Journal of Multimedia Information System VOL. 11, NO. 1, March 2024 (pp. 67-82): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2024.11.1.67

Enhancing Higher Vocational English Teaching through Edge Computing: A Framework for Real-Time Language Learning

Yinan Song[1*]

Abstract

In higher vocational English teaching, delivering teaching materials promptly and enabling instantaneous feedback is crucial for effective language learning. This paper presents a novel algorithm applying Adversarial Autoencoders (AAE) to reduce latency in 5G networks. The proposed algorithm integrates the characteristics of small base station collaboration, multicast, and predictable user behavior, namely the AAE-based collaborative multicast proactive caching scheme (AAE-CMPC). Initially, students are grouped into different preference clusters based on their characteristics. Subsequently, the AAE technique is employed to predict the content that each group might request. To reduce the redundancy of cache content, an ant colony algorithm is used to pre-deploy the predicted content to each small base station to realize the collaboration between small base stations. The proposed AAE-CMPC scheme demonstrates superior performance when compared to three benchmarks. The simulation results indicate that an increase in the storage capacity of the macro base station leads to a reduction in loss rate, and it can be attributed to the enhanced cache hit ratios achieved through proactive caching. The AAE-CMPC algorithm revolutionizes higher vocational English teaching by reducing latency and enabling instantaneous feedback. Students can access teaching materials promptly, receive real-time feedback on their progress, and engage in collaborative activities seamlessly. The framework also leverages edge computing, allowing for increased storage capacity, scalability, and reliability, resulting in an enriched learning experience.

Key Words: Higher Vocational English Teaching, Edge Computing, Adversarial Autoencoders, Collaborative Multicast Proactive Caching.

I. INTRODUCTION

English language proficiency has become increasingly important in today's globalized world, where cross-cultural communication and international collaboration are vital for personal and professional success. Particularly, students in higher vocational education require strong English language skills to thrive in their chosen fields, which often involve interactions with international clients, colleagues, and partners [1]. In many industries, proficiency in English is a prerequisite for job opportunities and career advancement. Whether in business, technology, healthcare, hospitality, or any other field, communicating effectively in English opens doors to a broader range of opportunities [2]. It enables individuals to engage in global networks. Employers seek professionals who can confidently interact with diverse stakeholders and navigate international markets, making English language proficiency a key asset in today's job market.

Although traditional English teaching methods are valuable in providing a foundational understanding of the language, they often need help to fully prepare students for real-world communication in higher vocational education. These methods typically rely on standardized curricula, traditional classroom settings, and one-size-fits-all instructional materials, which can limit their effectiveness in meeting the specific needs and contexts of students pursuing vocational careers [3-4].

The demands of real-world communication go beyond memorizing grammar rules and vocabulary lists; they involve effective oral and written communication, presentation skills, negotiation abilities, and intercultural competence. Moreover, traditional classroom settings may only partially reflect the dynamic and diverse environments that students will encounter in their vocational careers. Classroom interactions often involve limited opportunities for authentic language use and need more exposure to industry-specific terminology, cultural nuances, and communication styles. Students need exposure to real-world scenarios, interactive role-plays, and authentic materials that mirror the challenges they will face in their professional lives. Each vocational field has unique language requirements, and students may require specialized instruction tailored to their chosen career paths [5]. A standardized curriculum may

Manuscript received December 21, 2023; Revised March 02, 2024; Accepted March 10, 2024. (ID JMIS-23M-12-051)

Corresponding Author (*): Yinan Song, +86-18569999101, echosongyinan@163.com
[1]Department of Liberal Arts, Henan Women's Vocational College, Zhengzhou, China, echosongyinan@163.com

need to sufficiently address these specific language needs and contexts, leading to a gap between classroom learning and real-world application.

In English learning, real-time feedback is critical in helping students improve their language skills. Traditional teaching methods often provide feedback after a delay, hindering students' ability to correct mistakes and reinforce their learning immediately [6]. In today's digital age, the proliferation of connected devices and the exponential growth of data has led to new challenges and opportunities in computing. Edge computing has emerged as a promising paradigm that aims to address these challenges by bringing computation and data storage closer to the network's edge, near the source of data generation [7-8]. With edge computing, students can receive real-time feedback on pronunciation, grammar, vocabulary usage, and sentence structure. Reducing latency achieved through edge computing is particularly crucial for English learning. It enables students to correct mistakes promptly, reinforcing proper language usage and preventing the development of incorrect language habits. Students can receive instant feedback on their language production, facilitating a more efficient learning process. Additionally, edge computing allows for personalized and adaptive learning experiences. By leveraging data processing and machine learning algorithms on edge devices, the system can adapt to individual students' needs and provide customized feedback and recommendations. Students' performance and progress can be analyzed locally on edge devices, ensuring the privacy and security of personal data.

The reduced latency achieved through edge computing, mainly through edge caching, is significant for real-time English learning. Edge caching involves storing frequently accessed data closer to the edge devices, allowing quicker access and reducing the need to fetch data from distant servers [9]. Traditional approaches often rely on centralized servers or cloud-based solutions, which can introduce latency due to network congestion or longer data retrieval times. This latency can negatively impact the learning experience, as delays in accessing materials and receiving feedback hinder students' ability to address their language learning need promptly. By implementing edge caching, the framework for real-time English learning can leverage the proximity of edge devices to provide instant access to learning materials [10]. Frequently used resources, such as multimedia content, interactive exercises, language reference materials, and instructional videos, can be cached at the edge, reducing the time it takes for students to access them. It enhances the efficiency and responsiveness of the learning process, allowing students to engage with the materials without noticeable delays. Moreover, edge caching can improve the real-time feedback mechanism in English learning. Students' performance data can be processed and analyzed locally on edge devices as they interact with language learning applications or platforms. This analysis includes assessing pronunciation, grammar usage, vocabulary proficiency, and comprehension. The feedback generated based on this analysis can be cached and delivered in real-time, providing students with immediate insights into their strengths and areas for improvement. The integration of edge caching within a framework for real-time English learning significantly enhances the learning experience by reducing latency, enabling quick access to learning resources, and delivering immediate feedback [11]. By leveraging the proximity of edge devices, this caching mechanism contributes to building a responsive and efficient learning environment that supports students in their language acquisition journey.

Early research on integrating edge computing in contexts like augmented reality, smart classrooms, and adaptive assessments shows initial promise. However, comprehensive computational frameworks to unlock the potential of edge computing for transforming teaching, particularly in key areas like instantaneous feedback, remain open challenges. The motivation is to spark a broader discourse on edge-based pedagogical innovations through an exemplar algorithmic instantiation. Considering those mentioned above, the main contribution of this paper is that the adversarial autoencoders-based collaborative multicast proactive edge caching scheme is proposed to reduce latency and loss rate in higher vocational English teaching.

The remainder of the paper is organized as follows. Section 2 gives the related works. Section 3 proposes the adversarial autoencoders-based collaborative multicast proactive edge caching scheme. The simulation and results analysis is provided in Section 4. Lastly, Section 5 presents the conclusions.

II. RELATED WORKS

2.1. Application of Edge Computing in English Teaching

Edge computing is optimizing cloud computing systems by processing data at the network's edge near the data source, thereby reducing latency, bandwidth usage, and the amount of data sent to the cloud. In [12], the authors developed an autoencoder model using edge enhancement to tackle these issues and uncover the hidden communities in complex networks. In [13], the authors investigated a novel service architecture of traffic sensing based on mobile edge computing where collected data was pre-processed at the edge node and reconstructed at cloud servers, respectively. Notably, edge computing is helpful for real-time applications, such as video streaming, gaming, and language learning, where delay or lag can negatively impact user experience [14]. Edge computing can be used to create smart classrooms that can monitor student engagement, track progress, and provide personalized learning experiences [15-16]. For example, speech recognition can assess pronunciation and provide real-time feedback, while facial recognition can monitor student engagement and focus. Virtual and augmented reality applications can significantly benefit from edge computing by reducing latency and providing smoother, more immersive experiences. These technologies can create immersive language learning environments that simulate real-life situations, making it easier for students to practice speaking and listening skills. Simultaneously, edge computing can be integrated into language learning apps to provide real-time feedback, personalized content, and offline functionality, which can help students practice their skills on-the-go and receive immediate feedback without relying on a constant internet connection [17]. Additionally, edge computing can create adaptive assessments that adjust in real-time based on a student's performance. It can help teachers identify areas where students need more support and tailor their teaching accordingly [18-21].

