International Journal of Digital Crime and Forensics
Volume 15 • Issue 1

UAV Edge Caching Content Recommendation Algorithm Based on Graph Neural Network

Wei Wang, Hebei University of Engineering, China*
Longxing Xing, Hebei University of Engineering, China
Na Xu, Hebei University of Engineering, China
Jiatao Su, Hebei University of Engineering, China
Wenting Su, Hebei University of Engineering, China
Jiarong Cao, Hebei University of Engineering, China

ABSTRACT

When responding to emergencies such as sudden natural disasters, communication networks face challenges such as network traffic surge and complex geographic environments. Aiming at the problems of high transmission delay and insensitivity to user's preference in the current UAV edge caching strategy, this paper proposes a UAV caching content recommendation algorithm based on graph neural network. Firstly, the location of UAV is determined by clustering algorithm; secondly, the interest preferences of user nodes in the cluster are predicted by GCLRSAN model, and the UAV cache content is designed according to the result; finally, simulation experiments show that the model and algorithm proposed in this paper can effectively reduce the backhaul link overhead and outperform the comparison algorithms in the indexes such as accuracy rate, recall rate, cache hit rate, and transmission delay.

KEYWORDS

Attention Mechanism, Deep Learning, Edge Cache Network, Edge Computing, Emergency Communication, Graph Neural Network, Recommendation System, Session-Based Recommendation

INTRODUCTION

As the internet continues its relentless expansion, there has been an exponential surge in data traffic. To cater to users' communication demands, major telecommunications operators have deployed densely packed small cell stations. However, this has significantly burdened the backhaul links (X. Wang et al., 2014). In emergency scenarios such as natural disasters or other crises, certain base stations may

DOI: 10.4018/IJDCF.332774 *Corresponding Author

be damaged and complex geographic terrains complicate matters further, and the intricacies and vulnerabilities of communication systems are amplified. To address these challenges, the caching of popular content on UAVs integrated into the cellular network has emerged as a compelling research topic. This approach not only mitigates transmission latency but also alleviates the strain on backhaul links during peak hours (Li et al., 2019; S. Zhang et al., 2019).

Research by Navarro-Ortiz et al. (2020) forecasts a global increase in smartphone users from 6.3 billion to 12 billion between 2020 and 2030, accompanied by a 10 to 100-fold surge in global mobile communication traffic. This colossal surge in data traffic poses a substantial challenge to existing communication systems. In recent years, UAVs have found increasing applications in wireless communication. For instance, Chen et al. (2017) introduced a UAV deployment function with caching capabilities in a cloud access network, aiming to minimize UAV transmission distances. Liu et al. (2019) employed a genetic algorithm-based K-means (GA-K-means) method to partition users into cells and subsequently proposed a Q-learning-based deployment algorithm for UAVs. Park et al. (2019) optimized the placement of multiple UAVs in a base station-aided communication system, considering user demands to maximize service throughput. J. Yang et al. (2020) introduced a UAV collaboration scheme for caching in cognitive radio networks, enhancing CRN's transmission capacity while reducing redundant traffic loads. Zeng et al. (2022) presented a layered caching solution for different UAVs, caching specified layers of video based on Scalable Video Coding (SVC).

However, none of these solutions consider the user's preferences when requesting resources. Zhao et al. (2019) and T. Zhang et al. (2020) focused solely on caching popular content on UAVs. Traditional caching policies such as LFU, LRU, and FIFO, while effective in scenarios with consistent object sizes (Xu et al., 2018), struggle to adapt adequately to the wireless network environment due to a lack of consideration for factors like network topology and user preferences (Han et al., 2021).

On another front, the explosive growth of multimedia content has led to the pervasive problem of information overload. As one of the critical methods for alleviating this issue, recommendation systems provide users with services tailored to their interests. The evolution of recommendation system models can be broadly categorized into three phases: traditional shallow models, general neural network models, and graph neural network models. Early models relied on computing the similarity of user/item historical data directly in order to capture collaborative filtering (CF) effects (Koren et al., 2009; Su & Khoshgoftaar, 2009). However, these shallow models struggle to handle complex user behaviors or data inputs. With rapid advancements in deep learning research, neural network-based recommendation models emerged as an upgrade to shallow models. The Neural Collaborative Filtering (NCF) model, for instance, employs a Multi-Layer Perceptron (MLP) instead of the dot-product function in matrix factorization models (Kipf & Welling, 2016).

Nonetheless, both shallow and traditional deep models overlook structured information inherent in collected data. The rapid development of Graph Neural Networks (GNNs) provides an opportunity to address these issues (Z. Wang et al., 2023). Bruna et al. (2014) introduced the first Graph Convolutional Neural Network, albeit with limitations such as scalability issues and the risk of over-smoothing. Veličković et al. (2018) proposed Graph Attention Networks (GAT), employing attention mechanisms to weight and aggregate features from neighboring nodes. Jiang et al. (2021) and Song et al. (2019) integrated social information into embedded neighborhood matrices within the GNNs. Unlike the previous studies, their research doesn't assume that the social influence of friends is fixed and static, and it utilizes attention weights specific to users and context-aware attention weights to model interactions (Jiang et al., 2021b; Wu et al., 2019). Models like HGLR, CP-GNN, and GBK-GNN construct and propagate information within the graph structure to obtain user and item embeddings (Du et al., 2022; W. Yang et al., 2023).

However, due to the vast and sparse nature of the user-item graph, learning the entire graph structure through a graph neural network incurs high computational costs. As a result, sequence recommendation methods like FMLP-Rec, CL4SRec, and DGSR have begun using graph neural networks to model user interaction sequences. This reduces resource consumption while enhancing

recommendation performance (Xie et al., 2022; M. Zhang et al., 2022; Zhou et al., 2022). The successful application of deep learning in recommendation systems has opened up new possibilities for addressing issues in wireless networks, including content caching problems (Reiss-Mirzaei et al., 2023).

These studies and research endeavors prove that deep learning-based recommendation systems are much more effective at understanding user preferences when compared to traditional edge caching strategies. Within the recommendation system domain, graph neural networks and attention mechanisms hold great promise. By modeling user interaction sequences, they can effectively reduce computational complexity while simultaneously enhancing recommendation performance. However, in UAV caching within emergent scenarios, challenges such as data imbalance, high temporal requirements, and limited UAV cache space and computational resources persist (Cheng et al., 2018; M. Zhang, El-Hajjar, et al., 2022). These challenges render existing recommendation algorithms ill-suited to the cache requirements of emergent UAV scenarios.

This paper envisions solutions to these problems. Its main components are:

1. It proposes an innovative session recommendation model based on graph neural networks and low-rank decomposition self-attention networks. This model learns complex relationships and features between nodes, effectively reduces computational complexity, and better captures the topology of the graph through the low-rank decomposition attention network.
2. It designs a UAV caching content recommendation algorithm optimized for emergency scenarios. This algorithm can intelligently select and cache the data that best meets the user's personalized needs in order to improve the user experience and reduce the load of the backhaul link.
3. The edge caching algorithm and recommendation model proposed in this paper are applied to a simulation environment. Their effectiveness is demonstrated using several metrics such as recall, accuracy, and average transmission delay, which proves that they are innovative solutions to address the challenges in emergency communication scenarios.

SYSTEM MODEL AND PROBLEM STATEMENT

System Model

The UAV edge caching architecture for emergency scenarios is shown in Figure 1.

The edge computing architecture consists of base stations in the wireless access network and UAV cache units placed within the base stations. In this architecture, we consider a user set u consisting of u mobile users and a UAV set K consisting of k UAVs with caching capabilities. The number of video resources stored in the content server is denoted by n, and the video resource set is denoted by η.

