A Co-caching Strategy for Edges Based on Federated Learning and Regional Prevalence

Zhirong Zhu[1], Yiwen Liu[1,2,3(✉)], Yanxia Gao[1,2,3], Wenkan Wen[1], Yuanquan Shi[1,2,3], and Xiaoning Peng[1,2,3]

[1] School of Computer and Artificial Intelligence, Huaihua University, Huaihua 418000, Hunan, People's Republic of China
lyw@hhtc.edu.cn

[2] Key Laboratory of Wuling-Mountain Health Big Data Intelligent Processing and Application in Hunan Province Universities, Huaihua 418000, Hunan, People's Republic of China

[3] Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, Huaihua 418000, Hunan, People's Republic of China

Abstract. With the rise of data storage computing and IoT technology. The increase in data volume and user demand, the accurate delivery of data and low latency during transmission become important factors that affect the end-user experience. To address this issue, previous authors have proposed the concept of edge computings. In the general environment of edge computing, reasonable scheduling of edge caches can largely achieve low latency and high efficiency, thus improving user experience. In this paper, based on existing research, we propose a combination of a joint learning framework for cache prediction based on region popularity and an edge collaborative cache value optimization method to further improve cache hit rate and cache utilization efficiency. The method obtains excellent expected results through simulation experiments.

Keyword: Data Storage Computing · IoT Technology · Edge Computing · Cache Hit Rate

1 Introduction

With the advent of wireless network services and the era of Big Data Internet. Tens of thousands of users have chosen to access the Internet. And with the popularity of network technology, the corresponding data volume is increasing and the data type is complicated. And because of the previous cloud storage computing using a single form of storage distribution. This makes the server load too large, resulting in long latency and poor user experience. In order to solve this problem, the concept of edge computing has emerged in recent years to address this status quo [1].

Edge computing is a computing facility where data resources are deployed in the data center department and close to the user, and through this facility, a series of network devices link the edge computing devices to the user or process. This is how the Internet of

Z. Xiao et al. (Eds.): ICECI 2022, LNICST 478, pp. 291–307, 2023.
https://doi.org/10.1007/978-3-031-28990-3_20

Things and Smart Networks are born. [2]. The previous edge computing model and due to the open nature of edge computing, data privacy security is also a major challenge that we need to address. It also affects the efficiency of data transfer in the face of too large amount of data. Both are caused by the improper allocation of edge cache. And according to related studies [3, 4] in the Internet in the same type or region of the population will have roughly similar search content and browsing trends. Then this means that we can cache the data that users search frequently and may be hotly searched in advance in the user's edge device by using edge caching wisely. The content that may not be of interest to the user will be reduced or stopped. Thus, we can solve the large data traffic congestion to a certain extent and thus reduce the latency.

Under this hypothesis. In this paper, we categorize the population of possible viewers [5] with the same attributes based on machine learning and try to segment the regions according to the browsing popularity. The segmentation signal is transmitted to the corresponding edge cache controller. It then filters and distributes the content to be cached. And to ensure further improvement in accuracy and optimization of performance. In this paper, we use federation learning for data localization training, and the training process is shown in Fig. 1. The trained primitive models localized by each edge device are uploaded to an aggregation center consisting of collection base stations and aggregators. The aggregation center aggregates the sub-models to form a combined model. The combined model is then distributed by the collection base station to each edge device for the next round of training tests. The aggregation cycle results in a more accurate prediction model that fits the user's needs. Only parameters are transmitted between the base station and the edge devices. Thus, the amount of data transmission is reduced. And to a certain extent, the privacy and security of the data are guaranteed and the overall performance is optimized. And the strategy designed in this paper is experimentally verified to obtain better optimization results than the comparison model. The feasibility of the design is proved.

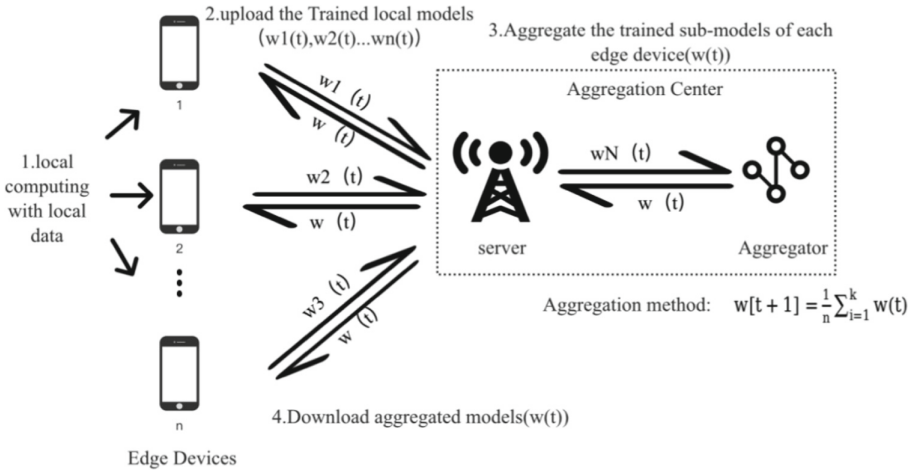

Fig. 1. Federal learning model training method under edge-based computing model

2 Relevant Domestic and International Studies

In the literature [2] the authors categorize and study the breakthroughs in edge computing technologies in recent years. And the study of edge caching technology becomes the most important and at the same time the most critical part of the optimized edge computing model.

