Journal of Multimedia Information System VOL. 10, NO. 2, June 2023 (pp. 137-144): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2023.10.2.137

Online Education Oriented Design and Estimation of Ideological and Political: An Edge Computing Approach

Ziqiao Wang[1], Zhefeng Yin[1*]

Abstract

With the rapid development of information technology, online education has become a new teaching mode in colleges and universities, and has made great contributions to China's education reform. The online education of ideological and political education for college students is still in its infancy. With the deepening and popularization of the mobile Internet, the use of mobile ends to carry out online ideological and political education courses is becoming a mainstream choice. However, the large number of click-to-play demands caused the problem of response delay. In order to solve this problem, this paper proposes an online ideological and political education architecture based on edge computing. In order to effectively utilize the edge computing architecture and optimize the transmission delay, this paper introduces a caching approach for the Genetic Algorithm utilized in Long and Short-term Memory Network, referred to as LSTM-GA. First, in the face of the fast refresh of video requests when user ends move between different edge computing servers, based on the "device-edge-cloud" system collaborative architecture, to reduce transmission delay and optimize QoE to build a network model for the target. Furthermore, the genetic algorithm optimized Long-Short-Term Memory Network (LSTM-GA) model is applied to forecast the trajectory of the user's endpoint, and seek the optimal solution with the minimum transmission delay, so as to ensure that the video content is cached in an appropriate location. According to the simulation results, the technique presented in this paper can significantly decrease transmission delays and enhance the quality of user experience.

Key Words: Ideological and Political Education, Online Education, Edge Computing, Caching Strategy.

I. INTRODUCTION

Currently, the enhancement of teaching quality and standards in colleges and universities has become a crucial topic across various industries due to the widespread access to higher education. As a novel pedagogical approach, online education has become one of the important teaching methods in college education. Ideological and political education is an important aspect of education that cannot be neglected. Under the guidance of China's new concept of building "curriculum ideological and political", the gradual formation of a large ideological and political system is inevitable for the development of education.

In traditional ideological and political education, the ideological and political classes offered by counselors cannot get a warm response from students due to strong theoretical and boring content. In particular, modern college students have a strong sense of autonomy and diverse personalities, coupled with the influence of modern Internet, so that they have a low degree of identification with the party, shallow family and country feelings, weakened collective consciousness, and insufficient cultural literacy. This is also the fundamental reason why contemporary colleges face challenges in instilling ideological and political values in students lies in the difficulty of capturing their hearts and minds, and it cannot guide their life, study, and social practice. In this context, online education is presented in a new form in the field of ideological and political education for college students, providing a space and direction for research and development of online ideological and political education in my country.

The essence of online ideological and political education is to help the majority of students establish correct ideological concepts, and then have a certain driving effect on their outlook on life and values, establish a good ideological and political quality and literacy for them, and obtain them through online activities. good interaction. The objective of online ideological and political education is to provide each student with an equitable chance to receive education, to transform the traditional teaching model into interactive teaching, to convert the identities between teachers and students, and to conduct equal exchanges, making ideological

Manuscript received March 24, 2023; Revised April 12, 2023; Accepted April 16, 2023. (ID No. JMIS-23M-03-012)

Corresponding Author (*): Zhefeng Yin, +86-13943391666, zfyin@ybu.edu.cn
[1]Department of Academic Affair Office, Yanbian University, Yanji, China, zqwang@ybu.edu.cn, zfyin@ybu.edu.cn

and political teaching more democratization. Online ideological and political education in colleges serves as a significant platform for ideological and political instruction. Proper integration of ideological and political education into teaching is crucial for reinforcing the value orientation of college students and enhancing their moral and ideological development.

Today's college students prefer using mobile devices to study ideological and political courses online. Learning through mobile platforms offers several advantages, such as environmental and resource efficiency, fragmented learning, interactive engagement, and enhanced cognitive benefits. With the rapid growth of mobile business data, the traditional centralized storage and computing model of cloud computing may lead to increased system response delays, network congestion, and other issues when responding to large numbers of on-demand or live broadcast requests for ideological and political courses from users. To address these drawbacks, the edge computing model has emerged. The edge computing model can migrate high real-time service and data requirements from the cloud computing center to the edge, where analysis and processing can take place close to the user end. This can effectively reduce the load pressure on the cloud center, decrease transmission delays, and improve the quality of user experience.