In conclusion, edge computing has the potential to revolutionize English teaching by providing faster response times, personalized learning experiences, increased security, reduced bandwidth and energy consumption, and offline functionality. By integrating edge computing into language learning applications and environments, educators can create more engaging and effective learning experiences for their students.

2.2. Edge Caching

Edge caching is a mechanism used in edge computing that involves storing frequently accessed data closer to the edge devices, reducing the need to fetch data from distant servers. By caching popular content or resources at the network's edge, edge caching improves the performance and responsiveness of applications and services [22]. Edge caching aims to minimize the latency and network congestion that can occur when data needs to be retrieved from remote servers or the cloud. Instead of accessing centralized storage, edge devices can quickly retrieve cached data from nearby edge servers, resulting in faster response times and reduced delays [23-24]. Edge caching is particularly beneficial in scenarios where real-time access to data is crucial, such as in real-time communication applications, streaming services, or content delivery networks. By bringing the data closer to the end users, edge caching reduces the time it takes to access and deliver content, improving the user experience and reducing network traffic. Furthermore, edge caching improves scalability and bandwidth utilization by offloading the centralized servers and distributing the computational load [25]. It allows for efficient content distribution, as popular or frequently requested data can be cached at multiple edge locations, reducing the strain on the network and optimizing data transmission. Edge caching is vital in optimizing performance, reducing latency, and improving the efficiency of edge computing systems. Using the proximity of edge devices and caching frequently accessed data enhances the responsiveness and reliability of applications, ultimately providing a better user experience. In [26], the authors proposed an alternating iterative algorithm-based efficient algorithm called task caching and offloading (TCO). In [27], the authors proposed a cache deployment strategy, i.e., large-scale WiFi edge cache deployment (LeaD). To solve the long-term caching gain maximization problem, they first group large-scale access points into appropriately sized edge nodes, test edge level traffic consumption stationary, sample enough traffic statistics to accurately characterize long-term traffic conditions, and then develop the traffic-weighted greedy algorithm. The authors of [28] suggested a system incorporating blockchain, edge nodes, remote cloud, and Internet of Things devices. They created a novel algorithm for the CREAT system that used blockchain assistance to compress federated learning for content caching.

III. METHODOLOGY

3.1. System Model

This paper considers the scenario of cooperative caching of multiple small base stations S under a macro base station M. As shown in Fig. 1, it contains macro base station M, small base stations S and students U. A set $S = \{s_1, s_2, s_3, \cdots, s_K\}$ is included in the range covered by the macro base stations, where K represents the total quantity of small base stations. The number of users presents within the coverage radius r of the small base station s_K is denoted by N_s_K. A Poisson distribution with parameter λ models this quantity.

$$\mathrm{P}(N_{s_K}) = \frac{(\lambda\pi r^2)^{N\_s_K}}{N\_s_K!} e^{-\lambda\pi r^2}, \tag{1}$$

where λ denotes the mean student count per unit area. The interconnection between the small and macro base stations is established via optical fiber. The macro base station is endowed with the complete information of the small base station and is responsible for the regulation and management of the small base station.

The content requested by student u_m belongs to the set $C = \{c_1, c_2, c_3, \cdots, c_N\}$, and N represents the contents' total number. Considering that both terminal and student equipment have specific cache capacity, V=$\{V_c$, V_m, C_s, $V_u\}$is defined, which represent the storage capacity of the cloud, macro base station, small base station, and student, respectively. Since the distance of student u_m obtaining

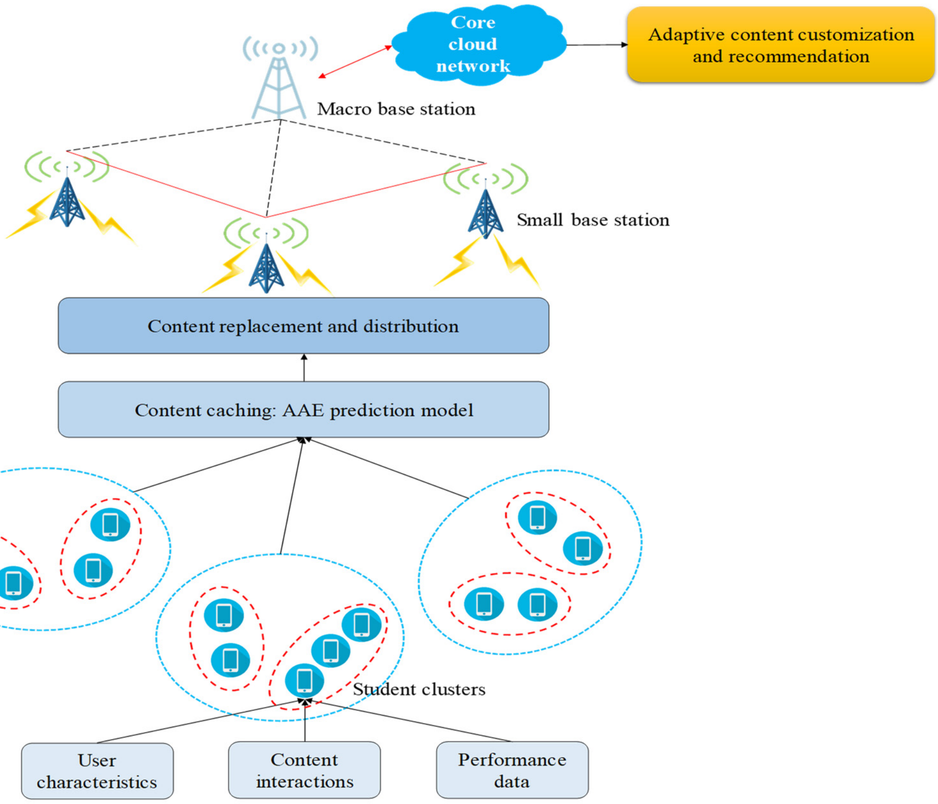

Fig. 1. Overall framework.

content from different places is different, the transmission latency in the system is assumed to be t , $t = \{t_l, t_s, t_m, t_c, t_{ss}\}$, which respectively represents the transmission latency between student and local, small base station, macro base station, cloud and the transmission latency between adjacent small base stations. For the request latency of the student, only the transmission latency of the content is considered, so the latency of the student u_m to get the content from locally is 0. However, the small base station is closer to the student. It is closer to the student than the cloud macro base station, so the transmission latency relationship satisfies $t_l = 0 < t_s < t_m < t_c$. Since the transmission latency between adjacent small base stations is t_{ss}, the transmission latency between small base stations s_{k_1} and s_{k_2} is defined as ht_{ss}, where h is the number of hops traversed between small base station s_{k_1} and s_{k_1}. Therefore, the present study defines the system's average latency T as the mean value of the request latency of the students under each small base station. The request latency is defined as follows.

$$T = \frac{1}{N\_u}\sum_{k=1}^{K}\sum_{m=1}^{N\_s_K}\left(t_{s,m} + t_{s,s}\right). \tag{2}$$

The variables in the equation are interpreted as follows: N_u denotes the aggregate count of students who are presently requesting content, $t_{s,m}$ represents the request latency of the current small base station to the neighboring small base station or macro base station, and $t_{s,s}$ represents the request latency of the student to the small base station to which they are affiliated. Since this paper realizes the prediction of content popularity at the macro base station, the loss rate is defined as the ratio of the number of requests that the macro base station cannot process to the total number of student requests, denoted by $\mathcal{L} = w_r/N\_u$, where w_r represents the number of requests that the macro base station cannot process. Consequently, pre-emptively allocating the widely popular content within a small base station can decrease the mean request latency T of the system and efficiently reduce the system's loss rate $\mathcal{L}$.

3.2. Adversarial Autoencoders-Based Collaborative Multicast Proactive Caching

3.2.1. Student Grouping

In the conventional edge caching network, each base station usually caches the global or local most popular content independently and transmits it unicast. However, in the actual scenario, due to the different preferences of students, the globally popular content often only represents the preferences of some students. Therefore, caching the most popular content at each base station only meets the needs of some students but also causes redundancy and reduces the utilization of cache resources. To meet the needs of different students, the average latency of the system is shortened, and the loss rate is reduced. We study from the students' point of view, predict the local popular content, and consider the cooperative caching among nodes and the use of multicast for distribution. The adversarial autoencoders-based collaborative multicast proactive caching (AAE-CMPC) algorithm consists of three parts: student grouping, cache content prediction, and content replacement and distribution.