A task scheduling based cache placement user approach is proposed in the literature [6] by starting from the cache scheduling mechanism. In contrast, the literature [7] builds a user-object-aware network. The user's perception of the network information is analyzed to filter the cache content, thus reducing information congestion. In contrast, the literature [8–10] preempts data with high cache demand into the edge cache pool by building a new secure resource allocation algorithm. And the data with small cache demand is stored in the data center. This can reduce the data congestion caused when the data volume is too large to some extent. At the same time, the security lock and key encryption are combined to ensure the security of data privacy.

In recent years, with the development of machine learning, deep learning and other content mining. It has led to a new foothold in the optimization of edge cache. For example, the literature [11] combines machine learning with statistical methods to derive a reference popularity scheme for cache storage. In contrast, the literature [12, 13] uses a fusion of machine learning and reinforcement learning to derive an optimal solution to the cache allocation scheme. Thus, it is demonstrated that artificial intelligence enables edge computing technology to step into a new level.

And if the cache hit rate is to be as high as possible. Then from the experience point of view. Then the preference data classes of edge end-users need to be taken into account. The literature [14] combines user and user data and analyzes its correlation to derive the differences in browsing data between users. This allows the edge cache to be able to store the desired data in a targeted manner, greatly improving the hit rate. In contrast, the literature [3, 4] analyzes from user behavior. The users with the same browsing class are clustered. So that the edge caching mechanism can be processed. The literature [5] uses user portraits to predict the click likelihood of different types of ads. And good experimental results were achieved.

In this paper, we further optimize the edge caching mechanism based on machine learning. And for the computational power reduction and possible security problems in processing data caused by large amount of data buffer processing in traditional edge computing. We use federation learning [15, 16] to ensure the processing performance of the policy and the security privacy of the data.

3 Caching Strategy Design

3.1 Regional Prevalence Prediction

Before making an edge cache resource allocation, we need to consider how to rationalize the limited resources in the best possible way. But it is not difficult to find out. There is a strong and weak relationship chain between users and users in terms of different demand information. Then we can assume. The more the requirement information between different users fits together, the stronger the relationship property between them. Then the

more likely this group of information will be selected as cache data. And as the demand of a certain group of information increases, all user-related edge devices associated with it will form a regional network.

3.1.1 Content Popularity

In this paper, by examining the Movielens dataset we can assume that the more the number of views, the higher the popularity of that data. The literature [14] through user relationship perception can conclude that the number of visits in the dataset for video data information is in accordance with Ziff's law. Which Combined with the Salp Swarm algorithm [17] and Then it means that close to 70%−80% of the requested information topics originate from 20% of the content among all the users requesting to view. Then, we cite the literature [11, 14] for the calculation of regional prevalence. Let the total number of video data in the dataset be F. The popularity of the content ranked as j in the video dataset is

$$\overline{p_j^{pop}} = \frac{j^{-1}}{\sum_{c=1}^{|F|} c^{-l}}, 1 \leq j \leq |F| \tag{1}$$

where: l is the offset index that follows Ziff's law. The higher the offset index is, the higher we can conclude that the video is in demand index.

3.1.2 User Favorability

In general, the content that users choose to browse is not limited to popular content, but depends in large part on the users' own preferences for different content. Then, we can consider the user's preference for a certain type of data information as.

$$PR_{pop} = \frac{t_{total}}{w_{total}} \tag{2}$$

Among them. PR_{pop}: The goodness of a certain type of information.
t_{total}: The total time the user spent viewing the message.
w_{total}: Total user browsing time.

The user's favorability can, to some extent, determine the probability that the user will view such information. The higher the favorability, the higher the probability that the user will choose the information in the next random selection.

3.1.3 Prediction Model Selection

In the model selection part of data prediction, this paper decided to use XGBoost model for user requested content prediction after cross-validation of multiple literature queries [18–20]. And in order to fit the data, content popularity and user favorability will be added to the predictor as important weights for weighting. in order to get good prediction results. The validation results are shown in Fig. 2.

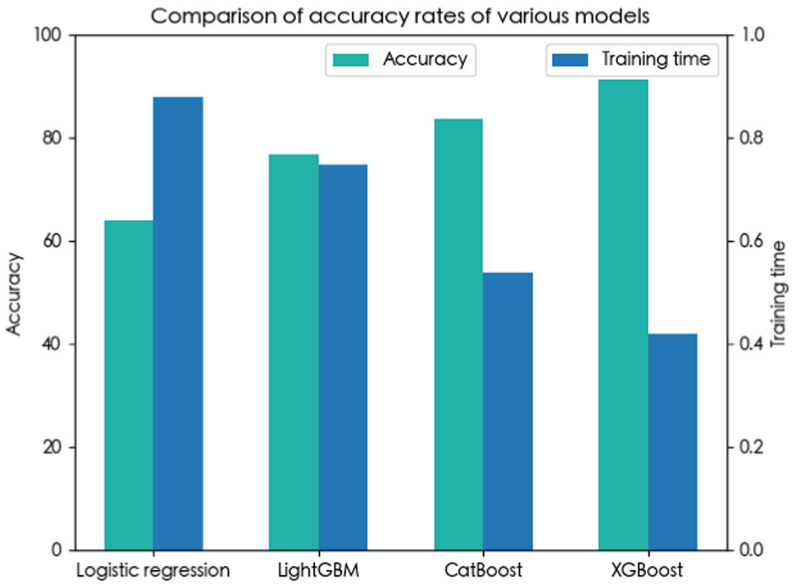

Fig. 2. Based on the single-model prediction results and training speed of Movielens dataset, it is not difficult to find that XGboost has the best actual results.