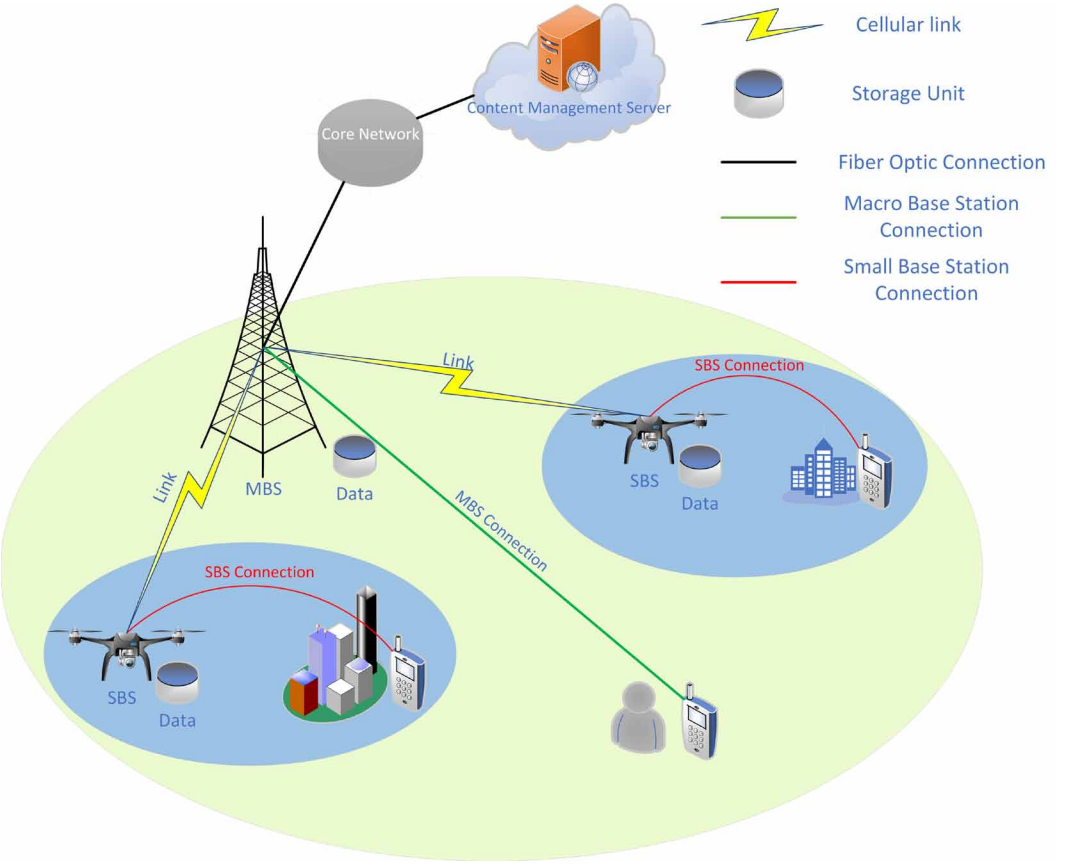
Due to the limited storage space of UAVs, we prioritize caching high-popularity and high-request probability video resources on edge devices. This helps to reduce user delay, decrease the overall system response time, and improve user experience. Specifically, when a mobile user requests data resources from the server, the following steps are executed:

1. We check whether the requested resource is cached in the UAV cache.
2. If it is, the UAV directly delivers the content to the requesting user, indicating a "cache hit."
3. Otherwise, the base station sends a request to the content server, and then delivers the results to the user, which is called a "cache miss." This results in longer information transmission delays.

Problem Statement

When networks are congested, users experience diminished average bandwidth, prolonged transmission latency, and heightened backhaul link loads. To address these challenges, content is cached within

Figure 1. UAV edge caching system model



unmanned aerial vehicles (UAVs) and deployed to areas of network congestion. This caching strategy serves the dual purpose of enhancing user experience while concurrently reducing the capacity demands on backhaul links. This paper professes that UAVs possess the capability to dynamically adjust cached resources based on user preferences within specific regions, thereby catering to users' content requests.

The optimization problem addressed in this study centers around the maximization of UAV cache hit rates. Given the finite capacity of UAV cache units, this optimization problem is subject to certain constraints. The variables at play encompass the positions of UAVs and the content they cache. It is imperative to emphasize that the user file request model plays a pivotal role in determining UAV cache hit rates. The probability matrix P of all user requests for a file is shown in Equation 1:

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{u1} & \cdots & p_{un} \end{bmatrix} \tag{1}$$

where p_{un} denotes the element in the $u\text{-}th$ row and $n\text{-}th$ column, indicating the probability that user u requests file n. According to Equation 1, the global prevalence is:

$$P_n = \sum_{u=1}^{U} \frac{P_{un}}{U} \tag{2}$$

The cache hit rate is defined as follows: A cache hit is considered to be received when user u is able to receive the requested content in a nearby UAV k at a certain moment t. The overall cache hit rate is shown in Equation 3:

$$h\left(F,L\right) = \sum_{u}\sum_{n} R_{un} \frac{F_{k(u)n}}{N_R} \tag{3}$$

where R_{un} denotes 1 if user u requested file n and 0 if not, and F_{un} denotes 1 if drone k cached file n and 0 if not. Further, $k_{(u)}\left(k_u \in K\right)$ denotes the number of drones that user u is connected to, and N_R is the number of times that the user has issued all requests.

To maximize the cache hit rate of UAVs, the optimization problem can be formulated as shown in Equation 4:

$$\begin{aligned} &max_{F,L} \sum_{u}\sum_{n} R_{un} \frac{F_{k(u)n}}{N_R} \\ &s.t. \sum_{n=1}^{N} F_{kn} \le c \end{aligned} \tag{4}$$

where c denotes the maximum number of cached files for the drone.

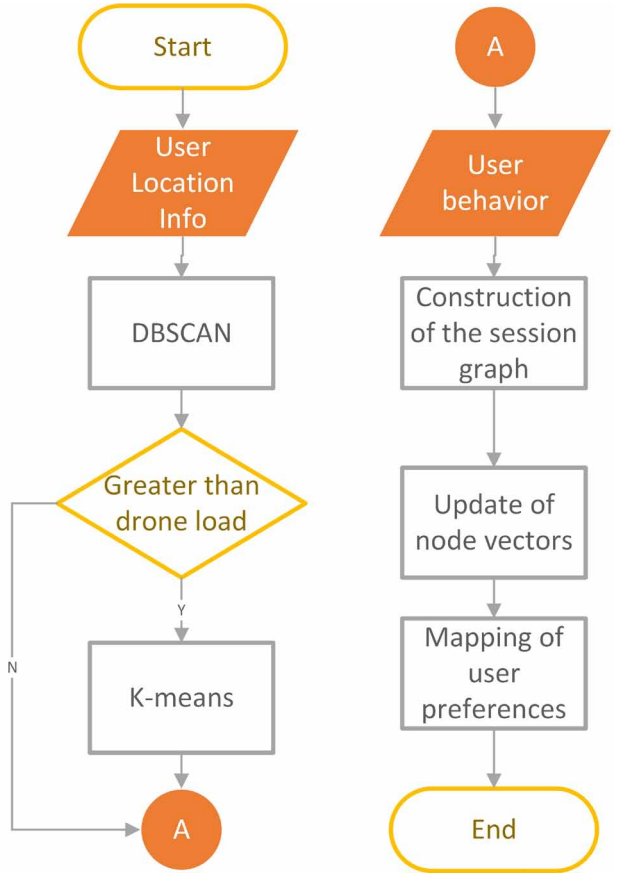
The problem studied in this paper is a multi-objective nonlinear planning problem, which is computationally too expensive for traditional solutions. Therefore, this paper proposes a graph neural network-based content recommendation algorithm for UAV edge caching. This first determines the deployment location of the UAV by analyzing the user's request location. Then, it predicts the content of the user's next request by using the GCLRSAN model and caches the content based on the prediction result and the location of the UAV. This method optimizes the caching strategy and improves overall system performance.

UAV EDGE CACHING CONTENT RECOMMENDATION ALGORITHM

Based on a Graph Neural Network

The overall algorithmic workflow of this study is presented in Figure 2. Initially, we apply clustering algorithms to group users into clusters, which enables a better understanding of their requirements. Next, we determine the deployment locations of drones based on the nearest drone-user connection strategy, and we establish a communication channel between users and drones to facilitate data transmission. Following this, we employ the GCLRSAN model, a graph neural network combined with a low-rank decomposition self-attention network, to analyze the preferences of local user clusters and provide personalized data recommendations for different users. During the recommendation process, we weight the results by considering the timestamp information of the data to ensure timeliness and accuracy. Finally, we recommend that the content of files be cached on drones, improving cache hit rates and mitigating network congestion caused by damage to ground infrastructure.

Figure 2. Algorithm flowchart



User Location Information Processing

The text employs DBSCAN and k-means clustering algorithms to process user location information. DBSCAN is a density-based clustering algorithm that can handle clusters of any shape without requiring the number of clusters to be specified in advance. However, it is sensitive to the estimation of data density and the choice of parameters. On the other hand, k-means is a distance-based clustering algorithm that is simple and easy to implement, but it requires the number of clusters to be predetermined and it is sensitive to the choice of initial centroids. To overcome these limitations, this paper adopts a hybrid clustering algorithm that combines DBSCAN and k-means. Specifically, the algorithm first employs DBSCAN to divide sample points into several clusters and then applies k-means to clusters with a number of nodes exceeding the maximum connection capacity of drones. For sample points not assigned to any clusters, k-means is used to assign them to the nearest cluster. By integrating DBSCAN and k-means algorithms, more accurate and stable clustering results can be obtained.

The clustering process is shown in Algorithm 1:

```
Algorithm 1
Hybrid Clustering Algorithm
Input: D : User Location Information
       ε : Radius parameters
       MinPts : Domain Density Threshold
```

k : Number of clusters
N : Maximum number of drones connected
L : Effective communication distance of UAV
Output: Clustering results
$DBSCAN\left(D,\varepsilon,MinPts\right)$
Cluster set = group the clustering results by cluster()
if Number of nodes in the cluster $> N$
for each cluster *in* the cluster set *do*
Clustering result $= k-means\left(cluster,k\right)$
Update the cluster set (clustering results)
Clusterless sample points = find the sample points that are not classified into any cluster()
for each unclustered sample point *in* unclustered sample points *do*
Nearest Cluster = find the cluster closest to the cluster-free sample point()
if the distance of the sample point from the nearest cluster centroid is less than the UAV communication distance L
Assign cluster-free sample points to the nearest cluster()
else Consider this node as a noise point
Output Cluster set

The proposed hybrid clustering algorithm offers several advantages, including: its ability to handle clusters of arbitrary shapes; more accurate handling of noise points; its suitability for high-dimensional data; and its consideration of the communication radius of drones and the maximum number of connections between users and drones. This algorithm is well suited for scenarios that require efficient and accurate clustering of data, such as item clustering in user preference analysis and recommendation systems.