Since different students have certain similarities in some preferences, this paper defines the student characteristics $Q_u = \{q_{u_1}, q_{u_2}, q_{u_3}, q_{u_4}, q_{u_5}\}$ according to the student's gender, age, major, learning style, and type of terminal equipment. Because the k-means algorithm has a general clustering effect and ill-conditioned initialization problems on non-convex space, this paper uses the k-means++ algorithm to divide students into I cluster centers [29]. It defines E as the set of cluster centers, $E = \{e_1, e_2, e_3, \cdots, e_I\}$. The k-means++ algorithm works as follows. First, a point u_m is randomly selected as the cluster center e_1, and then the feature distance between other unselected points $u_m{}'$ and e_1 is calculated. Euclidean distance expresses the feature distance, and the calculation formula is shown in equation (4). The point farthest from e_1 is chosen as e_2. And so on, calculate the minimum feature distance between each unselected node $u_m{}'$ and the selected i cluster centers, and then take the node with the most considerable minimum feature distance among all unselected nodes um' as the next cluster center e_{i+1}, which is calculated as follows.

$$e_{i+1} = \arg\max_{m\in[1,M]}\left(\min_{j=1\to i} q(u_m, u_j)\right). \tag{3}$$

$$q(u_m, u_j) = \left\| q_{u_m} - q_{u_j} \right\|^2. \tag{4}$$

After all cluster centers are selected, the characteristic distance between student um and each cluster center is calculated, and the cluster center e_i with the minimum distance is taken as the cluster to which student u_m belongs. The calculation for e_i is as follows.

$$e_i = \arg\min_{i\in[1,E]}\left(\left\| q_{u_m} - q_{u_j} \right\|^2\right). \tag{5}$$

When all students have finished the calculation, each group's new cluster center $u_e{}'$ is recalculated.

$$u_e{}' = \frac{1}{N\_e'}\sum_{n=1}^{N\_e'} q_n, \tag{6}$$

where N_e' denotes the total number of students in the old cluster center e'. Equations (5) and (6) are repeated until the cluster centers are stable and unchanged. At this point, the classification is finished. Students can be divided into groups A, the group set H can be expressed as $H = \{h_1, h_2, h_3, \cdots, h_A\}$, and each student u_m belongs to only a specific group set.

The standard k-means++ algorithm clusters students based on intrinsic features like demographics and learning styles. However, dynamic factors like academic performance, assignments, grades, and learning analytics offer additional clustering dimensions in enhancing English teaching through edge computing. Rather than just grouping students on static traits, incorporating multivariate academic data could better capture emerging language abilities, knowledge, and skills. Assessment results across reading, writing, listening, and speaking categories could be integrated into the distance calculations when identifying cluster centers in k-means++, ensuring student groupings adapt to competency development across diverse aspects of English language learning. Additionally, performance on personalized vocabulary apps, AI-driven writing evaluations, and speech recognition tools could provide regular inputs to the algorithm for responsive cluster updating keyed to individual progress. With edge nodes collecting and transmitting rich performance data, k-means++ could leverage it via academic-oriented proximity metrics between student data points.

The cluster assignments in k-means++ could be updated every two weeks based on the latest vocabulary app usage patterns, writing sample analytic scores, speech recognition metrics, and overall grades. As students demonstrate development across reading, writing, listening, and speaking skills, their relative peer groupings would adapt accordingly based on refreshed statistical proximity. Advanced students may migrate into clusters indicative of their burgeoning capabilities to access more challenging content. Peers exhibiting slower growth could get reassigned, maintaining parity. Rather than one-time grouping, cyclical updates would ensure students enter learning communities congruent with their current competency levels.

The k-means++ algorithm conventionally clusters students based on individual traits and performance data. However, there is potential to advance clustering in peer learning by incorporating team dynamics. Alongside attributes like grades, prior learning styles (e.g., visual, verbal, logical) could provide inputs to optimize group compositions for collaborative learning scenarios. The cluster formation process in k-means++ could assess students on dimensions like conceptual visualization skills, oral discussion abilities, written comprehension aptitudes, and logical reasoning strengths, which would map profiles across learning modalities. Some groups consolidate strong visualizers and logical thinkers to synthesize creative ideas. Others may combine analytical reviewers and eloquent speakers to craft high-impact presentations.

The cluster distance computations could emphasize vocabulary or pronunciation scores more heavily for specific students needing additional development in those areas. For other students, dimensions like sentence construction and logical reasoning could contribute more to distance scoring based on their progress, which would entail maintaining mastery profiles across knowledge dimensions for each learner. With localized processing and low latency data transfer, the edge computing infrastructure could readily sustain such personalized analytics. As students evolve differentially across modalities like reading versus writing, dynamically tuned distance metrics could tighter cluster peers with complementary competencies. It could promote more customized peer learning aligned to intricacies in mastery trajectories. Explaining how to map weights across scoring dimensions to individual learning objectives algorithmically offers research directions. With edge nodes continually updating progress data, responsive weight tuning and cluster re-computations become feasible.

Clustering students into groups with sizes aligned to the cache memory of edge nodes could enable efficient content multicasting. Larger clusters may overburden cache storage and undermine low-latency transmission. Smaller groupings could underutilize available edge resources, leading to redundancies. Exploring computational techniques to dynamically size clusters based on edge infrastructure constraints provides research potential. K-means++ computations could assess edge node attributes like CPU capacities, co-located cache sizes, and wireless bandwidth to statistically derive apt peer group sizes, maximizing on-device computations. The low latency data transfers facilitated by edge networks can sustain the reliable gathering of such infrastructure specifications. Additionally, the algorithm could evaluate the versatility of emerging edge hardware like MMPUs and shape cluster dimensions accordingly to harness specialized processing.

3.2.2. Cache Content Prediction

In terms of cache content prediction, because AAE can learn the potential characteristics of students, it can accurately predict the content that the grouped user group may request in the future according to the historical request records of students (students' preferences). Therefore, this paper will predict the content popularity of each group based on AAE.

AAE is a probabilistic autoencoder (AE) that combines generative adversarial networks (GAN) and variational autoencoders [30]. Its model architecture consists of two parts (Fig. 2). (i) The top half is AE, which can learn the latent variable z (z represents the latent features of the student) in an unsupervised manner. (ii) The bottom half is GAN, which discriminates whether the sample z is from the prior distribution $p(z)$ or the latent variable generated by AE.

The training of AAE involves a two-stage process of reconstruction and regularization, whereby the architecture of AE is augmented with a GAN to enable AE to function as a generative model within GAN. During the reconstruction phase, AE is employed to revise the encoder to minimize the reconstruction error of $\boldsymbol{X}$. First, and the hidden variable z is generated by the generative network $q(z|x)$. z reconstructs $\boldsymbol{Y}$ through the decoder $p(x|z)$, and the loss of

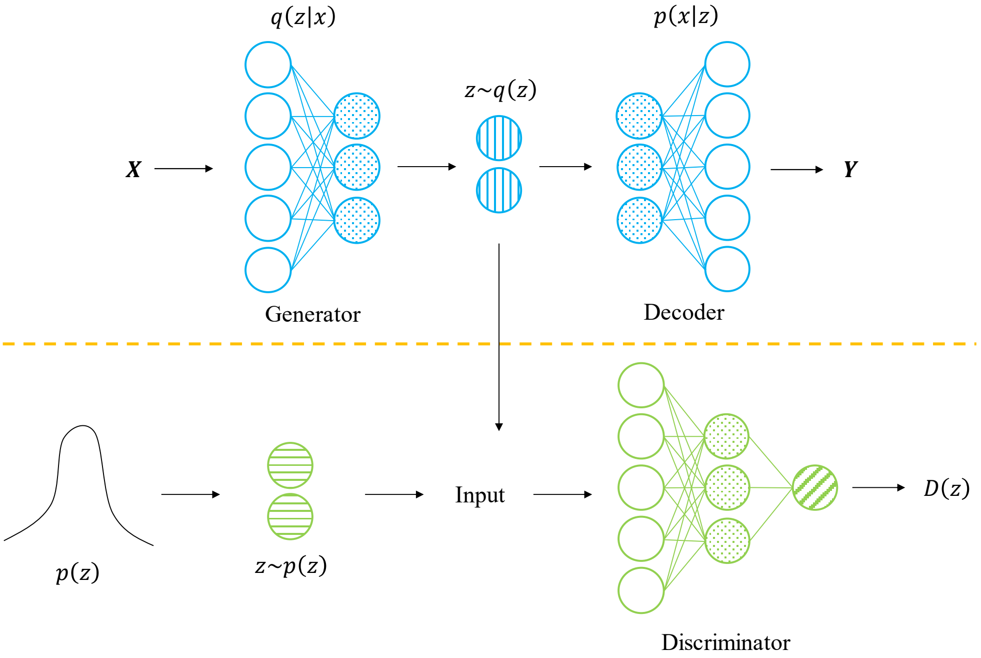

Fig. 2. Architecture of AAE.

the reconstruction of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is calculated. In the regularization stage, the discriminator first identifies whether the sample z is from the generated sample or the prior distribution to update the parameters. Then, to deceive the discriminator D, generator G will also be updated. Through the mutual game between the generator G and the discriminator D, the output of the discriminator D is maximized, and the output of the generator G is minimized, so the min-max game between G and D can be expressed as follows.

$$\min_{G}\max_{D} E_{x\sim\xi_{q(x)}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right], \tag{7}$$

where E denotes the desired distribution and $\xi_{q(x)}$ denotes the input data distribution.