3.1.4 Regional Relationship Perception

According to studies [21, 22], there is a close relationship between the social relationships between users and their data needs on edge devices. The stronger the social relationship between users, the more similar their data needs are to each other, so we can establish their social relationship by calculating the similarity of needs between users. Then the users with strong social relationships are grouped into a regional network. The social relationships within the same region are stronger than those within different regions. Then the data requests within the same region can be considered to be roughly similar. Edge caching then allows for direct localization model training and cache allocation on a region-by-region basis. This reduces the pressure on caching and funding compared to one-to-one data training allocation. Increased efficiency of cache usage leads to improved architectural performance. And a significant reduction in resource overhead (Fig. 3).

where we can derive a request probability of user m for all types of data and integrate it into the vector formulation.

$$P_m^{re} = \left(P_{m,1}^{re} P_{m,2}^{re} \cdots P_{m,F}^{re}\right) \tag{3}$$

P_m^{re} can represent user m as a demand degree for all types of data. Then, we can map the social relationship between users and users by the similarity degree of demand degree between users. For example, the strength of the social relationship between users n1, n2

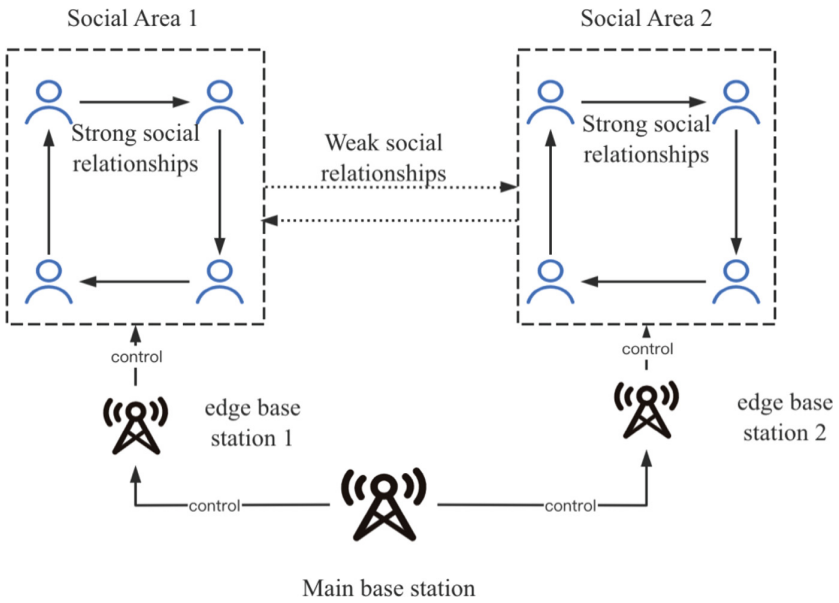

Fig. 3. Regional social relationship networks, where members in the same social area have strong social relationships with each other, while members in different areas have weak social relationships with each other, and each area has a sub-base station for independent control.

can be expressed as Eq. (4).

$$S_{n1,n2} = \frac{\left[P_{n1}^{re}\right] \bullet \left[P_{n2}^{re}\right]^{T}}{P_{n1}^{re} \times P_{n2}^{re}} = \frac{\sum_{j}^{|F|}\left(P_{n1,j}^{re} \times P_{n2,j}^{re}\right)}{\sqrt{\sum_{j}^{|F|}\left(P_{n1,j}^{re}\right)^{2}} \times \sqrt{\sum_{j}^{|F|}\left(P_{n2,j}^{re}\right)^{2}}} \tag{4}$$

Similarly, we can propose to impute all users based on the strength of the social relationship between two U_s (n1, n2 $\in$ U_s) between the social relationship strength relationship matrix expressed as Eq. (5).

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{U_s,1} & \cdots & a_{U_{s,n}} \end{pmatrix} \tag{5}$$

Then, the social intensity relationship of edge users in the same social area can be derived by combining Eqs. (4), (5) expressed as Eq. (6).

$$S_{area} = \frac{\left(\sum_{i=1}^{n} S_{ni,n(i-1)}\right) + S_{ni,n1}}{n} \tag{6}$$

Then, we can use Eqs. (4), (5), (6) to carry out the regional division of edge user devices, and divide the people with strong social relationship together. And the data popularity in the region is counted by Eq. (1) to get the data category with high popularity in the region.

Then the training parameters are pushed to the aggregation center after localization training by the machine learning model, and the model is distributed to regional base stations after aggregation by the aggregation center. The regional base stations use the models to make data predictions. The prediction results are pushed to the cache control center nearest to the requesting region to cache the most valuable cache content for that region, thus reducing edge requests and transmission latency.

3.2 Cache Optimization and Collaborative Model

Previous models of edge computing are based on a single base station [23] model network extended by the network. It provides services to all users within its coverage area. Suppose a user requests content data from a repository containing N different files, and for the purpose of edge caching, the cache space of the base station is assumed to cache up to n contents, where $n < N$. We assume that the user requests are generated sequentially one after another and the popularity of the requested content is not known. Suppose a user sends a request and the delivery flow after requesting content is shown in Fig. 4. When receiving a user's request, the base station first checks whether the requested content is in its cache. If it is, the base station delivers the content directly to the requesting user, a process called "cache hit". If the requested content is not in the cache, the base station requests the content from the data center first. Then it is forwarded to the requesting user, which is called a "cache miss" and takes much more time in the cache miss state than in the cache hit state.