In order to meet people's demand for video-on-demand, it has become a research hotspot to use effective video caching technology in edge computing network architecture to place content on appropriate edge computing servers. Based on the characteristics of fast refresh of user end video requests, this study focuses on the video caching approach for ideological and political courses, centering on the development of a caching strategy that minimizes transmission delays and enhances user experience quality within a "device-edge-cloud" collaborative architecture. Furthermore, we propose a caching strategy based on Long and Short-term Memory Networks (LSTM-GA) that utilizes genetic algorithms for edge computing. Simulation results demonstrate that this caching approach can significantly reduce transmission delays and improve the overall quality of user experience.

II. RELATED WORK

With the gradual maturity of edge computing, the literature [1] considered the computing power of edge computing servers, so it is possible to process videos or perform other related calculations on edge computing servers. Over the past few years, deep learning has made remarkable advancements in diverse fields, including but not limited to image recognition, speech recognition, and natural language processing. It can achieve high prediction accuracy and light up the development path of continuous data processing, such as text and speech processing [2]. References [3-7] have investigated the prediction of video content popularity using deep learning techniques. Li et al. [4], utilizing data from Youku, a prominent online video service provider in China, addressed the challenges of understanding popularity trends and predicting the future popularity of individual videos. Meanwhile, Liu et al. [5] proposed a Software-Defined Networking (SDN) based approach called DLCPP (Deep-Learning-based Content Popularity Prediction) for predicting content popularity using deep learning techniques, demonstrating higher prediction accuracy through extensive experimental results. But none of the above studies apply it to the pre-caching of video content. Due to the lack of computing resources and training data on the mobile end, literature [8] designed a learning-based system structure. After centralizing the training data to the cloud, the computing resources in the cloud are used to train the deep learning model. The video content popularity score predicted by the model is pre-cached for the video content. However, uploading local data to the cloud will bring the risk of privacy data leakage. In addition, the amount of data that needs to be uploaded is huge, and it will also cause a lot of communication overhead.

The use of effective video caching technology in the edge computing network architecture to place the content on the appropriate edge computing server. It has garnered significant attention from both domestic and international scholars. In 2014, Ahlehagh & Dey [9] established a user preference analysis model based on the caching strategy implemented by User Preference Profile (UPP), analyzed the user preference in the target area and adjusted the cache content. The strategy has a good effect on users with stable motion state, and when the user state changes, the performance of the strategy decreases significantly. Sengupta et al. [10] integrated machine learning technology and caching technology to implement a caching strategy under a distributed architecture. Facing the cell heterogeneous network model, Gu et al. [11] proposed a content caching strategy (Q-Learning Content Caching Strategy, QLC) of Q-learning technology. When user preferences are in a steady state, QLC is able to predict and adjust caching strategies, but is ineffective for sudden changing user request events. Reference [12] realizes the cooperative cache strategy (Cache Strategy Based on Cooperation, CSC) based on the idea of mutual cooperation. This strategy takes the mutual communication between base stations as the research basis, and it demonstrates outstanding performance in both cache capacity and user experience quality, but does not filter the cached content, resulting in a large amount of duplicate content being cached in different base stations, resulting in space loss. Traverso et al. [13] proposed a caching strategy

(A Caching Strategy for Noise Model to Dynamically Obtain Content Popularity, NMDOCP). It is effective in suppressing the temporal locality of content popularity. But the noise model used does not perform well in terms of system propagation delay. Poularakis & Tassiulas [14] proposed a User-Aware Data Resource Caching Strategy (UADR), which provides an effective solution for edge caching strategies when users move. At the same time, the base station exists as a closed individual. When the user switches between base stations, the phenomenon of delay and jitter will appear. Li et al. [15] proposed a low-cost task scheduling caching strategy aimed at reducing the transmission cost.