During the training process, the output of the discriminator is transmitted to the encoder through the adversarial network so that the hidden variable z is close to the distribution of $p(z)$. The weights of discriminator D are adjusted by backpropagation while the parameters of generator G are updated. The above process is repeated, and when the training is finished, the autocoded decoder is defined as the generative model. The prior distribution $p(z)$ is mapped to the data distribution $\xi_{q(x)}$, so the adversarial autoencoder can achieve $q(z)$ matching $p(z)$ in the regularization stage, where $q(z)$ is the aggregated posterior distribution, which is defined as follows.

$$q(z) = \int q(z|x)\, \xi_{q(x)}\, \mathrm{d}x. \tag{8}$$

The loss function for the discriminator in training is defined as follows.

$$\mathcal{L}_D = -\frac{1}{b}\sum_{a=1}^{b} \log\left(D(z')\right) + \log\left(1 - D(z)\right), \tag{9}$$

where b is the size of the batch data volume for each network training, the adversarial generator loss function is given below.

$$\mathcal{L}_G = -\frac{1}{b}\sum_{a=1}^{b} \log\left(D(z)\right). \tag{10}$$

Each group of student history search content matrix $\boldsymbol{X}$ is used as the input of the training model. $\boldsymbol{X}$ consists of sample variables x, $\boldsymbol{X} \in N^{A\times N}$, where A and N denote the number of user groups and the amount of requested content. In this case, the content requested by the user group ha is marked as interested. Additionally, the content of a student's future request is also related to the student's preference. To predict the content that students with different preferences may request, this paper takes the preference information as an additional matrix of the input information $\boldsymbol{X}$. Since unknown content and uninteresting content are mixed in the unrequested content, it is challenging to distinguish uninteresting content. However, marking all unrequested content as uninteresting is a bias prediction. Therefore, this paper uses random marking to mark whether the unknown content is of interest, and the probability of random marking is related to the student's preference for the content. AAE learns z from the input matrix $\boldsymbol{X}$, and then the prediction matrix $\boldsymbol{Y}$ is obtained from z. The contents are ranked according to the probability predicted by matrix $\boldsymbol{Y}$, and the highest-ranked contents are deployed to small base stations and macro base stations.

In the AAE prediction model, unlabeled content poses challenges regarding categorization as interested or uninterested data points. Simply encoding unknown content as uninterested can bias the model. To handle this, a probabilistic marking idea is proposed. The core premise is that for any student, the likelihood of unfamiliar content being relevant to them could be estimated from their preferences. For instance, a learner engaging frequently with science-related materials could imply a higher probability of unencountered science content being attractive to them. Similarly, a student with arts and design inclinations could have a higher probability of unfamiliar arts content being deemed captivating. In essence, individual interests and patterns of prior content interactions can guide likelihood estimates for categorizing unlabeled content.

Computationally, this translates to a randomized marking approach that assigns interest tags with probabilities tied to user preferences. Therefore, students would have content and a probability distribution over interest categories derived from their usage history. Then, unfamiliar content would get allocated randomized tags based on those category-wise probabilities. Effectively, this statistical supplementation allows some guesswork in gauging interest in new content by deriving odds from existing consumption behaviors. Caveats exist regarding heaping assumptions from limited user histories that warrant investigation before substantiating the approach as robust. However, directionally, the probabilistic marking paradigm offers the potential to improve the modeling of unlabeled data. The essence relies on extrapolating user content preferences onto unexplored materials through randomized interest assignments guided by probabilities.

3.2.3. Content Replacement and Distribution

Regarding content caching and distribution, it can be observed that the transmission latency between small base stations is significantly lower than the transmission latency to macro base stations due to the smaller transmission distance between small base stations in comparison to the distance between small base stations and macro base stations [31]. Therefore, this paper will combine the cooperation between small base stations and multicast content transmission to achieve the minimum average latency of the system.

Through the prediction of AAE, the request probability matrix $\boldsymbol{Y}$ of each content for each group h_a can be obtained, and the request probability of each content is superimposed and ranked. Then the request probability is sequentially considered to place the position from high to low. The placement rules are as follows: First, the request probability of requesting the popular content c_n can be obtained by prediction, and the small base stations requesting the most popular content c_n form a set, and the node with high request probability among the small base stations requesting the content c_n is taken as the source node, and the other nodes in the set are taken as the destination nodes. Transmitting data from a source node to multiple destination nodes is called a multicast tree. Additionally, the transmission from a small base station to the user end is also conducted in a multicast manner. Finally, the node with the minimum average request latency from the current position to all requesting users and sufficient storage resources is selected as the deployment location of c_n. Therefore, the optimal problem with the average request latency of the system as the optimization objective under the cooperative multicast scheme can be defined as follows.

$$\min T = \frac{1}{N\_u}\sum_{n=1}^{N\_s}\sum_{k_1=1}^{K}\gamma_{k_1,n}\times\sum_{k_2=1}^{K}\left(t_{k_1,r_n}+\sum_{j=1}^{J_{k_2}}t_{h_j,c_n}\right). \tag{11}$$

$$t_{k_1,r_n} = \max\{t_{k_1,r_n}^1, t_{k_1,r_n}^2, \cdots, t_{k_1,r_n}^J\}, n = 1,2,3,\cdots,N. \tag{12}$$

$$J+1 \le K. \tag{13}$$

$$\textstyle\sum_{k=1}^{K}\gamma_{k_1,n} = 1, n = 1,2,3,\cdots,N. \tag{14}$$

$$\textstyle\sum_{n=1}^{N}\gamma_{k_1,n} \le V_{s_{k_1}}, k_1 = 1,2,3,\cdots,K. \tag{15}$$

$$\gamma_{k_1,n} = \{0,1\}. \tag{16}$$

$$t_{h_j,u_m} = \begin{cases} t_s, & c_n \in y_1 \cup y_2 \cup y_3 \\ 0, & c_n \in v_u \end{cases}. \tag{17}$$

In equation (11), N_u represents the total number of user requests at the current time, N_s represents the top N_s of all content popularity sorted from high to low, and its value is equal to the sum of the capacity of all small base stations. Equation (12) reflects the maximum latency experienced by the source node's small base station while communicating with all the small base station nodes of the destination. Per equation (13), the summation of the number of small base stations requesting content c_n and the number of nodes placing content c_n must not exceed the total count K of small base stations. Due to the limited storage resources of small base stations, this paper considers the cooperation of each small base station to implement caching to make full use of the storage containers of each small base station and reduce redundancy and loss rate. Equation (14) indicates that only one copy of each content is cached in the system. Equation (15) represents the capacity limit of each small base station; Equation (16) represents the deployment matrix of small base stations. When $\gamma_{k_1,n} = 1$, the small base station s_k caches the content c_n; otherwise, it does not cache. The transmission latency for cases where the requested content is located in either the local or small base station is expressed by equation (17). To reduce the frequent requests for the small base station, this paper divides the storage area of the small base station into three parts where y_1 is the main buffer, which is used to cache the content deployed by the ant colony algorithm, and y_2 is the high-speed buffer, which is used to store and update the content of each request. The next time the content in y_2 is reaccessed, the content will be moved to the y_3 hot cache. When y_3 reaches its capacity limit, it will be replaced according to the request frequency.

The essence of solving $\gamma_{K,n}$ is the constrained 0−1 knapsack problem, a classical NP-Hard problem. If solved directly, its time complexity is too large, but the intelligent optimization algorithm can solve this problem well. The ant colony algorithm has a robust global search ability compared with other intelligent optimization algorithms. It adapts to the changed environment through cooperation between ants, thereby increasing the probability of finding the optimal global solution. Therefore, this paper will use the ant colony algorithm to solve the deployment matrix.