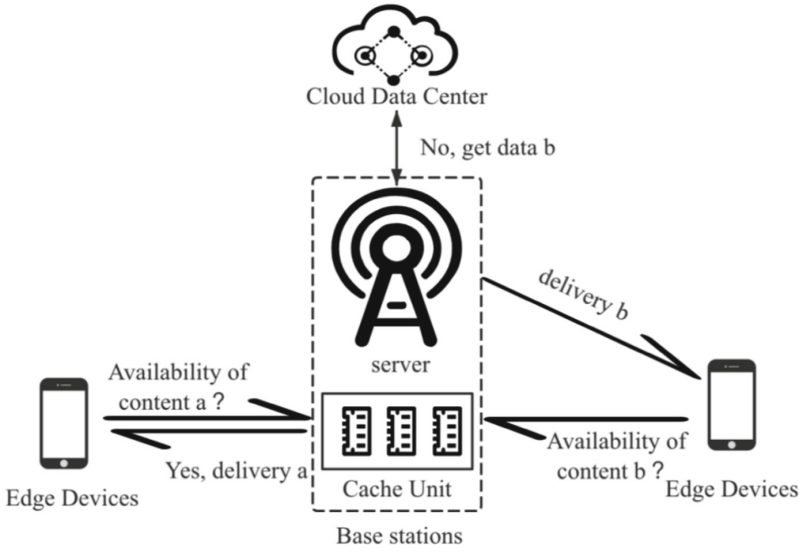

Fig. 4. Single base station based edge network request process, sending data directly to the terminal when the requested data is available in the cache, or fetching and sending from the data center if not

With the gradual normalization of the use of single base station mode, the problems brought about by it also come one after another. One is that the user needs to set up the local reference station by himself under the conventional situation of single base station mode. And there are errors when setting up the base station. It requires a lot of money and labor to set up and adjust. The most important problem is that the traditional network model does not have the ability of autonomous learning. Then it means that it cannot determine which resources are worth caching and which are not necessary. Therefore, conventional benchmarking strategies generally use random caching [24, 25]. When the cache space is already full, in each update, the total base station randomly selects the content in the cache to be replaced. And the cache hit rate of this caching method will generally be low. In most cases, there is no useful data stored in the cache, so the system can only go back to the cache one by one when the edge devices need data that is not in the cache and then distribute it to users. Both although with the advent of 5G era, ultra-dense networking, large-scale antenna technology, and millimeter wave technology [23] have alleviated the error latency problem to some extent. However, in terms of data updates, as each request and update must go through the cache control center, data center, and then return to the user. This request method will certainly bring about the problem of high latency, and this problem will become more obvious with the increase of data requests.

To address this pain, we design an autonomous edge cache request and update policy based on federated learning. Its update methods are divided into two types: passive and automatic updates.

Passive update: Assume that the cache has already stored the set of resources M predicted by the model. Edge users request resource d upward, at this time, the cache will automatically search whether there is resource d in M. If there is, the cache directly sends down the resource. If not, the cache center sends a request to the data center, which will obtain resource d to be dispatched to the corresponding edge device. At the same time, the aggregation center will send the most recently completed model of aggregation to the regional base station where the request is located for localization update training. After the training is completed, the trained update model is re-uploaded to the aggregation center for aggregation to complete the model update. The new model is also used to update the resources in the cache library to ensure high cache hit rate.

Automatic update: Considering the limited cache space of a single base station and the possible time-varying nature of content popularity, base stations deployed in each spatial region periodically update their cache contents to satisfy as many user requests locally as possible. Specifically, i.e., every N requests collected by the base station are recorded as a time series node t, where $t = \{1, 2 \ldots T\}$, and at each time node the base station starts to automatically update the prediction model with the cache content. Overall, the model update is divided into two processes: regional base station model update training and model reaggregation.The update of the cache is shown in Fig. 5: it consists of two actions: one is to select K new contents from the data center and put them into the cache, and the other is to remove K original contents from the cache space, where K indicates the number of contents to be replaced in each update. ($K \leq N$).

Another more direct reason for using Federated Learning is due to the fact that combining Federated Learning with spatial region technology allows for one-to-many request mapping. That is, regions are managed in a uniform manner. When there is a

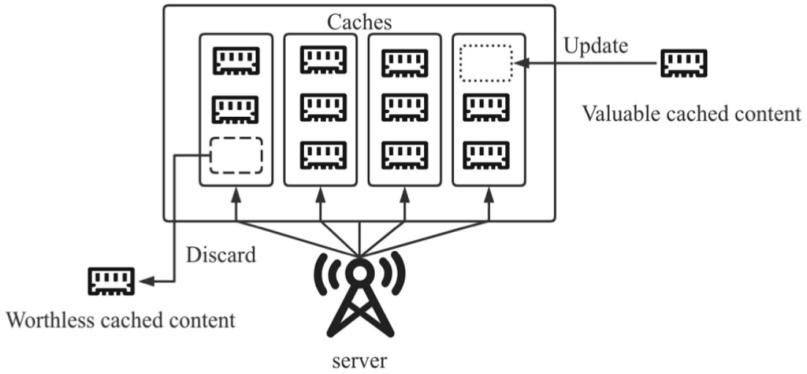

Fig. 5. The cache update process searches the cache for existing resources, and if there is data that has lost its value the cache controller will simply clear it and bring in valuable data from the data center to deposit it.

data update in a region, only the model needs to be trained for that region, and then the parent model and the corresponding cache center of that region can be updated. It avoids the traditional model of targetless.

The high latency and high resource consumption caused by cache updates. The specific control model is constructed as shown in Fig. 6.