GCLRSAN

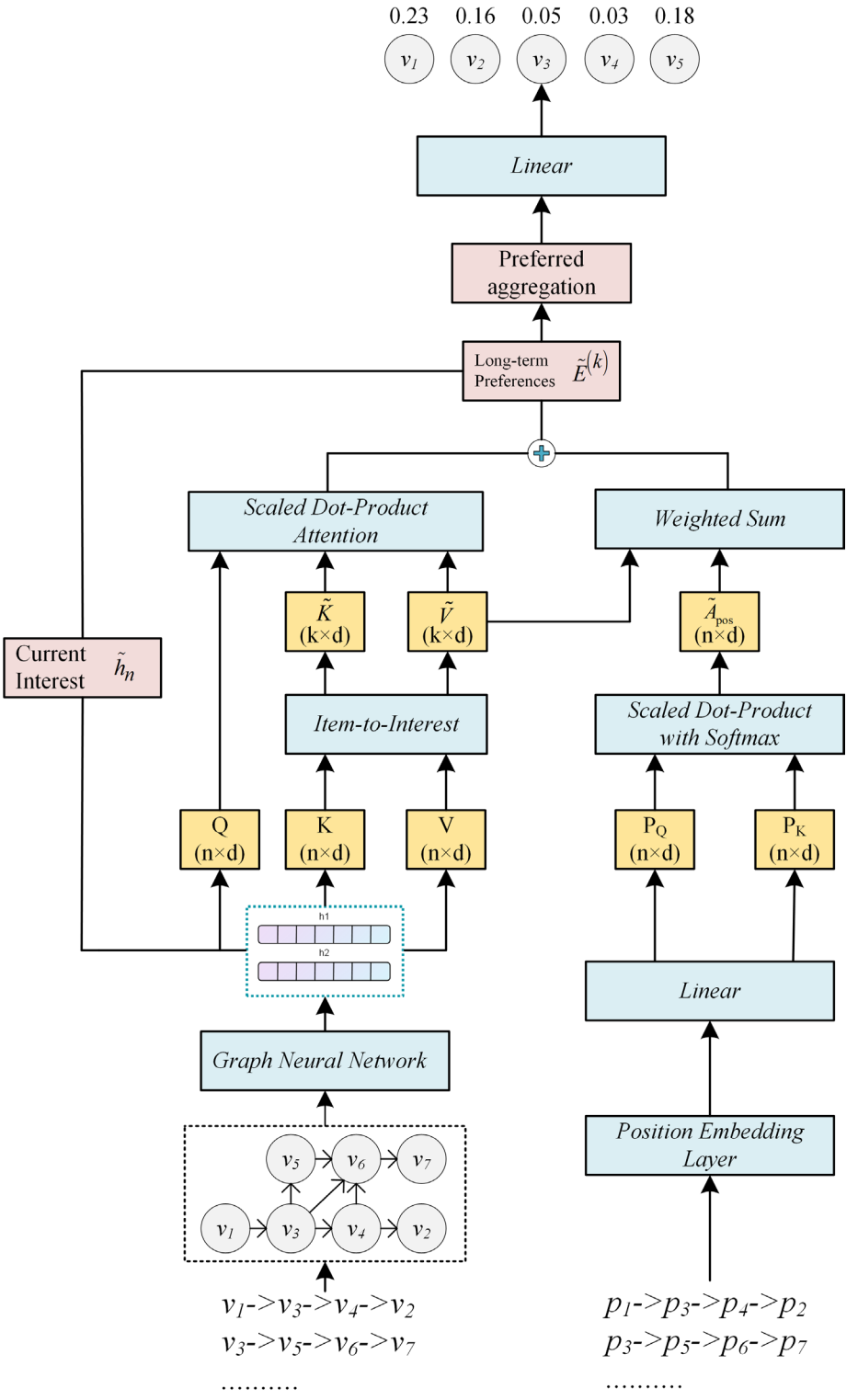
We propose the GCLRSAN model, a recommendation model based on graph neural networks and low-rank decomposed self-attention mechanisms. The aim of this model is to predict the content of the user's next file, and to use the prediction results for recommending the cache content carried by the UAVs in order to optimize content delivery efficiency in edge computing. The model's structure is shown in Figure 3.

Dynamic Graph Structure

Each session sequence $S=\left\{s_1,s_2,...,s_n\right\}$ can be represented as a directed graph $G_S=\left(V_s,E_s\right)$. Each item s_i in session s is treated as a node, and if a user clicks $v_{s,i-1}$ and then $v_{s,i}$ in session s, we consider $\left(s_{i-1},s_i\right)$ as an edge in the directed graph. The information propagation between different nodes is expressed in the form of Equation 5:

$$a_t=Concat\begin{pmatrix}\left(M_t^I\left[s_1,\ldots,s_n\right]W_a^I+b^I\right),\\ \left(M_t^O\left[s_1,\ldots,s_n\right]W_a^O+b^O\right)\end{pmatrix} \tag{5}$$

Figure 3. GCLRSAN model structure



The parameter matrices, denoted as W_a^I, W_a^O, belong to $R^{d\times d}$. The bias vectors are represented by b^I, b^O. Additionally, M_t^I, M_t^O both in $R^{1\times n}$ refer to the $t\text{-}th$ row of the node's in-degree matrix and out-degree matrix, respectively. a_t denotes the information of the node s_t aggregated neighborhood contexts. After that, a_t and s_{t-1} are input into the GNN network, and finally, h_t is obtained as shown in Equation 6:

$$\begin{aligned} z_t &= \sigma\left(W_z a_t + P_z s_{t-1}\right), \\ r_t &= \sigma\left(W_r a_t + P_r s_{t-1}\right), \\ \tilde{h}_t &= \tanh\left(W_h a_t + P_h\left(r_t \odot s_{t-1}\right)\right) \\ h_t &= \left(1 - z_t\right) \odot s_{t-1} + z_t \odot \tilde{h}_t \end{aligned} \tag{6}$$

where $W_z, W_r, W_h \in R^{2d\times d}, P_z, P_r, P_h \in R^{d\times d}$ are learnable parameters, $\sigma\left(\bullet\right)$ denotes the sigmoid function, and $\odot$ denotes element-by-element multiplication. z_t, r_t denote the update gate and reset gate, respectively, used to determine the retention and discarding of information.

Attention Layers

The self-attention mechanism is a technique that can capture the relationship between different input sequences. Due to the limited computing power of UAVs, this paper adopts a low-rank decomposition self-attention mechanism to generate context representations. This mechanism projects the items into k latent interests and integrates them with the context through interactions with these interests. This approach reduces the complexity from $O\left(n^2\right)$ to $O\left(nk\right)$, making the model more efficient.

Assuming that the user's historical items can be classified into no more than $k\left(k \ll n\right)$ potential interests, a learnable projection function $f : R^{n\times d} \to R^{k\times d}$ can be designed to summarize the historical items into latent interests. Next, the embedding sequence is transformed into matrices Q, K, V using linear projection matrices $W_Q, W_K, W_V \in R^{d\times d}$. They are input into the low-rank decomposed self-attention network. Through the item-to-interest aggregation function f, the original $K, V\left(n\times d\right)$ in the normal multihead attention mechanism are mapped to $\tilde{K}, \tilde{V}\left(k\times d\right)$, and then they are input into the self-attention layer, which can better capture global session preferences. The specific process is shown in Equation 7:

$$\begin{aligned} \tilde{F} &= softmax\left(\frac{Q\cdot \tilde{K}^T}{\sqrt{d}}\right)\tilde{V} \\ &= softmax\left(\frac{Q\cdot f\left(K\right)^T}{\sqrt{d}}\right) f\left(V\right) \end{aligned} \tag{7}$$

For the location embedding P, this paper uses decoupled location encoding to model the sequential relationship of items in the sequence; see Equation 8:

$$\tilde{E} = \tilde{F} + \tilde{A}_{pos}\cdot HW_V \tag{8}$$

where $\tilde{F}$ comes from Equation 7 and $U_Q, U_K \epsilon R^{d\times d}$, $\tilde{A}_{pos} = softmax\left(\frac{P_Q \cdot \left(P_K\right)^T}{\sqrt{d/h}}\right), P_Q = PU_Q, P_K = PU_K$ is the learnable matrix. According to the above equation, $\tilde{F}, \tilde{A}_{pos}$ is computed independently. $\tilde{A}_{pos}$, as an independent input, is computed only once and shared among all users in a joint batch, which greatly reduces the burden on the UAV computational unit. The above self-attentive mechanism is simplified and defined as:

$$\tilde{E} = SAN\left(H\right) \tag{9}$$

where H is the item embedding matrix. Different self-attentive layers can capture different types of features. The first layer is defined as $\tilde{E}^{(1)} = \tilde{E}$, and then the $k\text{-}th$ layer self-attention is defined as $\tilde{E}^{(k)} = SAN\left(\tilde{E}^{(k-1)}\right)$. $\tilde{E}^{(k)}$ is the ultimate output of the multilayer self-attentive network.