The ant colony algorithm is derived from an algorithm obtained by observing the foraging process of ants [32]. Studies have shown that ants choose the forward path in searching for food by the solubility of pheromone on the path and release pheromone on the selected path. Because pheromones will evaporate with time, and ants choose the following path by sensing the strength of pheromone concentration, the system will gradually stabilize from the initial random path search to the shortest path. The conventional ant colony algorithm is designed to pursue a single target, necessitating a substantial number of iterations. The AAE-CMPC scheme proposes a method for optimizing multi-objective search and enhancing iteration speed by considering small base stations caching content c_n as caves and small base stations requesting content c_n as food.

To avoid the solution falling into the local optimum, when the ant is located at the current node i, the pseudorandom proportional state transition rule is used to select the next node j to increase the probability of choosing a random path.

$$j = \begin{cases} \arg\max\limits_{u\in N^i}\{\varepsilon(i,u)^\alpha\eta(i,u)^\beta\}, & g \le \theta \\ P_{ij} & \text{others} \end{cases}. \tag{18}$$

$$P_{ij} = \begin{cases} \frac{\varepsilon(i,j)^\alpha\eta(i,j)^\beta}{\sum_{u\in N^i}\varepsilon(i,u)^\alpha\eta(i,u)^\beta}, & j \in N^i \\ 0 & \text{others} \end{cases}, \tag{19}$$

where g is a random number with uniform $[0,1]$ distribution, and θ is a given parameter that determines the exploration and exploitation weights ($\theta \in [0,1]$). The selection rule of P_{ij} is shown in equation (19), where $\eta(i,j)$ represents the heuristic information from node i to node j, generally taking the reciprocal of the distance between node i and j, and the reciprocal of the delay. N^i is the set of following alternative nodes, $\varepsilon(i,j)$ represents the pheromone concentration from node i to j, the initial value is set to 1, and the rule for each update is shown in equation (20). α and β denote the weight parameters of pheromone and heuristic information, respectively, which determine the proportion of $\eta(i,j)$ and $\varepsilon(i,j)$ in the decision-making process. According to equation (19), when more pheromones are on the path, and the distance is short, the probability of selecting this path will be more significant.

Pheromone updates are divided into two types: the local pheromone update and the global pheromone update. The local pheromone's update rule is that the ant has chosen this path and released the pheromone on this path. Due to the volatility of the pheromone, the update rule of the local pheromone is shown in equation (20).

$$\varepsilon(i,j) \leftarrow (1-\psi)\varepsilon(i,j) + \psi\Delta\varepsilon(i,j). \tag{20}$$

$$\Delta\varepsilon(i,j) = \begin{cases} \left(L_{i,j}\right)^{-1}, & j \in \{\text{target}\} \\ 0 & \text{others} \end{cases}, \tag{21}$$

where ψ is the volatilization factor of pheromone, $0 < \psi < 1$. $\Delta\varepsilon(i,j)$ is the local pheromone update value of path (i,j), and the calculation rule is given in equation (21). $L_{i,j}$ is the path length from current node i to next node j. When the ant finds the destination node, or there is no next node to choose from, the additional update rule is shown in equation (22).

$$\varepsilon_{\text{path}} = \begin{cases} \varepsilon_{\text{path}} + \Delta\varepsilon(\text{path}), & \text{if find} \\ (1-\psi)\varepsilon_{\text{path}} & \text{others}' \end{cases} \tag{22}$$

where find indicates that the destination node has been found and the reward pheromone has been added to the whole path. At this point, the pheromone evaporation mechanism is performed on the entire path to avoid selecting this path again. $\varepsilon_{\text{path}}$ denotes the path traversed from the starting node to the current node, and $\Delta\varepsilon(\text{path})$ is the pheromone of path reward.

In the content distribution phase, students u_m request different content, and the small base station distributes the content to users according to the situation requested by students. When the student u_m requests the content c_n, if the memory V_s of the affiliated small base station does not contain the content c_n, the content c_n is obtained through collaboration between the small base stations or the macro base station. If V_s contains content c_n or has acquired content c_n, the requested content c_n is distributed to the requesting student multicast. Since each student um belongs to a group h_a, if the content c_n requested by student u_m is the most popular content in the h_a group, the content c_n is actively cached to the student u_m that has not sent the request at the current time in the h_a group by multicast, instead of actively caching in the off-peak traffic period. In this way, it realizes active caching and saves energy. If the student capacity V_u reaches the upper limit, the content is replaced according to the popularity of the content.

The ant colony algorithm used for content caching and distribution has a tradeoff between exploitation and exploration when ants traverse paths to place content across edge nodes. Exploitation leverages learned knowledge to optimize placements based on past information. Exploration involves some degree of randomization to discover better solutions. To balance this, a pseudo-random proportional transition rule is introduced. The core idea was to incorporate some degree of arbitrary path probabilistically transitions to inject exploration amongst the exploitation-focused pheromone-driven walks.

To minimize latency, the content caching scheme allocates predicted popular content across distributed edge nodes. The multivariate allocation for optimizing caching locations is non-convex, combinatorial, and NP-hard. Simple greedy heuristics get trapped in local optima. Genetic algorithms require prohibitive cross-over computations. However, ACO offers several advantages aligned to the caching specification without the downsides. First, ACO allows adaptive discovery of globally optimal solutions via simulated ant walks guided by accumulating pheromone traces towards reward spots, which handles non-convex objectives. Second, the probabilistic transition function balances focused local search with exploratory random walks to avoid entrapment. Next, concurrent, collaborative walks parallelize evaluations to improve efficiency. The pheromone evaporation mechanism also promotes diversity. Finally, incremental computations during ant trails make it scalable for combinatorial problems. These adaptation characteristics, multi-objective search efficiency, randomness injection, and computational parallelism tailor ACO for optimized edge caching distributions versus alternatives. The ants probabilistically transitioning between nodes based on distance-latency pheromone concentrations can iteratively discover superior allocated configurations.

Within the AAE-CMPC edge caching framework, the key purpose of the AAE model is to predict content popularity for specific student groups by leveraging their historical interactions. The input to the AAE model is a matrix representing previous content requests by various student groups over time. Encoded as matrices, this interaction data trains the adversarial autoencoder in an unsupervised manner to learn latent representations reflecting content preference patterns for different student clusters.

The encoder module in the trained AAE model captures intrinsic content preferences and taste dimensions based on past consumption history. The decoder then uses these latent features to reconstruct likely content affinity distributions for targeted user groups. Therefore, the trained AAE model can predict preferences and probable content requests for new students mapped to specific clusters by utilizing the encoded latent patterns learned from past observations.

These content popularity predictions, encoded as request probability distributions over content catalogs for student groups, become inputs for the ant colony edge caching optimization. Computationally, the AAE model provides the predictive analytics to determine what content to cache where based on group and content latent dimensions derived through adversarial reconstruction mechanisms.

IV. SIMULATION AND RESULTS ANALYSIS

4.1. Parameters Setting

The simulation scenario comprises a singular content server, a solitary macro base station, ten small base stations, and a group of students. The simulation environment used in this study is founded on the simulation platform described in the reference [33]. This paper adds small base station equipment while retaining some parameters according to the actual scene. Assuming that the size of the transmitted content is 6 MB, the transmit rate between the student and the small base station is 2 MB/s, the transmit rate from the student to the macro base station is 1.2 MB/s, the transmit rate from the student to the cloud is 1 MB/s, and the transmit rate between adjacent small base stations is 24 MB/s. Therefore, the transmission latency between different terminals is $t_l = 0$, $t_s = 3$, $t_m = 5$, $t_c = 6$, and $t_{ss} = 0.25$, respectively. Through multiple simulations and comparison of loss functions with different values, the final number of groups $A = 20$ is taken. The key simulation parameters are described in Table 1.

Regarding content prediction, the content requested by users and the dataset for AAE training come from MovieLens 1M Dataset, which includes 3,883 movies, 6,040 users, and 1,000,209 user ratings [34]. To ensure sufficient iterations, this paper preprocesses the original data to delete users with less than 50 user records. The training and prediction of AAE are implemented based on PyTorch. Regarding the parameter design of the ACO algorithm, the choice of pheromone and heuristic factor, as well as θ, determines whether the problem of premature stagnation or falling into local optimum will occur during the exploration process. According to repeated experiments and comparisons, this paper sets $\alpha = 1$, $\beta = 5$, $\psi = 0.1$, and $\theta = 0.3$. In the whole simulation process, the essential information and preference information of each user are derived from the data of real users. Each requested content will randomly request content in its preference domain. To better simulate the actual situation, whether the user requests at a specific moment is also random.

Table 1. simulation parameters.