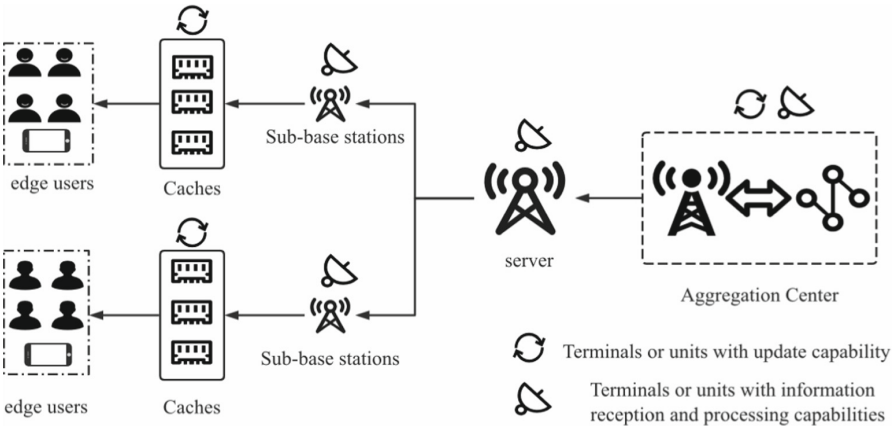

Fig. 6. Edge cache control process built based on federated learning, with the aggregation center distributing models to the corresponding sub-base stations in each region and indirectly controlling the corresponding cache controllers to store content

3.3 Co-caching Data Block Value Optimization

Edge co-caching [26] is an essential part of edge computing applications. Its as one of the prominent technologies to provide computing and communication capabilities

as well as network caching capabilities, where the edge server is located at the same location as the regional base station and the content is closer to the user, which not only relieves the pressure during network transmission, but also avoids network congestion caused by repeated transmission of the same content and achieves low latency response of data. For edge caching, the main issue is how to choose the right cache content and the allocation of cache resources. In this paper, the optimization of data blocks in the cache is quantified in terms of the popularity of the data blocks and the real-time value of the data blocks to establish the value model of the data blocks. The cache resources are updated in a timely manner to ensure high utilization of the cache.

3.3.1 Data Block Prevalence

Data block popularity has similarities with content popularity. Both are related to the number of visits, frequency of visits, etc. We can derive the access frequency f of a data block t in the cache based on the content popularity as Eq. (7).

$$f_t^{re} = \frac{W_t}{T_t^{latest} - T_t^{Initial}} \tag{7}$$

where W_t: denotes the sum of the number of times data block t was accessed between the current time and the time it was first accessed.

T_t^{latest} : Indicates the time of the last access to data block t

$T_t^{Initial}$ Indicates when the data block t was first accessed

From this, we can obtain the prevalence of data block tP_{pop}^t as Eq. (8):

$$P_{pop}^t = \frac{f_t^{re} * \frac{1}{f_{te}}}{\frac{1}{T_t^{ratest} - T_t^{Intitial}}} = \frac{f_t^{re} * \frac{1}{T_t^{latest} - T_t^{Initial}}}{\frac{1}{f_t^{re}}} \tag{8}$$

3.3.2 Residual Value of Data Blocks

The residual value of a data block represents the residual value of the data in that data block for the entire system, as well as for the architecture and the users in many ways.

Then it can be deduced that the proportion of residual value of data block tRe_t as Eq. (9).

$$Re_t = \frac{T_t^{\exp} - T_t^{now}}{T_t^{\exp} - T_t^{store}} \tag{9}$$

where $T_t^{\exp}$ denotes the expiration time of data block i.

T_t^{now} indicates the current time of the data block.

T_t^{store} Indicates the time when the data block is stored in the cache.

Therefore, the residual value of data block t can be deduced by combining Eqs. (8), (9) as Eq. (10).

$$Value = P_{pop}^t * Re_t \tag{10}$$

Then, based on the residual value of the data block, this paper adds an elimination mechanism for data block updates in the cache: if there exists an updated data block with a residual value greater than the residual value of the data block in the internal cache. Then the new data block is replaced with the data block with the lowest value in the cache. Otherwise, it is not replaced. This ensures that the data in the data cache is updated in real time. It also further improves the cache hit rate and achieves full utilization of resources. The data block update mechanism is shown in Fig. 7.

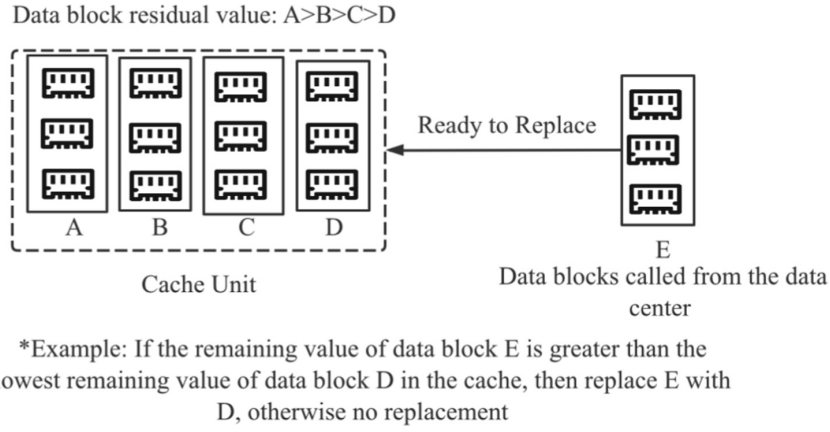

Fig. 7. Data block update elimination mechanism

4 Experiment and Analysis

To perform a validation analysis of the design strategy in this paper, we perform simulation experiments using pycharm in a python based environment. The dataset uses the publicly available dataset movielens to simulate the content of requests sent from edge endpoints to higher levels. The dataset has 18 hierarchies. Each sub-level contains more than 10,000 ratings from multiple users for various movies. The dataset is pre-processed with pycharm and datagrip to ensure that the results are not biased by human factors such as missing datasets.

Also, for statistical purposes, we consider each rating as a number of views. (i.e., each view is considered as a rating for each view). We will compare all aspects of this paper's strategy with the literature [14], literature [11], and literature [13] for effectiveness and draw final conclusions.