Prediction Layer

After undergoing multiple layers of self-attention processing, a long-term self-attention representation $\tilde{E}^{(k)}$ is obtained. In emergency situations, the UAV needs to quickly access the latest data; thus, the UAV caching system needs to be very timely. To better predict the user's next click target, this paper combines the user's long-term demand for disaster relief items with their current needs and uses this combined embedding as the embedding representation for the session. The last dimension of $\tilde{E}^{(k)}$ is taken as the global embedding representation, while the set of the user's last session clicks on disaster relief items, $\tilde{h}_n$, is taken as the local embedding vector. The final session embedding is thus:

$$\tilde{S}_f = \omega \tilde{E}_n^{(k)} + \left(1-\omega\right)\tilde{h}_n \tag{10}$$

Finally, the model is trained by minimizing the objective function:

$$\vartheta = -\sum_{i=1}^{n} y_i \log\left(\hat{y}_i\right) + \left(1-y_i\right)\log\left(1-\hat{y}_i\right) + \alpha\left\|\theta\right\|^2 \tag{11}$$

where y denotes the one-hot encoding of the true value of the item, $\hat{y}_i$ denotes the probability that a given item $v_i \epsilon V$ is clicked, and θ is the set of all learnable parameters.

EXPERIMENTS AND ANALYSIS

Experimental Dataset

The dataset used in this study consists of:

- 100,000 data ratings from 1,000 users, focusing on 1,600 disaster relief items. The ratings range from 1 to 5, where a rating of 3 indicates that the user has no particular preference or dislike for the item, while ratings of 1 and 5 indicate strong likes or dislikes;
- longitude and latitude information;
- and rating timestamps.

The detailed structure of the dataset is shown in Table 1.

Evaluation Metrics

In this study, we employed multiple experimental evaluation metrics to assess the performance of our proposed user preference-based edge caching recommendation system. These metrics aimed to evaluate the recommendation system from different perspectives to comprehensively understand its performance. The main evaluation metrics used include: $precision@N$, $recall@N$, $NDCG@N$, $Mrr@N$, hit rate, and average transmission delay. The description of these metrics is shown in Table 2.

The variables used in the evaluation metric are defined as follows:

- TP represents the number of recommended items that users are truly interested in,
- FP represents the number of recommended items that users are not interested in,

Table 1. Dataset structure

User ID	Item ID	Rating	Timestamp	Latitude	Longitude
290	1016	4	879373848	39.730 469°	-104.999 073°
296	357	5	887646170	39.726 924°	-105.000 121°
69	237	2	889241426	39.751 300°	-105.000 017°

Table 2. System evaluation indicators

Metrics	Description	Formula
Precision	The proportion of correct recommendations out of the total recommendations can indicate the recommendation's effectiveness	$\frac{TP}{\left(TP+FP\right)}$
Recall	The proportion of correct recommendations to the user's relevant items can be used to evaluate the recommendation's effectiveness	$\frac{TP}{\left(TP+FN\right)}$
NDCG	Measuring the ranking quality of the recommendation list	$\frac{DCG}{IDCG@N}$
MRR	For each user, their items of interest are sorted from smallest to largest by the position in which they appear in the recommendation results, and the average is obtained by taking the reciprocal	$MRR=\frac{1}{\left\|U\right\|}\sum_{u\in U}\frac{1}{rank_u}$
Cache hit rate	The number of items of interest to users among the first N recommended items	$\begin{cases}1, & if \ \ TP>0\\ 0, & else\end{cases}$
Average transmission delay	The ratio of the file to be transferred to the maximum transfer rate of the channel	$\frac{L}{B\times\log_2\left(1+\frac{s}{n}\right)}$

- FN represents the number of items that users are interested in but were not recommended,
- DCG denotes the discounted cumulative gain, and
- $IDCG$ denotes the ideal discounted cumulative gain, which is calculated as follows:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2\left(i+1\right)},\ IDCG@k = \sum_{i=1}^{min\left(k,\left|R\right|\right)} \frac{2^{rel_i} - 1}{\log_2\left(i+1\right)} \tag{12}$$

Here, k denotes the length of the recommended list, rel_i denotes the relevance of the $i\text{-}th$ item in the recommended list, $\left|R\right|$ denotes the total number of items of interest to users in the test set, $\left|U\right|$ is the number of users, $rank_u$ is the reciprocal of the ranking position of user u in the recommended list of the first item that is of interest to the user, L is the size of the file to be transmitted, B is the bandwidth, S is the signal power, and N is the noise power.

Baselines and Experimental Setup

In this study, we employed several fundamental algorithms:

a) SRGNN (S. Wu et al., 2019) utilized user historical behavior sequences as input and modeled them with graph neural networks to learn user interest preferences and generate recommendation results.
b) GCSAN (C. Xu et al., 2019) integrated self-attention mechanisms into graph convolutional neural networks, which can help the model focus more on important features and reduce noise interference, thus improving the recommendation performance of the model.
c) LightSANs (Fan et al., 2021) mapped user interests to a small number of constant latent interests and used the interaction between items and interests to generate context-aware representations.
d) TAGNN (Yu et al., 2020) introduced a novel target-aware GNN model specifically designed for session-based recommendation.

In this study, we employed Python 3.7, PyTorch 1.13.1, and CUDA 11.6 for experimentation. All algorithms were implemented using Python and third-party libraries such as Numpy and Scipy. We utilized precision, recall, and average transmission delay to assess the performance of the algorithm. The hyperparameters of GCLRSAN were set as shown in Table 3. The model's default learning epochs were set to 300, and the training was terminated if the model's performance on the validation set continuously decreased for five epochs. The embedding dimensions, learning rates, and batch sizes of the benchmark methods were kept the same as the model presented in this study, while other hyperparameters were set to their default values in the original papers or codes. For the top-N recommendation task, we set N to {5, 10, 15, 20, 25}. The objective of the experiment was to recommend the top N items for each user in the testing set, and we utilized $precision@N$, $recall@N$, $NDCG@N$, $Mrr@N$, and average transmission delay to evaluate the performance of each model.

Model Analysis and Discussion

Recommended Performance

We investigate how the recommended number of recommendations affects the evaluation results for the project recommendation task. For a more comprehensive comparison of the performance

Table 3. Hyperparameter settings of the model

Hyperparameter Name	Parameter Value	Hyperparameter Name	Parameter Value
Embedding Dimension	64	Train Batch Size	2048
Weights	0.8	Number of Attention Layers	2
Learning Rate	0.01	Dropout	0.2

differences between GCLRSAN and the benchmark algorithm, the number of recommendations N is set from 5 to 25 with an interval of 5 in order to carefully analyze and compare their metric variations.

RECOMMENDATION ALGORITHM PERFORMANCE ANALYSIS

The experimental results of the algorithm and the benchmark algorithm are shown in Figures 4-7.

As shown in Figures 4-7, when recommending five items, the GCLRSAN model outperforms SRGNN, GCSAN, LightSANs, and TAGNN in terms of recall rate, with improvements of 40.66%, 31.06%, 16.98%, and 26.73%, respectively. Similarly, the MRR and NDCG metrics also display enhancements of 34.55%, 16.72%, 8.50%, and 32.14%, and 36.78%, 22.05%, 11.74%, and 29.35%, respectively, when compared to the other algorithms. It is noteworthy that both the GCLRSAN and the baseline algorithms exhibit a direct correlation between $recall@N$, $Mrr@N$, $NDCG@N$, and the recommended item quantity. As the number of top-N items increases, more items are