In the research of edge caching strategy, although there are relatively mature caching solutions, most of them are generated for specific application scenarios and are relatively independent. The characteristics of fast refresh of video scenes of user ends have not been comprehensively reviewed. Analyze and consider. In order to solve the above problems, this paper takes the video caching strategy as the core, and conducts research on the video caching strategy for ideological and political courses for edge computing under the "device-edge-cloud" system collaborative architecture. Through an information technology perspective, this approach supports the advancement of online ideological and political education by improving the user experience of online courses and increasing their accessibility.

III. SYSTEM ARCHITECTURE AND NETWORK MODEL

3.1. System Architecture

The utilization of edge caching strategy in video scenes is becoming increasingly popular and is considered the general trend [16]. For edge computing-focused video caching strategies, a "End-Edge-Cloud" system collaborative architecture, as depicted in Fig. 1, is proposed based on the fast video refresh requirements and the user's on-demand video needs along with the scenarios of users frequently switching between various edge computing servers.

When a requested resource is cached on a nearby edge computing server, it will be directly responded to. If not found locally, adjacent edge computing servers will be searched. If the requested resource is found, the response will be coordinated through collaboration between the edge computing servers. If the resource is not found in the adjacent servers, the local edge computing server will search for it in the cloud center, which will respond with the search result. In the collaborative architecture of the system, different edge computing servers are controlled through a Software Defined Network (SDN) [17], and resources are configured uniformly. Within the "End-Edge-Cloud" collaborative architecture, the user end can request video resources as required. Implementing an appropriate caching strategy to distribute specific video resources across edge computing servers can significantly decrease data transmission redundancy and transmission delay, ultimately enhancing user experience quality.

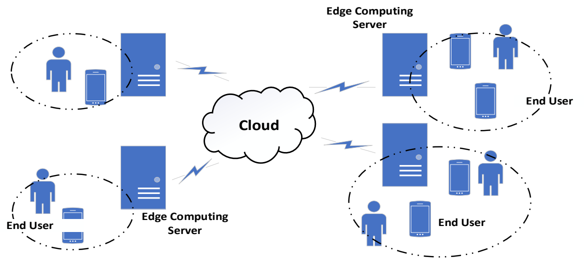

Fig. 1. "End-edge-cloud" system architecture.

3.2. Network Model

Considering that the network connection rarely switches in non-mobile scenarios, the connection is relatively stable, and the transmission delay and user experience quality are relatively good. This paper mainly considers the mobile scenario of the user terminal. Faced with the characteristics of fast video refresh, a network model is established with the goal of reducing transmission delay and improving user experience quality. The mobile scene here can be not only a smartphone, a tablet, etc., but also a laptop, a mobile workstation, etc. Let M be the set of M_{BS} edge computing cache servers at the edge, denoted as $\{1, 2, 3, ..., M_{BS}\}$, and U be the set of N users, represented as $\{U_1, U_2, U_3, ... , U_N\}$. The cache file matrix of the edge computing server is defined as X, where X_{if} equals 1 if the file f is cached in the edge computing service i, and 0 otherwise. The area covered by each edge computing server is considered a circle with a radius of r_i. The probability that user u requests file f is denoted as P_{uf}, and N_i represents the number of neighbors of the edge computing server. The user's movement speed in the service area is defined as v, and the time interval for the edge computing server to respond to the requested content is τ. The variables defined in this study are:

$$\begin{cases} Q = 0, \frac{r_i}{v} > \tau \\ Q = 1, \frac{r_i}{v} \leq \tau \end{cases}. \tag{1}$$

When Q=0, the user remains within the current service area during time τ. When Q=1, the user moves out of the current service area within time τ.