As can be seen from Fig. 8, the system cache capacity has a significant impact on the cache hit rate of all four policies in the experiments of this paper, and the cache hit rate of all three policies increases slowly as the system cache capacity increases. However, the federated learning and regional popularity-based edge collaborative caching strategy proposed in this chapter fully considers the data requests of different users in different regional environments, combines data popularity and regional users' own interests, reduces the redundancy of system cache contents, and improves the cache hit rate

compared with the other three compared schemes. On the one hand, increasing the total cache space of the system reduces the redundancy of the system cache content, and on the other hand, considering the variability of regional users further improves the overall cache hit ratio [27].

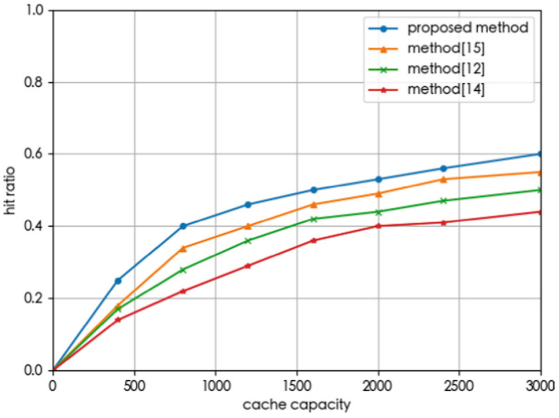

Fig. 8. Comparison of cache capacity-cache hit rate by policy

In addition, this paper also verifies the effect of cache size on the average download latency [28–31], and it is easy to see that the strategy with autonomous learning prediction clearly has a greater improvement than the traditional system strategy, while the strategy in this paper is more likely to get a lower latency in the comparison strategy. As shown in Fig. 9.

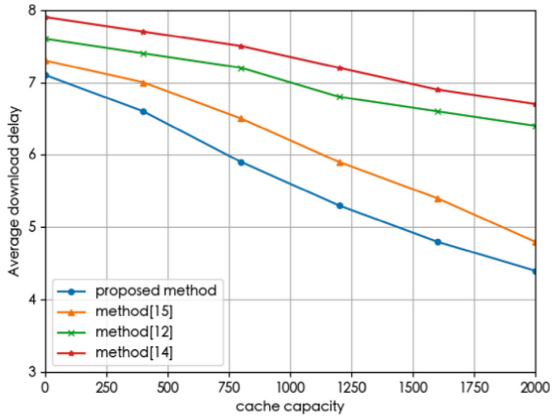

Fig. 9. Comparison of cache size - average download latency by policy

Also, considering the stability of the caching policy from the other side, we only consider the cache space of users. It can be concluded that the total number of users also has an impact on the cache hit rate. This is shown in Fig. 10.

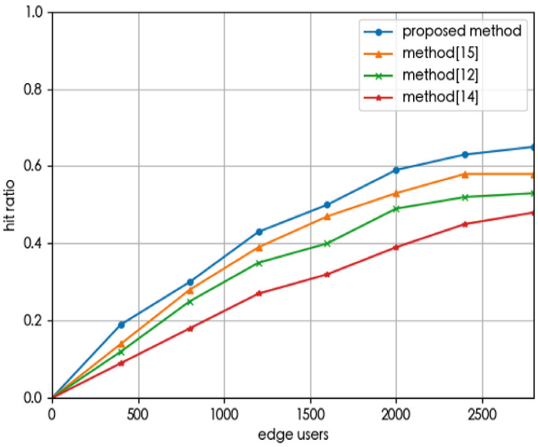

Fig. 10. Number of users at the edge of each policy - cache hit ratio comparison

Finally, from the economic efficiency consideration, this paper proposes the strategy with linear regression cache, literature [14], literature [11], literature [13] for the comparison of the number of edge users-energy savings as shown in Fig. 11.

(*Energy saving ratio [32] was derived from a comprehensive analysis of the number of devices, the time required for learning training of the devices, and the energy consumption for running the devices).

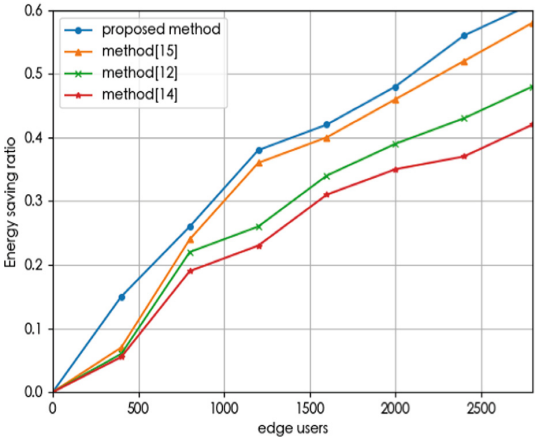

Fig. 11. Comparison of the number of users at the edge of each policy - energy savings ratio

这 From the above experiments, it is easy to see that the edge cache management policy with autonomous learning will be continuously updated with the increase of the user base and the prediction allocation accuracy will increase, which brings much higher positive benefits than the traditional allocation model. The policy proposed in this

paper has better experimental results compared with the traditional autonomous learning policy.

5 Conclusion

In today's Internet era. Both mobile terminals and user data volume are exploding. In such a big data context. Edge computing has become the obvious choice to solve this dilemma, and as it is widely used. The drawbacks also come one after another. Cache clogging, low security, etc. become the problems we need to solve urgently. In this paper, based on this background, the proposed innovation points are as follows.