Figure 4. Recall metric

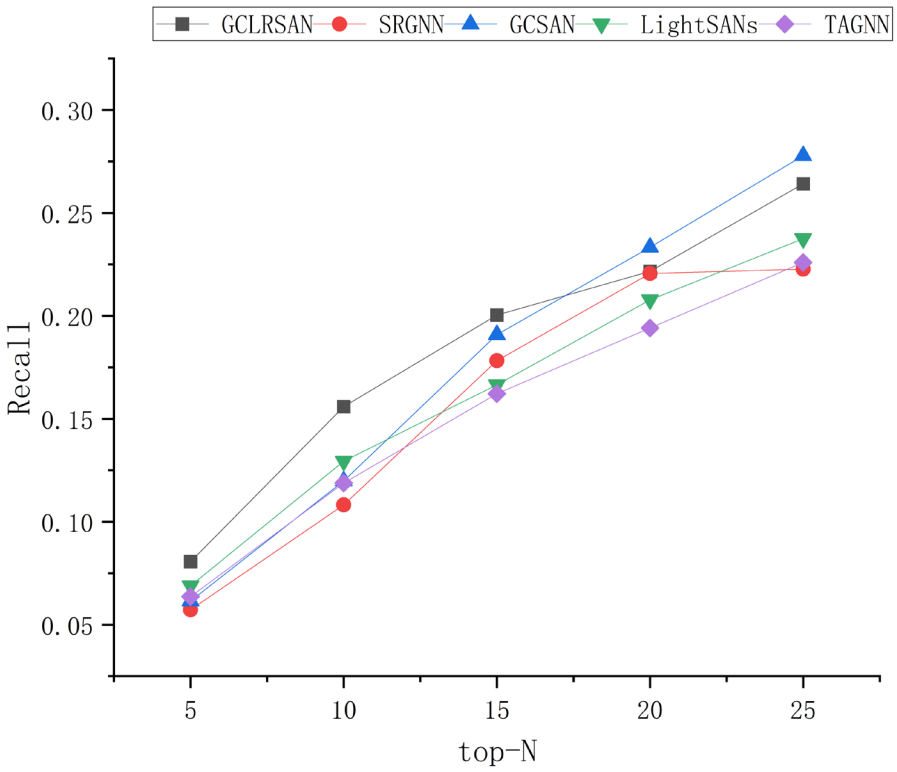
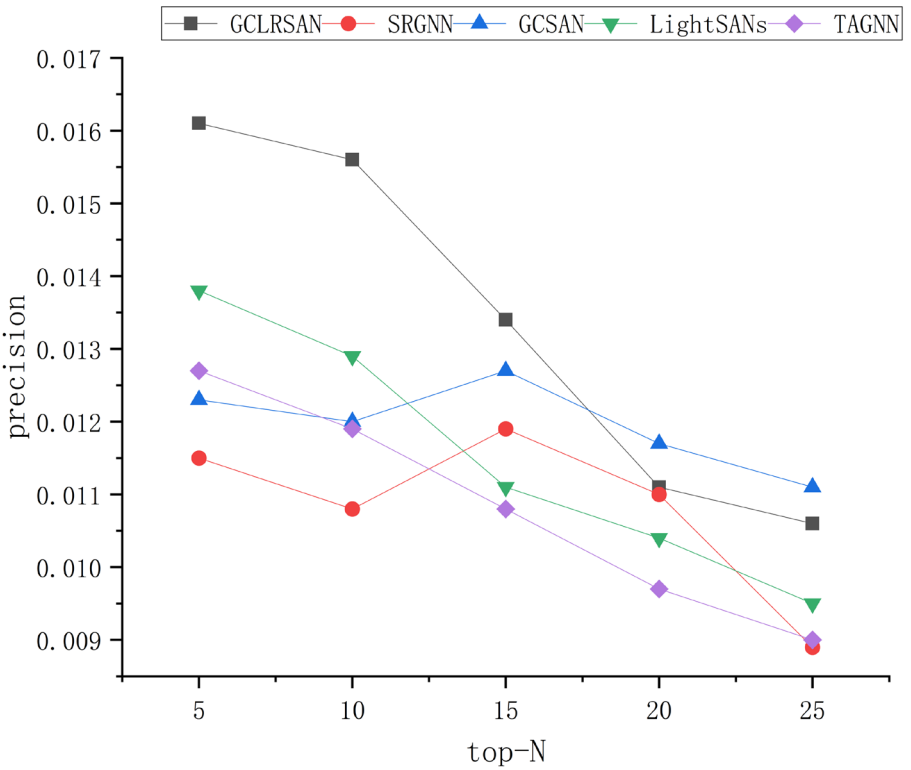


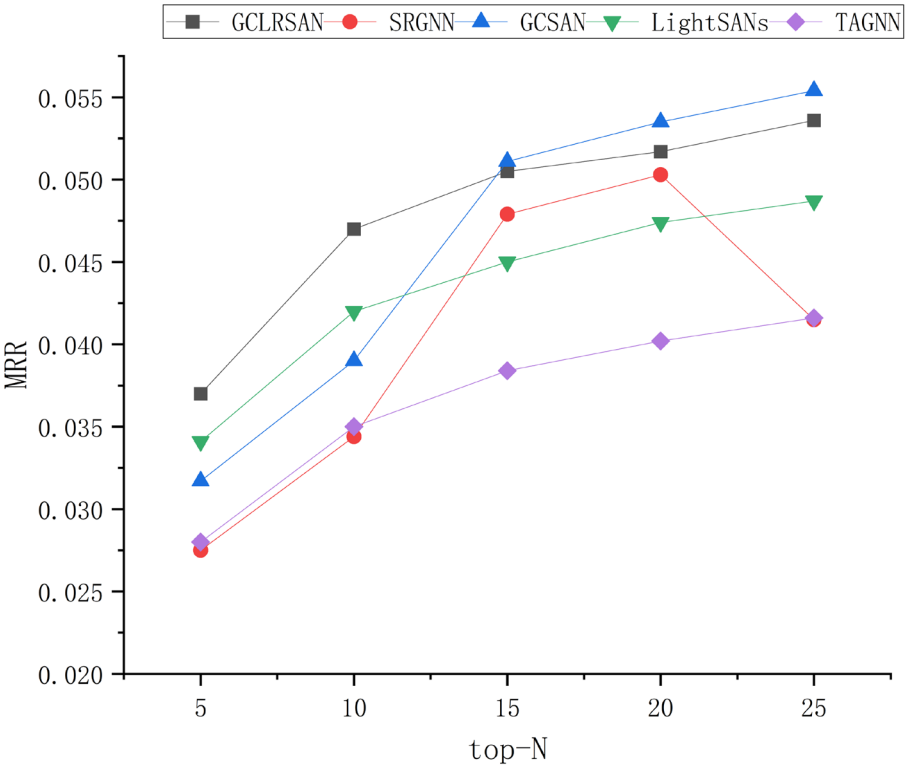
Figure 5. Precision metric



recommended to users, and as a result, the number of actually recommended items is also augmented. Therefore, if some items that truly interest users are not being recommended, increasing the top-N quantity would offer a higher chance of them being recommended to users, leading to improvements in these metrics.

The GCLRSAN model also outperforms SRGNN, GCSAN, LightSANs, and TAGNN in precision, exhibiting improvements of 40.00%, 30.89%, 16.67%, and 26.77%, respectively. However, all the algorithms show an inverse relationship between $precision@N$ and the recommended item quantity. As the number of recommended items increases, the recommendation system may face the "long-tail problem," where the recommendations gradually become more focused on items that are obscure and less popular. These items may increase diversity, but their recommendation accuracy is often lower than that of popular items. When the number of recommended items is set to five, the GCLRSAN model performs the best, followed by LightSANs, TAGNN, GCSAN, and SRGNN. While SRGNN employs attention mechanisms, it has difficulty handling sequence data, leading to suboptimal performance. On the other hand, TAGNN and GCSAN use self-attention mechanisms and attention network models specifically designed for target items, enabling them to handle sequence data more effectively, and therefore exhibiting higher recall rates than SRGNN. However, their use of implicit positional encoding results in inaccurate modeling of the order relationship between items, leading to inferior performance compared to LightSANs. While LightSANs perform well, they lack handling of data sparsity, resulting in slightly inferior performance compared to GCLRSAN. As shown in the figure, the GCLRSAN model performs the best when the number of top-N items is relatively small, and its performance gradually approaches the mainstream level as the number of recommended items increases.

Figure 6. MRR metric



The experimental results lead to the conclusion that the GCLRSAN performs well in all aspects, indicating that the model is able to better capture user preferences and provide more comprehensive recommendation candidates. Notably, a significant improvement in performance was observed in the GCLRSAN model compared to the benchmark algorithm when recommending 5 and 10 items. As the cache space of the drone is limited, it is important to consider the cache space constraint during recommendation, and it is not advisable to recommend too many items for each user, as this may cause cache overflow. With relevance to this, when the number of recommended items is low, the performance improvement of the GCLRSAN model is particularly evident, which is in line with the research purpose of this paper. In summary, the GCLRSAN model has potential application prospects in recommendation systems and can provide better recommendation services for application scenarios such as drones.

COMPARISON OF THE MODEL ITERATION PROCESS

To assess theperformance of different models on low-computational platforms, this experiment was conducted using a single Intel Xenon Gold 5218R CPU@2.10 As shown in Figure 8, the training time for each epoch indicates that SRGNN has the shortest training time, while its attention mechanism is primarily used to model the relationships between nodes, resulting in significant improvements in capturing both the local structures and global features of graph data. In contrast, the self-attention mechanism used in this study can learn the correlations and importance between each sequence element, allowing for better capturing of long-term dependencies. The experimental results in Figures 4-7 show that our model outperforms SRGNN in terms of recall, accuracy, and other metrics.