The hit rate is defined as assuming that the edge computing server moves to position j in the next time period:

$$\theta = \sum_{i\in M} p(Q=0) \cdot P_{uf} \cdot P(X_{if}=1) + \sum_{i\in M} p(Q=1) \cdot P_{uf} \cdot P(X_{if}=1). \tag{2}$$

In the spatial dimension, the path of the user end's movement can be regarded as a Markov chain model of several states. The moving direction and distance of the user end can be used to predict the current location information through the LSTM model through the historical parameters of the user end moving. Assuming that the number of videos cached at the edge computing end is m, which is expressed as F = {f_1, f_2, f_3, ... , f_m}, and the content popularity of fi is greater than f_j (i > j), and the size of video i is |f_i|. User requests for videos are independent and do not interfere with each other. According to the request probability P_{uf}, the assumption is made that the request probability follows the Zipf distribution, where the parameter σ is taken into consideration [18]:

$$P_f = \frac{f^{-\sigma}}{\sum_{k=1}^{m} k^{-\sigma}}, f = 1,2, \ldots, m, \tag{3}$$

where σ and m are the parameters of the Zipf distribution, respectively. According to the characteristics of fast video refresh, it is assumed that the user may request multiple video contents within a period of time, and the size of each video content is randomly distributed. Each video content has a delay requirement d_t, that is, the edge computing server must complete the response to the requested content within this time period, otherwise the request is deemed invalid.

The caching strategy's fundamental concept is to store a portion of the video collection to the edge computing cache server based on video collection attributes and user requests, generating a caching matrix **X**. The overall delay in the caching system is the total of the user's request delay in receiving a response from the edge computing cache server and the delay in receiving a response when requesting from the cloud center. To minimize the total delay, it is ideal to handle most of the requests on the edge computing cache server. If d_M represents the delay caused by the user requesting from the edge computing server, and d_Y represents the time it takes for data to travel from the cloud center. Thus, the total delay can be represented as:

$$D_T = \theta d_M + (1 - \theta) d_Y. \tag{4}$$

In the formula, θ is the hit rate. Constraints are: (1) The total size of each edge computing cache server's cache files cannot be greater than the size of the edge computing cache server's cache capacity; (2) Each video content file must have a transmission delay that is shorter than its required delay d_t. Additionally, the request delay should follow a Zipf distribution with a parameter of (σ, N). Thus, the problem of optimization strategy can be defined as follows:

$$\min\{D_T\}; \sum_{i\in M, f\in F} X_{if}|f| \le m_i \;;\; X_{if} \in \{0,1\}; d_t \sim Zipf(\sigma, N). \tag{5}$$

Under certain constraints, it is necessary to find an optimal solution that minimizes the time delay D_T of the system. Specifically, there may be multiple decision variables and multiple constraints, and it is necessary to optimize the time delay DT under these constraints, and find a set of optimal decision variables to minimize the time delay D_T. In this paper, the objection function is $\min\{D_T\}$, while the constraints are: $\sum_{i\in M, f\in F} X_{if}|f| \le m_i$, $X_{if} \in \{0,1\}$ and $d_t \sim Zipf(\sigma, N)$.

IV. LSTM-GA CACHING STRATEGY

The Long Short-Term Memory Network (LSTM) [19] is a temporal recurrent network that was proposed to address long-term dependency issues. As the user mobility problem in this study can be modeled as a Markov process, LSTM is well-suited for time series prediction tasks. With its unique structure of input, output, and forget gates, the LSTM network model can accurately predict the target edge computing server based on the user's historical trajectory information. Therefore, selecting the LSTM network model for this caching strategy model can lead to precise target edge computing server predictions.

The Genetic Algorithm is a powerful global optimization method that seeks to find the best possible solution. It mimics the principles of biological evolution, where new chromosomes are generated through natural crossover and mutation. The implementation of the genetic algorithm usually involves four steps: encoding, population initialization, fitness value calculation, and genetic operation. Due to its ability to enhance and optimize other algorithms, genetic algorithms have been extensively applied in various optimization problems.

There are three main advantages of genetic algorithms. Firstly, they possess excellent global search capabilities, which can prevent getting stuck in local optimal solutions and enable finding the global optimal solution. Secondly, they are highly compatible and can be combined with other algorithms for optimization. Thirdly, compared with general optimization problems, genetic algorithms require lower mathematical requirements, and do not need to establish conditions such as objective functions. They can be solved based on the given problem. Drawing on the advantages of the above three aspects, this paper employs the genetic algorithm to optimize the neural network, to find the global optimal solution, and conduct modeling.