Parameter	Value
Number of small base stations	10
Number of macro base stations	1
Content server	1
Number of users	Varied
Content size	6 MB
Student to small base station rate	2 Mbps
Student to macro base station rate	1.2 Mbps
Student to cloud rate	1 Mbps
Inter-base station rate	24 Mbps
Student to small base station latency	3 ms
Student to macro base station latency	5 ms
Student to cloud latency	6 ms
Inter-base station latency	0.25 ms

4.2. Results Analysis

To verify the role of collaboration and multicast in edge caching, this paper first evaluates whether random caching (RC) adopts four strategies combining collaboration and multicast, i.e., (i) No collaboration and multicast (RC-N). (ii) Only collaboration (RC-C). (iii) Only multicast (RC-M). (iv) Collaboration and multicast (RC-CM). The users in the simulation use the first 500 users in the MovieLens 1 M Dataset, the storage capacity of the macro base station is 200, and the number of iterations is 50. The simulation of its execution time and the system average request latency is shown in Fig. 3. To illustrate the changes in the four strategies with the number of users, based on the above simulation, this paper makes the number of users change from 0 to 2,000, increases 40 users each time, and iterates 50 times each time and calculates the average of the results after each iteration. Fig. 4 depicts the results of the simulation.

Fig. 3 shows that collaboration and multicast schemes can reduce the average transmission latency of the system. The mean value of each group of data was calculated. The results showed that the transmission latency of the collaborative strategy was reduced by 0.13 s while using the multicast strategy resulted in a decrease of 0.08 s. This is because the essence of collaboration is to jointly consider and cache the adjacent small base stations so that the storage

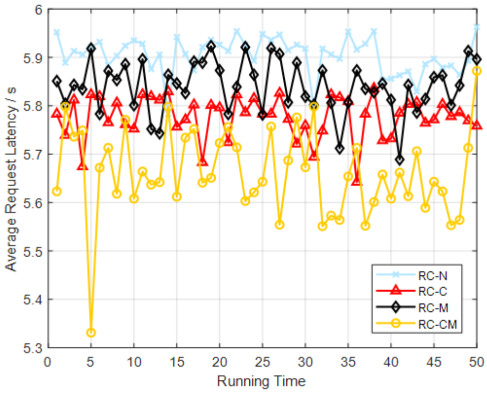

Fig. 3. Average latency comparison of different strategies with RC.

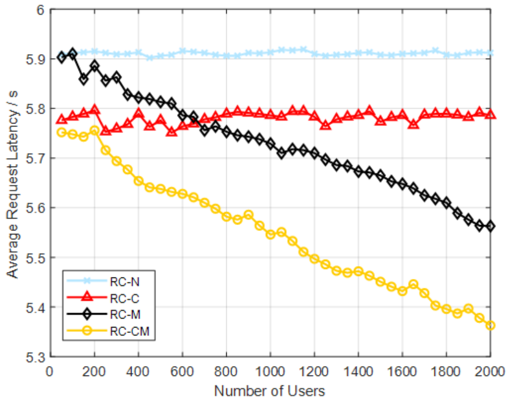

Fig. 4. Average latency of different strategies versus the users' number.

capacity of the adjacent small base stations can be shared, which is equivalent to increasing the storage capacity of the current small base station, so the average latency of the system is reduced. As illustrated in Fig. 3, the multicast strategy's efficiency could be better than that of the collaborative strategy. The result is consistent with real-world scenarios, where the likelihood of multiple users concurrently requesting identical content is low, thereby rendering the impact of multicast less pronounced. Fig. 4 illustrates that the average latency of the multicast and collaboration strategies intersect as the number of users increases, with the collaborative strategy's impact being surpassed by that of the former. This is because as the number of users increases, the probability that different users will request the same content increases, so the multicast strategy performs better. Additionally, it can be seen from Figs. 3 and 4 that the effect of using a random caching algorithm is poor, and the average transmission latency is still above 5 s; that is, most of the requested content needs to be obtained from the cloud.

To verify the effect of the proposed proactive caching scheme combining AAE content prediction and multicast on reducing the average request latency, this paper will repeat the first simulation and replace the RC algorithm in the simulation with the AAE-CMPC algorithm, as shown in Fig.

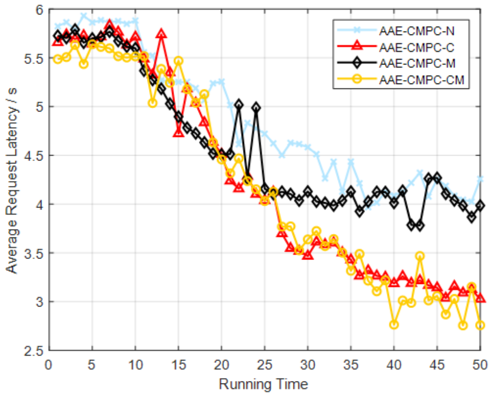

Fig. 5. Average latency comparison of different strategies with AAE-CMPC.

5. While AAE-CMPC-N, AAE-CMPC-C, AAE-CMPC-M, and AAE-CMPC-CM represent the AAE-CMPC algorithm combined with no collaboration and multicast, only collaboration, only multicast, and collaboration and multicast respectively.

After verifying the effect of collaboration and multicast, this paper compares AAE-CMPC with TCO [26], LeaD [27], and CREAT [28], as follows.

- TCO: An efficient algorithm, called task caching and offloading (TCO), based on alternating iterative algorithm.
- LeaD: Cache deployment strategy, i.e., large-scale WiFi edge cache deployment.
- CREAT: A new algorithm in which blockchain-assisted compressed algorithm of federated learning is applied for content caching, called CREAT to predict cached files.

Two key metrics are used for evaluation - system latency and cache loss rate. Average request latency reflects the average delay end users face in accessing requested content. Meanwhile, the loss rate computes the fraction of content requests that cannot get served from edge caches, resulting in transmissions from distant cloud servers.

The simulation results are shown in Fig. 6. Fig. 6 shows the simulation comparison between AAE-CMPC and three benchmarks under the collaborative multicast strategy. The AAE-CMPC scheme employs a holistic approach to joint optimization and outperforms other schemes to reduce the system's average transmission latency. As the number of iterations increases, the average transmission latency of the system is reduced, and the average transmission latency is also decreasing. This is because, through AAE's prediction of content popularity, the macro base station can predict the user's request intention and pre-cache the content that may be requested. Fig. 6 depicts that the mean transmission latency remains below 3 s, indicating that a significant portion of the requested content has been cached at the small base station. Conversely, the average latency of the three

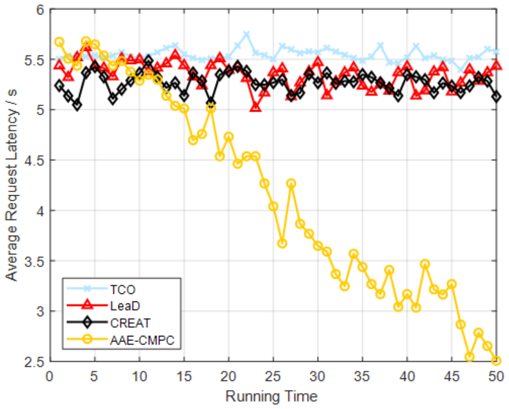

Fig. 6. Average latency comparison with different algorithms.

benchmarks exceeds 5.5 s.

The proposed AAE-CMPC algorithm focuses on a collaborative multicast strategy and aims to optimize the overall joint performance by reducing the average transmission latency of the system. It can enhance higher vocational English teaching by reducing latency and enabling instantaneous feedback. The AAE-CMPC algorithm utilizes content popularity prediction through an AAE model, enabling the macro base station to predict the user's request intention and pre-cache the content that may be requested. In higher vocational English teaching, relevant teaching materials, resources, or multimedia content can be pre-cached at small base stations closer to the students. Having the content readily available at the small base stations can significantly reduce the latency for accessing teaching materials. Students can quickly access the required materials without waiting for data to be fetched from distant servers, and the reduced latency ensures students can access the content they need promptly, enabling a seamless learning experience. The reduced latency facilitated by the AAE-CMPC algorithm can also enable instantaneous feedback in higher vocational English teaching. For example, if the teaching materials include interactive quizzes or assessments, students can receive immediate feedback on their responses. With traditional systems that rely on high latency, students may experience delays in receiving feedback, which can hinder the learning process, as students may need help correcting their mistakes or promptly reinforcing their understanding. However, with lower latency enabled by AAE-CMPC, students can receive feedback on their performance almost instantly, allowing them to promptly address any misconceptions or improve their skills. The reduced latency provided by the AAE-CMPC algorithm can enhance the interactivity and real-time collaboration aspects of higher vocational English teaching. For instance, if the teaching platform includes features like live video conferencing or collaborative document editing, the lower latency ensures smoother and more effective communication between instructors and students. Moreover, students can actively participate in real-time discussions, ask questions, and receive immediate responses from instructors or peers. This interactivity promotes engagement and active learning, as students can contribute to the learning process without being hindered by high latency issues. The AAE-CMPC algorithm can enhance higher vocational English teaching by creating a more efficient and interactive learning environment, reducing transmission latency and enabling instantaneous feedback. Students can access teaching materials quickly, receive feedback promptly, and actively engage in real-time collaboration, leading to improved learning outcomes.