Figure 7. NDCG metric

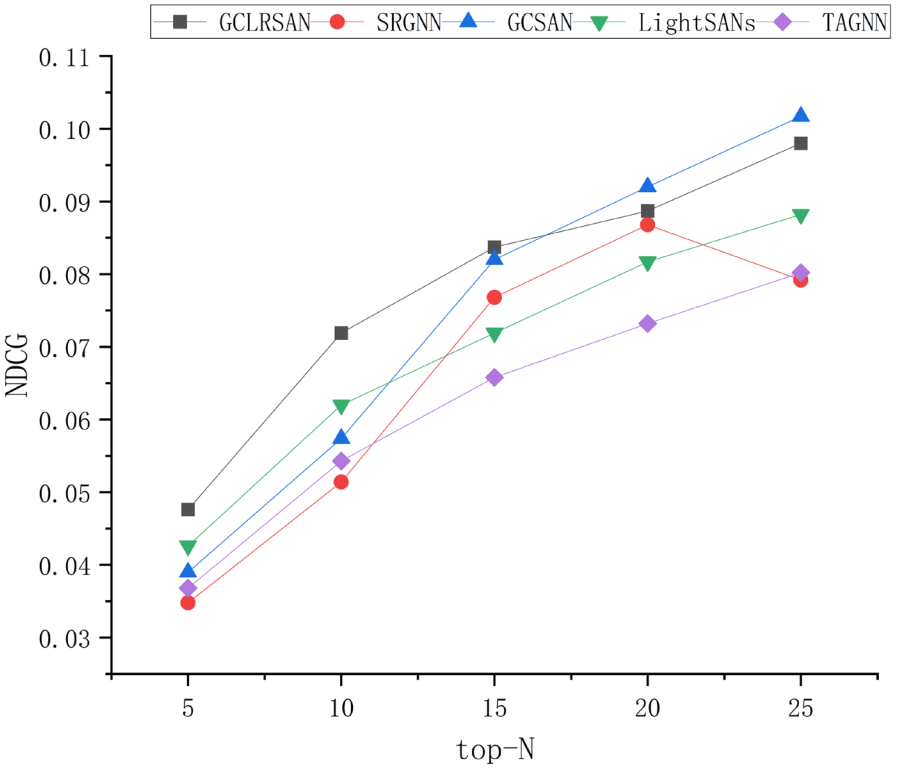
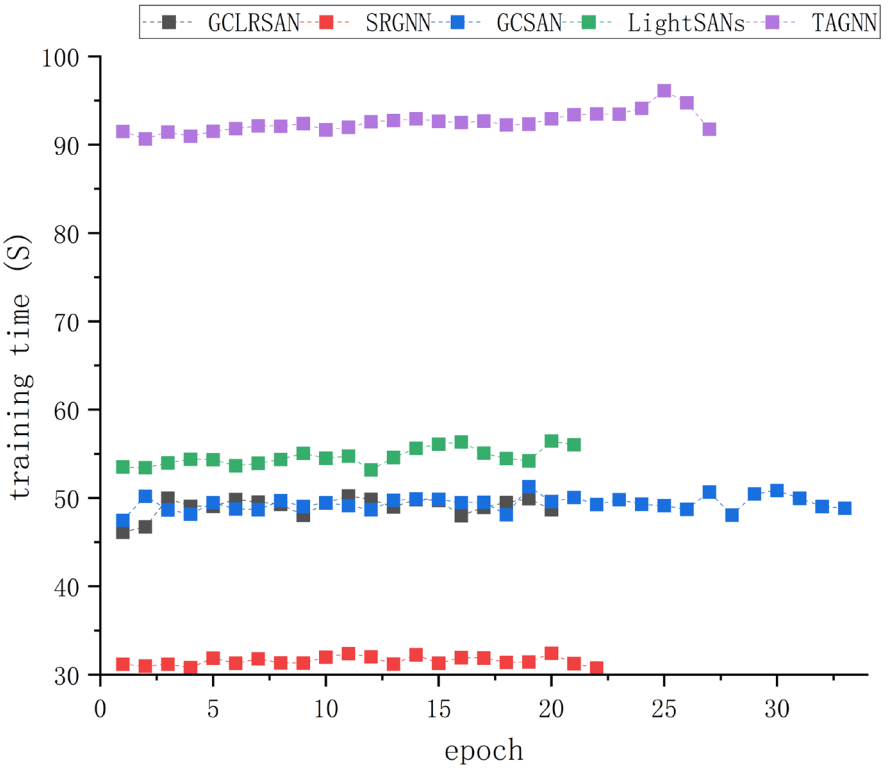


Figure 9 demonstrates that GCLRSAN achieves the fastest convergence rate among all models, with the training time order being SRGNN, GCLRSAN, LightSANs, GCSAN, and TAGNN. Despite SRGNN's faster speed, self-attention mechanisms remain an effective method for sequence data processing. By analyzing the model complexity, it can be seen that the graph neural network overhead of GCLRSAN is approximately $O\left(n^2d + nd^2\right)$, while the attention mechanism layer overhead is approximately $O\left(\left|V\right|k + cn\right)$, where n is the session length, d denotes the vector embedding dimension, $\left|V\right|$ denotes the total number of items, k is the potential user interest, and c is the model training period.

For GCSAN, which also uses a graph neural network, the time complexity of its attention mechanism section is approximately $O\left(\left|V\right|^2\right)$. GCLRSAN divides a user's history projects into no more than k categories of potential interests, using a mapping function to linearly aggregate the latent vectors from n to k dimensions, which results in a more compact interest distribution. This directly reduces the calculation overhead from $O\left(\left|V\right|^2\right)$ to $O\left(\left|V\right|k\right)$, where k is much smaller than $\left|V\right|$. Further, this greatly reduces the model complexity and effectively improves model performance, making it more suitable for use on low-computational platforms such as UAVs.

Figure 8. Epoch training time



Simulation Experiments in Application Scenarios

Cache Hit Rate

The generic parameter settings used for the cache hit rate simulation are shown in Table 4.

This article proposes an algorithm that combines cluster analysis with local popularity to optimize content caching for UAVs. The UAV's position is determined using a hybrid clustering algorithm, and the content cache is obtained using the GCLRSAN model.

In Algorithm A, the UAV's position is obtained using the hybrid clustering algorithm, and the content cache is based on the global popularity of the user group, which is the average preference of all users in the simulation area.

Algorithm B also uses the hybrid clustering algorithm to determine the UAV's position, but the caching strategy is based on random caching of files in the UAV's cache unit.

In Algorithm C, the UAV's position is uniformly distributed, and the caching strategy is based on the overall preference of users.

The cache hit rate, as influenced by changes in the storage capacity of the drones, is presented in Figure 10. The criteria for cache hits in this study require the user to be within the coverage area of the drone and for the requested content to be cached within the drone. The results indicate that as the number of cached files in the drone increases, both the proposed algorithm and the baseline algorithm exhibit a gradual increase in cache hit rates. The proposed algorithm leverages a hybrid clustering model to group users and it utilizes the GCLRSAN model to analyze the preferences of