This paper proposes the Genetic Algorithm-based Cache Strategy for Long and Short-Term Memory Network (LSTM-GA). The LSTM neural network model is integrated with the genetic algorithm to optimize the model parameters and establish the model. The aim of this caching strategy is to reduce the transmission delay to the greatest extent possible while maximizing content requests at the edge computing cache server when the user is in motion.

4.1. Prediction of Target Edge Computing Servers

In real-world scenarios [20], the video content stored in the edge computing cache server becomes obsolete when the user end moves between different servers. If the user switches the edge computing server while requesting a video file, the request gets interrupted. To address this issue, this paper proposes using the LSTM network model to predict the target edge computing server that the user may switch to in the next time period. By caching the requested or currently watched video file in advance in the predicted server, seamless switching of edge computing server cache resources can be achieved, resulting in smooth video playback.

The LSTM network model consists of several components, including an input gate, an output gate, a forgetting gate, and a memory unit. The forget gate is used to selectively forget some previous secondary state information. After forgetting some secondary state information, new memory needs to be added from the current state, which is realized by the input gate. The LSTM model needs to output the current state after calculating the new state, which is realized by the output gate at this time.

In the LSTM network model, the cell state used to control information transmission is $C_{t;}$ the previous sequence state is h_{t-1} and the current sequence input data is X_t. The forget gate's output, denoted as f_t, is computed using an activation function and has a range of [0,1]. The value of f_t corresponds to the probability of discarding the information stored in the previous hidden state. Its expression is:

$$f_t = \sigma\left(W_f[f_{t-1}, X_t]\right) + b_f. \tag{6}$$

In the formula, W_f and b_f represent the coefficient and bias of the linear relationship, respectively. σ is the activation function. The input gate is responsible for implementing the input of the current sequence. The output consists of two parts: the first part outputs i_t, which is implemented using the σ activation function; The second part outputs a_t, which is implemented using the *tanh* activation function.

The prediction of the target edge computing server that the user end will switch to in the next time period involves several steps. Firstly, the edge computing server set **H** and the time slice τ based on the user end's historical movement trajectory are inputted. Next, the LSTM network model is utilized to predict the edge computing server that the user end is likely to switch to in the next time period. Then, o_t is generated, and the edge computing server h_t to which the user end will move is determined based on o_t and C_t. Subsequently, the total delay D_t of the requested video transmission at the edge computing server h_t is calculated to obtain the local optimal solution, and the local buffer matrix **X** is updated.

4.2. Model Establishment

The LSTM is first optimized using the GA algorithm. By training the LSTM model and evaluating the appropriateness, the optimal parameters are selected. Then use the optimized LSTM network model to predict the user end moving path. Determine the target edge computing server with the smallest transmission delay and cache the video to achieve the optimal solution. Using the GA algorithm to find the optimal window and number of units for LSTM is roughly divided into the following four steps:

(1) Decode the genetic algorithm solution to obtain the window size and number of cells. (2) Use the window size found by GA to prepare the dataset and divide it into training and validation sets. (3) Input the LSTM model, calculate the RMSE on the validation set, and return the value as the fitness value of the current genetic algorithm solution to obtain the optimal parameters. (4) Establish an LSTM neural network model, and use the optimal time window size and the optimal number of hidden layer units of the neural network obtained in the above steps as corresponding parameters to predict the result.

Chromosomes used in this paper are encoded as binary bits, representing the size of the time window and the number of LSTM neural units. The initial population is randomly generated, initialized randomly using the Bernoulli distribution. Evaluation based on fitness function and selection, followed by crossover and mutation, using ordered crossover, random mutation, and wheel selection. This process is repeated for a defined number of iterations. Finally, a solution with the highest fitness score is selected as the best solution. The fitness function is an important part of the genetic algorithm and must be chosen carefully. In this paper we use RMSE (root mean square error) to calculate the fitness of each chromosome, and return the smallest RMSE subset as the optimal solution to obtain the optimal window width and number of cells.

V. SIMULATION EXPERIMENTS

The performance of the LSTM-GA caching strategy under the "end-edge-cloud" system collaborative architecture is evaluated by comparing it with other caching strategies such as UPP [9], QLC [11], and CSC [12]. Various performance indicators such as hit rate, transmission delay, throughput, and packet loss rate are compared and analyzed to assess the effectiveness of the LSTM-GA caching strategy.

