon market," *Computational Economics*, vol. 52, no. 1, pp. 105–121, 2018.