Finally, this paper verifies the accuracy of AAE prediction by simulating the loss rate. The simulation involves a user group of 500, with the macro base station's capacity ranging from 0 to 800 and increasing by 16 at each interval. The process involves performing 50 iterations and subsequently computing the mean output values, as depicted in Fig. 7. Fig. 7 illustrates that an increase in storage capacity of the macro base station results in a decrease in loss rate for the four algorithms. This is due to the ability of the macro base station to cache more content, thereby increasing the hit ratio and reducing the cache loss rate. It can be found that the proposed AAE-CMPC scheme has a lower loss rate than that of the three benchmarks.

Numerically, the average latency and loss rate are shown in Table 2.

The proposed edge caching scheme benefits low-latency content delivery to enrich teaching. However, translating these information-theoretic gains into learning outcomes

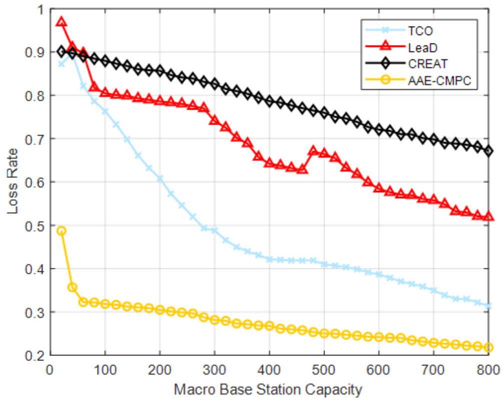

Fig. 7. Loss rate versus cache capacity of macro base station.

Table 2. Comparison of average latency and loss rate.

Algorithms	Average latency (s)	Loss rate
TCO	5.55	0.50
LeaD	5.35	0.69
CREAT	5.27	0.78
AAE-CMPC	4.17	0.27

involves bridging technological possibilities with practical constraints. Modularly integrating the prediction, optimization, and personalization components requires overcoming enterprise challenges. Inventorying existing IT assets and formulating execution roadmaps needs administrator buy-in. Gradually transitioning current monoliths into micro-services-based edge-native architectures mandates alignment across teams. Moreover, the reliability and security implications of distributed caching need evaluations considering access control policies. Quantifying returns on investments and navigating budgetary approvals across stakeholders could pose adoption barriers. Beyond technical integrations, selling the vision of data-driven, personalized learning crucially hinges on addressing teacher concerns regarding transparency and agency. Ongoing demos and constructive feedback cycles are imperative. In summary, alongside algorithmic advancements, holistic frameworks factoring procedural, social, and economic realities warrant equal attention to fulfill the promise of enhanced pedagogies through edge computing.

The AAE-CMPC algorithm revolutionizes higher vocational English teaching by effectively reducing latency through edge caching optimizations, enabling instantaneous learner feedback essential for language acquisition. Predicting content popularity and proactively placing materials on edge nodes nearer to students minimizes transmission delays to access teaching resources or assessments. This allows prompt evaluation of student input, including vocabulary usage, pronunciation, grammar accuracy, etc., with automated feedback on corrections dispatched instantly over the low-latency edge connections. The real-time responses keep students iteratively improving language construction without falling into the habituation of errors that delays would cause. Such tightly coupled review cycles catalyzed by sub-second system lag times allow personalized, adaptive learning. Progress data streams back to models, updating student cluster profiles, refining group-wise content popularity predictions, and caching distributions for perpetually enhancing teaching quality. The system stimulates an interactive paradigm with agile feedback tailored to individual needs. While the essence is enabling micro-iterative personalized progress tracking and guidance by reducing lag times to nearly imperceptible levels using edge-centric optimizations.

The AAE-CMPC algorithm optimizes content caching and storage at the macro base station based on content popularity prediction. As the storage capacity of the macro base station increases, it can cache more content relevant to higher vocational English teaching, meaning that a more considerable amount of teaching materials, resources, or multimedia content can be stored at the edge, closer to the students. With increased storage capacity, the macro base station can hold diverse teaching materials, including videos, audio files, e-books, or interactive applications. The availability of a wide variety of teaching resources enables a richer and more comprehensive learning experience for higher vocational English students. The AAE-CMPC algorithm's ability to predict content popularity helps optimize the cache hit ratio. When the storage capacity of the macro base station increases, more content can be cached, leading to a higher probability of content being readily available at the edge, reducing the cache loss rate, meaning that students are more likely to find the requested teaching materials already cached at the edge, resulting in faster access times. Additionally, the AAE-CMPC algorithm ensures that students can access the required teaching materials without experiencing delays due to content retrieval from remote servers. The efficient retrieval process facilitated by edge computing enhances the learning experience by providing seamless and instant access to resources. The AAE-CMPC algorithm offers improved reliability and scalability for higher vocational English teaching. Since the teaching materials are stored at the edge, closer to the students, they are not solely dependent on a centralized server or data center. This decentralized approach reduces the risk of network congestion or server failures affecting access to teaching materials. Likewise, edge computing facilitates scalability in proportion to the growth of the user base. The AAE-CMPC algorithm can efficiently manage and distribute content based on predicted popularity, ensuring that teaching materials are available even during peak usage. Edge computing, enabled by the AAE-CMPC algorithm, significantly reduces latency by bringing the teaching materials closer to the students. With edge-based caching, students can access teaching materials with minimal delay, enhancing the real-time nature of interactions, assessments, and feedback. The lower latency facilitates real-time collaboration, interactive exercises, and instant feedback, as students can seamlessly interact with teaching materials and instructors without being hindered by network latency. Students can participate in virtual classrooms, engage in live discussions, or receive immediate feedback on their progress, promoting active learning and engagement. Using edge computing and the AAE-CMPC algorithm, higher vocational English teaching can be enhanced through increased storage capacity, reduced cache loss rate, improved reliability and scalability, lower latency, and enhanced interactivity. These advancements contribute to a more efficient and immersive learning experience, empowering students to access teaching resources seamlessly and enabling effective knowledge acquisition.

While the AAE-CMPC scheme demonstrates promising improvements, certain limitations exist. Firstly, the evaluation involved movie rating datasets that have clear contextual patterns. However, applicability to multidomain educational content with greater diversity needs validation. Content variety could impede prediction accuracy. Next, population sizes were limited to the order of thousands. Scaling to larger groups requires hierarchical clustering and distributed model parallelization.

Furthermore, the algorithms entail several configurable parameters like pheromone decay factor and cluster dimensions. Suboptimal tuning could undermine caching gains seen in controlled simulations. Additionally, optimized edge cache allocation necessitates extensive monitoring of node loads. This telemetry gathering could add considerable coordination overhead, eroding networking Fabric efficiencies. Finally, user studies are imperative to assess true efficiency gains in learning outcomes versus synthetic request patterns alone.

While the evident potential exists, translating these information-theoretic improvements to actual student comprehension requires further investigation. Testing factors like model generalization across topics, robustness to parameter tuning, alternative predictive models, hierarchical scaling architectures, and evaluation against real-world usage would help mature the solutions.

CONCLUSION

In higher vocational English teaching, the prompt delivery of teaching materials and the facilitation of instantaneous feedback are pivotal for effective language learning. By combining the advantages of small base station cooperation, multicast, and predictable user behavior, the AAE-CMPC algorithm offers an innovative approach. The AAE-CMPC algorithm begins by categorizing students into different preference groups based on their characteristics. It then uses AAE to predict the content each group will likely request. To reduce cache redundancy, an ant colony algorithm is employed to pre-deploy the predicted content across small base stations, fostering collaboration between them. During content distribution, if a student within a group requests popular content, it is actively cached and shared with other students in the group who have yet to make the same request. Otherwise, the content is distributed conventionally. The superiority of the AAE-CMPC scheme is demonstrated through comparative analysis with three benchmarks. The simulation results validate that an increase in the storage capacity of the macro base station leads to a reduction in the loss rate, which is attributed to the proactive caching approach that enhances cache hit ratios. The AAE-CMPC algorithm revolutionizes higher vocational English teaching by effectively reducing latency, enabling instantaneous feedback, and streamlining the learning process for students. It empowers them to access teaching materials promptly, receive real-time feedback on their progress, and engage seamlessly in collaborative activities. Moreover, the framework leverages edge computing, facilitating increased storage capacity, scalability, and reliability, enhancing the learning experience. However, there are certain limitations and avenues for future work. First, the AAE-CMPC algorithm assumes predictable user behavior and relies on accurate content popularity predictions. Further research is needed to explore more robust and accurate prediction models to handle variations in user preferences and dynamically changing content popularity. Then, the algorithm's performance should be evaluated under diverse network conditions and scaled-up scenarios to ensure its applicability in larger educational contexts. Additionally, it would be valuable to investigate the potential impact of the AAE-CMPC scheme on the network infrastructure and resource allocation to ascertain its feasibility and practical implementation.