5.1. Simulation Environment

In the simulation experiment, the input video data mainly includes video ID, video name, video size and popularity. User data mainly includes user ID, location information, number of connected edge computing servers, distance moved within a time slice, user preference set, total popularity, popularity fluctuation parameters, and target edge computing server location. The primary information stored on the edge computing server comprises the quantity of base stations, the range of each base station, the number of users connected to each base station, and the cache queue.

The genetic algorithm evaluates individuals in each iteration using a fitness function. Individuals with higher fitness scores are considered better solutions, and they are more likely to be selected for reproduction. The traits of these individuals are then manifested in the next generation, improving the overall quality of the solution. As the genetic algorithm progresses, the quality of the solution will improve and the fitness will increase. This paper mainly evaluates the influence of the number of iterations of the GA algorithm on the experimental results. The experimental results are shown below. The hit ratio of resource requests is evaluated first, followed by the throughput of the system. The experimental results are shown below. The hit ratio of resource requests is evaluated first, followed by the throughput of the system.

5.2. Result Analysis

The hit rate measures the proportion of successful responses to video resource requests at the edge computing cache server (or neighboring servers) compared to the total number of requests. Fig. 2 presents the hit rate results for the caching strategy proposed in this paper and three other strategies (UPP [9], QLC [11], and CSC [12]) under different iteration times.

Based on the results depicted in Fig. 2, the hit rate of the proposed LSTM-GA strategy is initially slightly inferior to that of the other strategies. This is mainly due to the lack of adequate historical trajectory information of the end user at the start of the iteration, which causes a deviation in the target MEC server that the user moves. Nonetheless, with an increase in the number of iterations, the system's collection of user terminal trajectory data improves, leading to more accurate prediction results and an increase in the hit rate. Overall, it can be observed that the proposed strategy outperforms the other three strategies in terms of hit rate as the number of iterations increases.

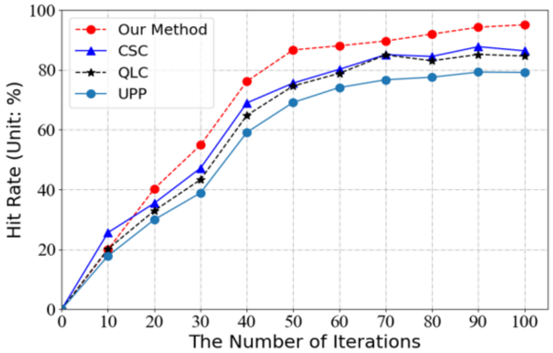

Fig. 2. Experimental results on hit rate over the number of iterations.

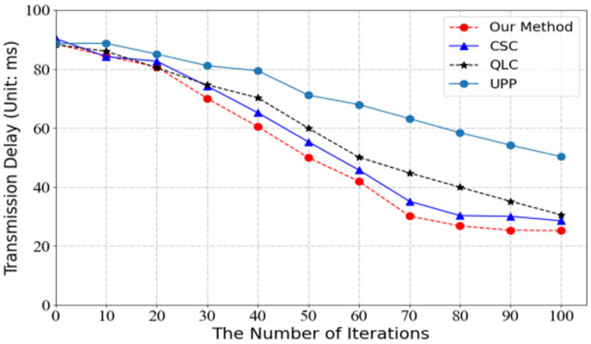

Fig. 3. Experimental results on transmission delay over the number of iterations.

The primary meaning of transmission delay in this context is the overall time taken from when a request is sent to when the user end receives a response. The smaller the transmission delay, the better the system cache policy performance and the higher the user experience quality. Fig. 3 shows the comparison results of the transmission delay performance between the proposed strategy and the above three strategies under different iteration times. The simulation results indicate that the LSTM-SA approach suggested in this paper exhibits superior performance concerning the total transmission delay, particularly as the number of iterations increases.