Figure 9. Number of training iterations and average training time

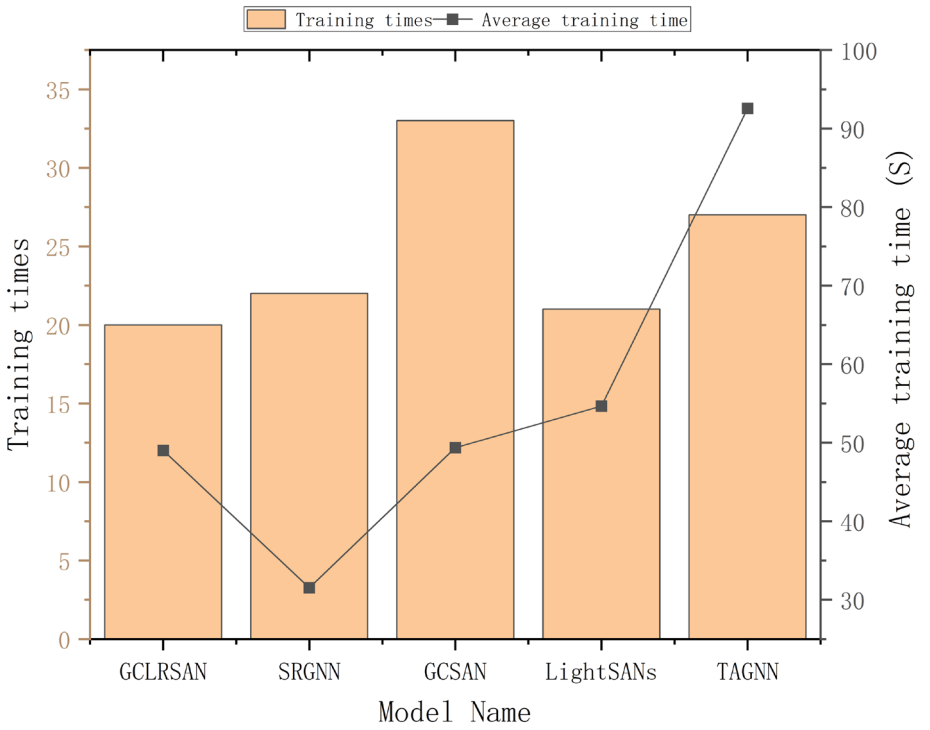


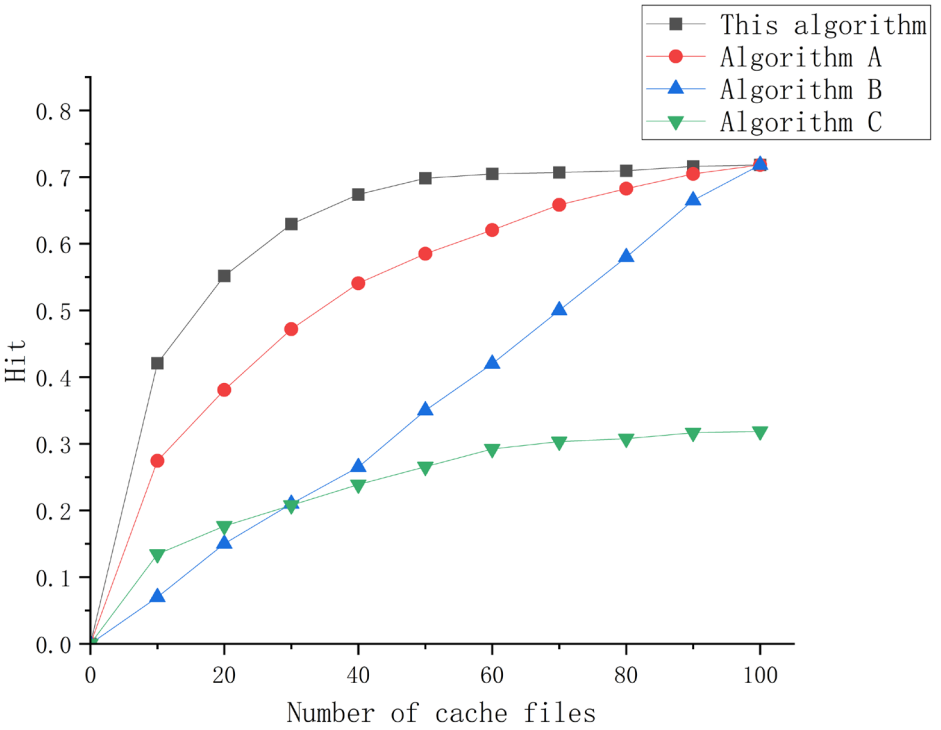
Table 4. Same simulation parameter settings

Parameter Name	Parameter Value	Parameter Name	Parameter Value
Users (pcs)	30	Number of documents (pcs)	100
Number of UAVs (frames)	6	Geographical area (m)	500*500
Height of base station (m)	50	Drone flight altitude (m)	20
Clustering distance threshold (m)	200	Cluster minimum number of users (pcs)	3

clustered user groups, which is more aligned with user behavior. This accounts for the higher cache hit rate of the proposed algorithm compared to the baseline algorithm.

When the number of cached files in the drone reaches 100, i.e., all requested files are backed up in the drone, both the proposed algorithm and Algorithms A and B adopt the same hybrid clustering method and exhibit the same cache hit rate. In contrast, Algorithm C, which distributes drone locations uniformly, performs suboptimally with 100 cached files. Through comparative experiments, the superiority of employing the hybrid clustering algorithm and the nearest drone-user link strategy for allocating drone locations is highlighted. Furthermore, the proposed algorithm achieves higher accuracy in predicting the preferences of local user groups using the GCLRSAN model, which leads to a significant improvement in the cache hit rate compared to algorithms that randomly cache files without predicting user preferences.

Figure 10. Cache size and hit rate



AVERAGE TRANSMISSION DELAY

In this study, the relationship between the number of cached files and the average transmission delay was investigated, and the results are presented in Figure 11. Two modes of user file requests were considered: Mode 1 involves a macro base station (MBS) connection, where user requests are transmitted through MBS to the user terminal with the content servers providing the transmission service. Mode 2 involves a small base station (SBS) connection, where user requests are directly transmitted from the SBS to the user terminal if the requested files are cached in the SBS within the user's coverage area. If there is no SBS available within the user's coverage area or if the requested files are not cached in the storage of the SBS, then file transmission is carried out via MBS connection. The simulation parameters for the average transmission delay are provided in Table 5.

Based on the data presented in Figure 11, it can be observed that the average transmission delay of users gradually decreases as the number of cached files in the drones increases. This trend is

Table 5. Average transmission delay experimental simulation parameters

Parameter Name	Parameter Value	Parameter Name	Parameter Value
MBS connection bandwidth (MHZ)	10	SBS connection bandwidth (MHZ)	100
Base station transmitting power (dBM)	30	UAV transmitting power (dBM)	20
Gaussian white noise power spectral density (dBM/HZ)	-174	Individual file size (MB)	100

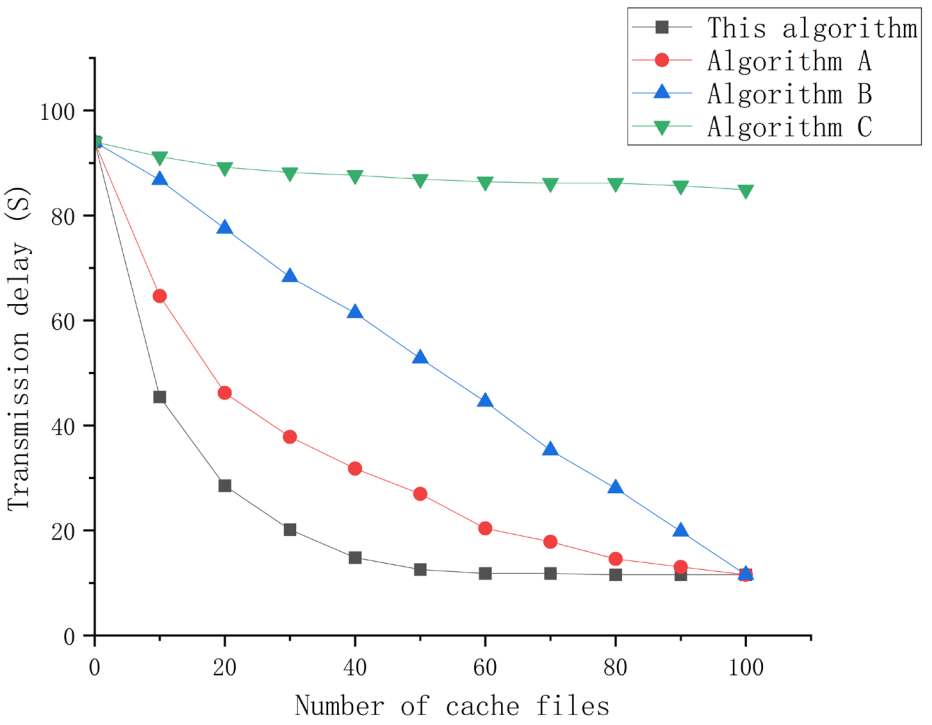
particularly pronounced for the algorithm proposed in this study. Our proposed algorithm first uses DBSCAN clustering to consider the communication radius of the drones, and it can handle clusters of any shape. However, since a cluster separated by DBSCAN may contain too many users, k-means clustering is used again with the maximum number of connected users as the threshold for further partitioning these clusters. Subsequently, we designed the GCLRSAN model to recommend drone cache resources.

Experimental results demonstrate that our proposed algorithm performs well in emergency scenarios for edge caching using drones. This research can help to ensure reliable communication services during emergency situations. In the event of natural disasters or other urgent crises, the UAV cache content recommendation system presented in this paper can substantially enhance communication capabilities. It can aid rescue teams in carrying out their operations more effectively, thereby reducing disaster-related losses and casualties. Furthermore, this study has the potential to positively impact the economic feasibility of communication infrastructure. By alleviating the burden on traditional base stations, this system holds the promise of lowering operational and maintenance costs for service providers. This, in turn, allows operators to allocate resources more efficiently, resulting in cost savings.

CONCLUSION AND FUTURE WORK

This paper proposes and validates the GCLRSAN model, which performs well in UAV caching content recommendation. Through experiments and analysis, the GCLRSAN model is shown to have great potential in recommender systems since it can effectively meet users' personalized needs and provide more accurate content recommendations. Especially when the number of recommended

Figure 11. Cache size and hit rate



items is small, the GCLRSAN model shows significant performance improvement over traditional benchmark algorithms, and this is especially important in emergency scenarios when UAV cache space is limited.

The caching algorithm in this paper provides a solution to the problems of low UAV load and sparse data in emergency scenarios, it improves the cache hit rate in the edge caching system, and it also reduces the average transmission latency of user requested resources. This research also has the potential to improve the performance of UAV applications, especially in disaster response and emergency situations. By managing and delivering information more efficiently, the potency of UAVs in search and rescue, surveillance, and communication can be increased.