Firstly. The popularity prediction model proposed in this paper is able to have a better cache hit rate than the comparative classical model, and also has a stronger learning classification capability. At the same time, the prediction model will also be updated according to the micro-migration of user preferences. The popularity updateability of the data is guaranteed.

Secondly. For the resource allocation problem of the cache module, we refer to and optimize the cache block utility model [6] based on the cached data blocks, determine the comprehensive benefits of the data blocks through model construction so as to perform reasonable allocation scheduling, and introduce an elimination mechanism to ensure the update of the data blocks. So as to maximize the utilization of cache and also reduce the unnecessary system overhead to some extent than the classical strategy.

Finally. Due to the previous machine learning caching strategy. All the data needs to be put into a cloud learner for centralized training. And this will significantly reduce the processing efficiency in the face of a large number of data processing [33]. And there are inevitable data privacy and security issues in the data transmission of highly dense spatial data. So we adopt the form of federation learning. Its core idea is that in the presence of multiple data sources jointly participating in model training, the model is jointly trained only by interacting model intermediate parameters without the need for raw data flow, and the raw data can be kept out of the local terminal. This approach not only reduces the amount of data computation in the cloud, but also achieves a balance between data privacy protection and data sharing and analysis, which is also called "transparent model". It greatly increases the security of data. Moreover, through federal learning + machine learning, the model can be updated in time to achieve the optimal predictive scheduling effect when deviations in data browsing hobbies occur in some regional edge devices [34, 35]. The overall policy performance is improved. In the future, we may also optimize the real-time interaction of some edge devices based on this architecture. For example, precise recognition of vehicles, pattern estimation, state detection [36–38], or the analysis and optimization of privacy and security during the interaction [39, 40]. This is still an important direction to be developed in the field of edge computing and IoT.

Acknowledgment. This work was supported in part by the Scientific research projects funded by the Department of education of Hunan Province (No. 22C0497), the Huaihua University Double First-Class initiative Applied Characteristic Discipline of Control Science and Engineering(No. ZNKZN2021-10), the National Natural Science Foundation of China (No. 62172182), the Hunan Provincial Natural Science Foundation of China (No. 2020JJ4490), the Project of Hunan Provincial

Social Science Foundation (No. 21JD046), the Huaihua University Project (No. HHUY2019-25), the Philosophy and Social Science Achievement Evaluation Committee of Huaihua (No. HSP2022YB40) and the Science and Technology Innovation 2030 Special Project Sub-Topics (No. 2018AAA0102100).

Hunan University Students' Innovation and Entrepreneurship Training Program (202210548064).

References

1. Zhao, M.: A review of edge computing technologies and applications. Comput. Sci. **47**(S1), 268–272+282 (2020)
2. Zhou, J.: A review of edge computing technology research at home and abroad. Comput. Age **08**, 8–11 (2021). https://doi.org/10.16644/j.cnki.cn33-1094/tp.2021.08.002
3. He, Z.Y., Dong, X.C., Zhu, Q.H.: Research on the classification of Baidu encyclopedia entries based on the perspective of users' usage behavior. Data Anal. Knowl. Discov. **3**(06), 117–122 (2019)
4. Zhang, L.-B., Guo, Q., Wu, X.-B., Liang, Y.-Z., Liu, J.-G.: Research on user clustering method based on multidimensional behavior analysis. J. Univ. Electron. Sci. Technol. **49**(02), 315–320 (2020)
5. Zhou, K., Wu, Y.C., Wu, J.K.: Research on the prediction model of Internet advertising click rate based on user portrait. Software **42**(02), 171–174 (2021)
6. Chen, N.N.: Research on integrated utility-based cache placement and task scheduling optimization methods in edge computing environment. Zhengzhou University of Light Industry (2022). https://doi.org/10.27469/d.cnki.gzzqc.2022.000027
7. Jiang, H., Dai, X., Xiao, Z., Iyengar, A.: Joint task offloading and resource allocation for energy-constrained mobile edge computing. IEEE Trans. Mob. Comput. (2022). https://doi.org/10.1109/TMC.2022.3150432
8. Zhou, J., Shen, H.J., Lin, C.Y., Cao, Z.F., Dong, X.R.L.: Advances in privacy-preserving research on edge computing. Comput. Res. Dev. **57**(10), 2027–2051 (2020)
9. Wang, Q.: Research on security and privacy protection technologies in edge computing. J. Jinling Inst. Sci. Technol. **36**(04), 11–17 (2020). https://doi.org/10.16515/j.cnki.32-1722/n.2020.04.003
10. Fei, L.: Research on computational offloading and resource allocation strategies in edge computing. University of Electronic Science and Technology (2022). https://doi.org/10.27005/d.cnki.gdzku.2022.003502
11. Liu, H.-Y., Wang, G., Yang, W.-C., Wang, J.-L., Xu, Y., Zhao, D.-L.: Popularity edge caching strategy based on random geometry theory. J. Electron. Inform. **43**(12), 3427–3433 (2021)
12. Wu, R.: Research on efficient edge caching strategy based on machine learning. Huazhong University of Science and Technology (2021). https://doi.org/10.27157/d.cnki.ghzku.2021.001857
13. Kai, J.: Research on computational offloading and content caching based on reinforcement learning in mobile edge computing. Three Gorges University (2021). https://doi.org/10.27270/d.cnki.gsxau.2021.000193
14. Liu, M.: Research on edge caching strategy based on spatio-temporal correlation analysis of user experience. Nanjing University of Posts and Telecommunications (2021). https://doi.org/10.27251/d.cnki.gnjdc.2021.000586
15. Yan, M., Lin, Y., Nie, Z.S., Cao, Y.F., Pi, H., Zhang, L.: A training method to improve the robustness of federated learning models. Comput. Sci. **49**(S1), 496–501 (2022)