REFERENCES

[1] C. M. Chen, J. Y. Wang, and M. Lin, "Enhancement of English learning performance by using an attention-based diagnosing and review mechanism in paper-based learning context with digital pen support," *Universal Access in the Information Society*, vol. 18, pp. 141-153, 2019.

[2] Z. Yu, W. Xu, and P. Sukjairungwattana, "Motivation, learning strategies, and outcomes in mobile English language learning," *Asia-Pacific Education Research*, vol. 32, no. 4, pp. 545-560, 2022.

[3] P. Zhang, "Cloud computing English teaching application platform based on machine learning algorithm," *Soft Computing*, 2023.

[4] H. Guo and X. Jiang, "English teaching evaluation based on reinforcement learning in content centric data center network," *Wireless Networks*, 2022.

[5] S. Löfgren, L. Ilomäki, J. Lipsanen, and A. Toom, "How does the learning environment support vocational student learning of domain-general competencies?," *Vocations and Learning*, vol. 16, no. 2, pp. 343-369, 2023.

[6] X. Wang, Y. l. Liu, B. Ying, and J. Lin, "The effect of learning adaptability on Chinese middle school students' English academic engagement: The chain mediating roles of foreign language anxiety and English learning self-efficacy," *Current Psychology*, vol. 42, no. 8, pp. 6682-6692, 2023.

[7] V. Hurbungs, V. Bassoo, and T. P. Fowdur, "Fog and edge computing: Concepts, tools and focus areas," *International Journal of Information Technology*, vol. 13, no. 2, pp. 511-522, 2021.

[8] C. Ding, A. Zhou, J. Huang, Y. Liu, and S. Wang, "ECDU: An edge content delivery and update frame-

work in mobile edge computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 268, 2019.

[9] X. Liu, J. Wang, Q. F. Song, and J. H. Liu, "Colla-borative multicast proactive caching scheme based on AAE," *Computer Science,* vol. 49, no. 9, pp. 260-267, 2022.

[10] Y. Chen, S. Chen, and X. Chen, "Efficient caching strategy in wireless networks with mobile edge computing," *Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1756-1766, 2020.

[11] M. Yasir, S. K. uz Zaman, T. Maqsood, F. Rehman, and S. Mustafa, "CoPUP: Content popularity and user preferences aware content caching framework in mobile edge computing," *Cluster Computing*, vol. 26, no. 1, pp. 267-281, 2023.

[12] L. Chaudhary and B. Singh, "Autoencoder model using edge enhancement to detect communities in complex networks," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 1303-1314, 2023.

[13] P. Dai, J. Luo, K. Zhao, H. Xing, and X. Wu, "Stacked denoising autoencoder for missing traffic data reconstruction via mobile edge computing," *Neural Computing and Applications*, vol. 35, no. 19, pp. 14259-14274, 2023.

[14] C. Hou, L. Hua, Y. Lin, J. Zhang, G. Liu, and Y. Xiao, "Application and exploration of artificial intelligence and edge computing in long-distance education on mobile network," *Mobile Networks and Application*, vol. 26, pp. 2164-2175, 2021.

[15] R. Zhu, L. Liu, H. Song, and M. Ma, "Multi-access edge computing enabled internet of things: Advances and novel applications," *Neural Computing and Application*, vol. 32, pp. 15313-15316, 2020.

[16] G. Rong, Y. Xu, X. Tong, and H. Fam, "An edge-cloud collaborative computing platform for building AIoT applications efficiently," *Journal of Cloud Computing*, vol. 10, no. 1, p. 36, 2021.

[17] J. Li, D. Shi, P. Tumnark, and H. Xum, "A system for real-time intervention in negative emotional contagion in a smart classroom deployed under edge computing service infrastructure," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 1706-1719, 2020.

[18] F. Li and C. Wang, "Artificial intelligence and edge computing for teaching quality evaluation based on 5G-enabled wireless communication technology," *Journal of Cloud Computing*, vol. 12, no. 1 p. 45, 2023.

[19] P. Joshi, M. Hasanuzzaman, and C. Thapa, "Enabling all in-edge deep learning: A literature review," *IEEE Access*, vol. 11, pp. 33431-3460, 2023.

[20] R. Ma and X. Chen, "Intelligent education evaluation mechanism on ideology and politics with 5G: PSO-driven edge computing approach," *Wireless Networks*, vol. 29, no. 3, pp. 685-696, 2023.

[21] S. Shen, "Metaverse-driven new energy of Chinese traditional culture education: Edge computing method," *Evolutionary Intelligence*, vol. 16, no. 1, 2022.

[22] B. Huang, Z. Ran, D. Yu, Y. Xiang, X. Shi, and Z. Li, et al., "Stateless Q-learning algorithm for service caching in resource constrained edge enviroment," *Journal of Cloud Computing*, vol. 12, no. 1, p. 132, 2023.

[23] L. Bao and P. Yu, "Evaluation method of online and offline hybrid teaching quality of physical education based on mobile edge computing," *Mobile Networks and Applications*, vol. 26, no. 5, pp. 2188-2198, 2021.

[24] H. Xu, Y. Sun, J. Gao, and J. Guo, "Intelligent edge content caching: A deep recurrent reinforcement learning method," *Peer-to-Peer Networking and Applications,* vol. 15, no. 6, pp. 2619-2632, 2022.

[25] L. Chunlin and J. Zhang, "Dynamic cooperative caching strategy for delay-sensitive applications in edge computing environment," *The Journal of Supercomputing*, vol. 76, no. 10, pp. 7594-7618, 2020.

[26] D. Wu, H. Xu, Z. Li, and R. Wang, "Video placement and delivery in edge caching networks: Analytical model and optimization scheme," *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3998-4013, 2021.

[27] W. Fan, J. Han, J. Chen, Y. A. Liu, and F. Wu, "Probabilistic computation offloading and data caching assisted by mobile-edge-computing-enabled base stations," *Annals of Telecommunications*, vol. 76, pp. 447-465, 2021.

[28] Y. X. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloding for mobile edge computing," *IEEE Access*, vol. 6, pp. 11365-11373, 2018.

[29] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, and Y. Zhang, et al., "LeaD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Transactions on Mobile Computing*, vol. 20, no. 8, pp. 2607-2623, 2021.

[30] L. Z. Cui, X. X. Su, Z. X. Ming, Z. Chen, S. Yang, and Y. Zhou, et al., "CREAT: Blockchain-assisted compression algorithm of federated learning for Content caching in edge computing*," IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14151-14161, 2022.

[31] M. Alian and G. Al-Naymat, "Questions clustering using canopy-K-means and hierarchical-K-means clustering," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3793-3802, 2022.

[32] M. Dong, L. Yao, X. Wang, X. Xu, and L. Zhu, "Adversarial dual autoencoders for trust-aware recom-

mendation," *Neural Computing and Applications*, vol. 35, pp. 13065-13075, 2023.

[33] K. Raja, B. Anbalagan, S. Anbalagan, K. Dev, and A. Ganapathisubramaniyan, "Popularity based content caching of YouTube data in cellular networks," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 37165-37182, 2022.

[34] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[35] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of Edge Computing systems," in *2017 Second Inter-national Conference on Fog and Mobile Edge Computing (FMEC)*, 2017.

[36] V. Padhye, K. Lakshmanan, and A. Chaturvedi, "Proximal policy optimization based hybrid recommender systems for large scale recommendations," *Multimedia Tools and Applications*, vol. 82, no.13, pp. 20079-20100.

AUTHOR

Yinan Song received her B.S. degree from Luoyang Normal University in 2010 and M.S. degree Gannan Normal University in 2014. She is currently working at Henan Women's Vocational College as a Lecture. Her main research interests include English Teaching, Real-Time Language Learning, Computer Applications, etc.