VI. CONCLUSIONS

Ideological and political education conducted online removes the constraints of time and location on ideological and political learning, providing a new approach for this form of education. This is particularly meaningful for college students' ideological and political education. Focusing on video caching strategy as its core, this study investigates the caching strategy for ideological and political courses in edge computing under the "device-edge-cloud" collaborative architecture. Accordingly, a caching strategy based on the long-short-term memory network and genetic algorithm is proposed in this paper, which reduces transmission delay and optimizes user experience quality by predicting the best caching location for video content. Simulation experiments show that the proposed LSTM-GA caching strategy has higher hit rate and throughput than other caching strategies, and effectively improves the quality of user experience. Undeniably, this paper also exists two aspects of limitations. On one hand, this paper lacks of the large-scale data collection on ideological and political opinions, which causes that the evolutionary intelligent method based on GA has no better stability. On the other hand, the task scheduling in the edge computing has not been considered, which reduces the efficiency of edge computing. In future, the mentioned limitations will be addressed.

ACKNOWLEGMENTS

This work was supported by the General Subject of Higher Education and Scientific Research of Jilin Province in 2022 (Granted No. JGJX2022D44) and the Higher Education Reform Project of State Ethnic Affairs Commission in 2011.

REFERENCES

[1] M. Chen and V. C. Leung, "From cloud-based communications to cognition-based communications: A computing perspective," *Computer Communications*, vol. 128, pp. 74-79, 2018.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[3] R. Devooght and H. Bersini, "Collaborative filtering with recurrent neural networks," *arXiv preprint arXiv: 1608.07400*, 2016.

[4] C. Li, J. Liu, and S. Ouyang, "Characterizing and predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 1630-1641, 2016.

[5] W. X. Liu, J. Zhang, Z. W. Liang, L. X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075-5089, 2017.

[6] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3520-3535, 2017.

[7] K. C. Tsai, L. Wang, and Z. Han, "Mobile social media networks caching with convolutional neural network," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW),* Apr. 2018, pp. 83-88.

[8] K. Thar, N. H. Tran, T. Z. Oo, and C. S. Hong, "DeepMEC: Mobile edge caching using deep learning," *IEEE Access*, vol. 6, pp. 78260-78275, 2018.

[9] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking,* vol. 22, no. 5, pp. 1444-1462, 2014.

[10] A. Sengupta, S. Amuru, R. Tandon, R. M. Buehrer, and T. C. Clancy, "Learning distributed caching strategies in small cell networks," in *Proceedings of the 2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, Aug. 2014, pp. 917-921.

[11] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *Proceedings of the 2014 IEEE International Conference on Communications (ICC),* Jun. 2014, pp. 2648-2653.

[12] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1382-1393, 2016.

[13] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today’s content caching: Why it matters and how to model it," *ACM SIGCOMM Computer Communication Review*, vol. 43 no. 5, pp. 5-12, 2013.

[14] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyperdense small-cell networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 675-687, 2016.

[15] C. Li, J. Bai, and J. Tang, "Joint optimization of data placement and scheduling for improving user experience in edge computing," *Journal of Parallel and Distributed Computing*, vol. 125, pp. 93-105, 2019.

[16] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680-698, 2018.

[17] Z. Jianmin, F. Yang, Z. Wu, Z. Zhengkun, and W. Yuwei, "Multi-access edge computing (MEC) and key technologies," *Telecommunications Science*, vol. 35, no. 3, pp. 160-160, 2019.

[18] C. Li and J. Liu, "Large-scale characterization of comprehensive online video service in mobile network," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC),* May, 2016, pp. 1-7.

[19] K. Kawakami, "Supervised sequence labelling with recurrent neural networks" Ph.D. dissertation, Technical University of Munich, 2008.

[20] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proceedings of the 2017 13th annual Conference on Wireless on-Demand Network Systems and Services (WONS)*, Feb. 2017, pp. 165-172.

AUTHORS

Ziqiao Wang received his Bachelor Degree and Master Degree at Yanbian University in 2013 and 2016 respectively. He is currently an Experimental Engineer at the Educational Technology Center of Yanbian University. His research interests include English translation and computer technology..

Zhefeng Yin received his Bachelor Degree and Master Degree at Yanbian University in 2002 and 2009 respectively. He is currently an Associate Researcher at the Academic Affairs Office of Yanbian University. His research interests include English translation and computer technology.