16. Yin, C., Qu, R.: Federated learning algorithms based on personalized differential privacy. Comput. Appl. 1–9 (2022)
17. Ali, T.A.A., Xiao, Z., Sun, J., Mirjalili, S., Havyarimana, V., Jiang, H.: Optimal Design of IIR wideband digital differentiators and integrators using salp swarm algorithm. Knowl.-Based Syst. **182** (2019)
18. Wu, X., Li, J., Mao, W., Wu, Y.H., Zheng, L.Y.: Prediction of e-commerce users' purchase behavior based on GA-XGBoost. J. Zhejiang Wanli Coll. **35**(04), 86–92 (2022). https://doi.org/10.13777/j.cnki.issn1671-2250.2022.04.011
19. Xu, D., Xiao, Y.: Website user behavior prediction based on machine learning technology. Mod. Electron. Technol. **42**(04), 94–96+100 (2019). https://doi.org/10.16652/j.issn.1004-373x.2019.04.022
20. Zhang, S.: Research on user purchase behavior prediction based on machine learning. Chang'an University (2020). https://doi.org/10.26976/d.cnki.gchau.2020.000772
21. Xiao, L.: Analysis of user behavior in social networks. Small and medium-sized enterprise management and technology. Zhongjian J. (08), 115–116 (2019)
22. Zeng, F., Li, Q., Xiao, Z., Havyarimana, V., Bai, J.: A Price-based optimization strategy of power control and resource allocation in full-duplex heterogeneous macrocell-femtocell networks. IEEE Access **6**, 42004–42013 (2018)
23. Fu, J.: Research on angular time delay estimation and single base station localization algorithm based on 5G large-scale antenna. Beijing University of Posts and Telecommunications (2021). https://doi.org/10.26969/d.cnki.gbydu.2021.000248
24. Hu, Q., Wu, M., Guo, S., Peng, L.: A cache random placement policy for content-centric networks. J. Xi'an Univ. Electron. Sci. Technol. **41**(06), 131–136+187 (2014)
25. Lv, H., He, Y.X., Huang, C.H.: Randomized caching reliable multicast algorithm. J. Wuhan Univ. Technol. **31**(18), 24–27+75 (2009)
26. Dynasty, Gao, L., Gao Full Force: Collaborative caching strategy for data hierarchy in edge computing. J. Basic Sci. Text. Univ. **33**(03), 106–112 (2020). https://doi.org/10.13338/j.issn.1006-8341.2020.03.017
27. Zeng, F., et al.: Resource allocation and trajectory optimization for QoE provisioning in energy-efficient UAV-enabled wireless networks. IEEE Trans. Veh. Technol. **69**(7), 7634–7647 (2020)
28. Zhou, T.Q., Wu, W.J., Li, H.L., Dong, J.Y., Gao, J.J.: Analysis of uplink transmission performance and design of base station configuration for ultra-dense networks enhanced by mobile edge computing. High. Tech. Commun. **31**(09), 942–952 (2021)
29. Wu, Z.: Research on dynamic resource allocation delay optimization scheme for edge computing. Civil Aviation Flight Academy of China (2022). https://doi.org/10.27222/d.cnki,gzgmh.2022.000050
30. Hu, Z., Zeng, F., Xiao, Z., Fu, B., Jiang, H., Chen, H.: Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems. IEEE Trans. Netw. Sci. Eng. **8**(2), 1630–1645 (2021)
31. Liu, D., Cao, Z., Hou, M., Rong, H., Jiang, H.: Pushing the limits of transmission concurrency for low power wireless networks. ACM Trans. Sens. Networks **16**(4), 40:1–40:29 (2020)
32. Li, R..: Design and implementation of accurate sub-circuit metering of electricity consumption at base stations based on edge computing. Tianjin Normal University (2022). https://doi.org/10.27363/d.cnki.gtsfu.2022.000905
33. Qian, C., Liu, D., Jiang, H.: Harmonizing energy efficiency and QoE for brightness scaling-based mobile video streaming. In: IWQoS 2022, p. 1 (2022)
34. Liu, D., Cao, Z., He, Y., Ji, X., Hou, M., Jiang, H.: Exploiting concurrency for opportunistic forwarding in duty-cycled IoT networks. ACM Trans. Sens. Networks **15**(3), 31:1–31:33 (2019)

35. Liu, D., Hou, M., Cao, Z., He, Y., Ji, X., Zheng, X.: COF: exploiting concurrency for low power opportunistic forwarding. In: ICNP 2015, pp. 32–42 (2015)
36. Xiao, Z., et al.: Toward accurate vehicle state estimation under non-Gaussian noises. IEEE Internet Things J. **6**(6), 10652–10664 (2019)
37. Hu, J., et al.: BlinkRadar: non-intrusive driver eye-blink detection with UWB radar. In: Proceedings of IEEE ICDCS 2022 (2022)
38. Jiang, H., Xiao, Z., Li, Z., Xu, J., Zeng, F., Wang, D.: An energy-efficient framework for internet of things underlaying heterogeneous small cell networks. IEEE Trans. Mob. Comput. **21**(1), 31–43 (2022)
39. Liu, D., Wu, X., Cao, Z., Liu, M., Li, Y., Hou, M.: CD-MAC: a contention detectable MAC for low duty-cycled wireless sensor networks. In: SECON 2015, pp. 37–45 (2015)
40. Su, W., Liu, D., Zhang, T., Jiang, H.: Towards device independent eavesdropping on telephone conversations with built-in accelerometer. Proc. ACM Interact. Mob. Wearable Ubiquit. Technol. **5**(4), 177:1–177:29 (2021)