Although the research in this paper addresses some of the shortcomings of existing UAV edge caching, there are other problems to be tackled in this field, which opens up avenues for further research. In future work, we will further optimize the caching strategy by considering more factors such as timeliness and users' mobile characteristics in order to improve the system performance. We also plan to apply the GCLRSAN model to real UAV communication systems and study its feasibility and benefits in commercial scenarios. In addition to this, deploying the research results of this paper into a UAV communication network involves a number of practical operations in real-world applications. This includes choosing where to deploy the UAVs, how to manage and update the cached content, and how best to coordinate with the UAV operations team in order to ensure optimal performance. We plan to explore these practical operations in future research to ensure that our algorithms and models achieve optimal results in real-world deployments.

AUTHOR NOTE

The research presented here is supported by the National Natural Science Foundation of China (No. 61802107), Technology Research Project of Higher Education in Hebei Province (No. ZD2020171).

REFERENCES

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). *Spectral networks and locally connected networks on graphs.* Cornell University. https://arxiv.org/pdf/1312.6203

Chen, M., Mozaffari, M., Saad, W., Yin, C., Debbah, M., & Hong, C. S. (2017). Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. *IEEE Journal on Selected Areas in Communications*, *35*(5), 1046–1061. doi:10.1109/JSAC.2017.2680898

Cheng, N., Xu, W., Shi, W., Zhou, Y., Liu, N., Zhou, H., & Shen, X. (2018). Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities. *IEEE Communications Magazine*, *56*(8), 26–32. doi:10.1109/MCOM.2018.1701092

Du, L., Shi, X., Fu, Q., Liu, H., Han, S., & Zhang, D. (2022). GBK-GNN: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. *Proceedings of the ACM Web Conference 2022*. doi:10.1145/3485447.3512201

Fan, X., Liu, Z., Lian, J., Zhao, W. X., Xie, X., & Wen, J. (2021). Lighter and Better: Low-Rank Decomposed Self-Attention Networks for Next-Item Recommendation. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* doi:10.1145/3404835.3462978

Han, S., Xue, F., Yang, C., Liu, J., & Lin, F. (2021). Data-supported caching policy optimization for wireless D2D caching networks. *IEEE Transactions on Communications*, *69*(11), 7618–7630. doi:10.1109/TCOMM.2021.3104634

Jiang, Y., Ma, H., Liu, Y., Li, Z., & Chang, L. (2021). Enhancing social recommendation via two-level graph attentional networks. *Neurocomputing*, *449*, 71–84. doi:10.1016/j.neucom.2021.03.076

Kipf, T., & Welling, M. (2016). *Semi-supervised classification with graph convolutional networks.* Cornell University. http://export.arxiv.org/pdf/1609.02907

Koren, Y., Bell, R. M., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. doi:10.1109/MC.2009.263

Li, B., Fei, Z., & Zhang, Y. (2019). UAV communications for 5G and beyond: Recent advances and future trends. *IEEE Internet of Things Journal*, *6*(2), 2241–2263. doi:10.1109/JIOT.2018.2887086

Liu, X., Liu, Y., & Chen, Y. (2019). Reinforcement learning in multiple-UAV networks: Deployment and movement design. *IEEE Transactions on Vehicular Technology*, *68*(8), 8036–8049. doi:10.1109/TVT.2019.2922849

Luo, L., Fang, Y., Cao, X., Zhang, X., & Zhang, W. (2021). Detecting communities from heterogeneous graphs: A context path-based graph neural network model. *CIKM 21: Proceedings of the 30th ACM International Conference on Information & Knowledge management* (pp. 1170–1180). Association for Computing Machinery. doi:10.1145/3459637.3482250

Navarro-Ortiz, J., Romero-Diaz, P., Sendra, S., Ameigeiras, P., Ramos-Munoz, J. J., & Lopez-Soler, J. M. (2020). A survey on 5G usage scenarios and traffic models. *IEEE Communications Surveys and Tutorials*, *22*(2), 905–929. doi:10.1109/COMST.2020.2971781

Park, Y. M., Lee, M., & Hong, C. S. (2019). Multi-UAVs collaboration system based on machine learning for throughput maximization. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 1–4). IEEE. https://doi.org/ doi:10.23919/APNOMS.2019.8892962

Reiss-Mirzaei, M., Ghobaei-Arani, M., & Esmaeili, L. (2023). A review on the edge caching mechanisms in the mobile edge computing: A social-aware perspective. *Internet of Things*, *22*, 100690. doi:10.1016/j.iot.2023.100690

Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., & Tang, J. (2019). Session-based social recommendation via dynamic graph attention networks. In *WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (pp. 555–563). Association for Computing Machinery. doi:10.1145/3289600.3290989

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, *2009*, 1–19. doi:10.1155/2009/421425

Su, X., & Khoshgoftaar, T. M. (2009). *A survey of collaborative filtering techniques*. Academic Press.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. Cornell University. 10.17863/cam.48429

Wang, X., Chen, M., Taleb, T., Ksentini, A., & Leung, V. C. M. (2014). Cache in the air: Exploiting content caching and delivery techniques for 5G systems. *IEEE Communications Magazine*, *52*(2), 131–139. doi:10.1109/MCOM.2014.6736753

Wang, Z., Chen, Y., Huang, X., Li, J., & Min, G. (2023). GNN-based long and short term preference modeling for next-location prediction. *Information Sciences*, *629*, 1–14. doi:10.1016/j.ins.2023.02.073

Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., & Chen, G. (2019). Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. *WWW '19: The World Wide Web Conference* (pp. 2091-2102). Association for Computing Machinery. doi:10.1145/3308558.3313442

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 346–353. doi:10.1609/aaai.v33i01.3301346

Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Ding, B., & Cui, B. (2022). Contrastive learning for sequential recommendation. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. doi:10.1109/ICDE53745.2022.00099

Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang, F., Fang, J., & Zhou, X. (2019). Graph Contextualized Self-Attention Network for Session-based Recommendation. *IJCAI (United States)*, 3940–3946. Advance online publication. doi:10.24963/ijcai.2019/547

Xu, J., Ota, K., & Dong, M. (2018). Saving energy on the edge: In-memory caching for multi-tier heterogeneous networks. *IEEE Communications Magazine*, *56*(5), 102–107. doi:10.1109/MCOM.2018.1700909

Yang, J., Xiao, S., Jiang, B., Song, H., Khan, S., & Islam, S. U. (2020). Cache-enabled unmanned aerial vehicles for cooperative cognitive radio networks. *IEEE Wireless Communications*, *27*(2), 155–161. doi:10.1109/MWC.001.1900301

Yang, W., Li, J., Tan, S., Tan, Y., & Lu, X. (2023). A heterogeneous graph neural network model for list recommendation. *Knowledge-Based Systems*, *277*(110822), 110822. Advance online publication. doi:10.1016/j.knosys.2023.110822

Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., & Tan, T. (2020). TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. doi:10.1145/3397271.3401319

Zeng, B., Zhan, C., Yang, Y., & Liao, J. (2022). Access delay minimization for scalable videos in cache-enabled multi-UAV networks. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. doi:10.1109/GLOBECOM48099.2022.10001632

Zhang, M., El-Hajjar, M., & Ng, S. X. (2022). Intelligent caching in UAV-aided networks. *IEEE Transactions on Vehicular Technology*, *71*(1), 739–752. doi:10.1109/TVT.2021.3125396

Zhang, M., Wu, S., Yu, X., Liu, Q., & Wang, L. (2022). Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *1*, 1. Advance online publication. doi:10.1109/TKDE.2022.3151618

Zhang, S., Zeng, Y., & Zhang, R. (2019). Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective. *IEEE Transactions on Communications*, *67*(3), 2580–2604. doi:10.1109/TCOMM.2018.2880468

Zhang, T., Wang, Y., Liu, Y., Xu, W., & Nallanathan, A. (2020). Cache-enabling UAV communications: Network deployment and resource allocation. *IEEE Transactions on Wireless Communications*, *19*(11), 7470–7483. doi:10.1109/TWC.2020.3011881

Zhao, N., Yu, F. R., Fan, L., Chen, Y., Tang, J., Nallanathan, A., & Leung, V. C. M. (2019). Caching unmanned aerial vehicle-enabled small-cell networks: Employing energy-efficient methods that store and retrieve popular content. *IEEE Vehicular Technology Magazine*, *14*(1), 71–79. doi:10.1109/MVT.2018.2881228

Zhou, K., Yu, H., Zhao, W. X., & Wen, J. (2022). Filter-enhanced MLP is all you need for sequential recommendation. In *WWW '22: Proceedings of the ACM Web Conference 2022* (pp. 2388–2399). Association for Computing Machinery. doi:10.1145/3485447.3512111

Wei Wang is a professor and master's supervisor at the School of Information and Electrical Engineering, Hebei Engineering University. His research interests include implicit human-computer interaction and the application of IoT in public safety. He is the corresponding author of this paper and can be contacted via wangwei83@hebeu.edu.cn.

LongXing Xing is a researcher specializing in recommendation systems and deep learning. He completed his undergraduate studies in Internet of Things Engineering at Hebei Engineering University in 2021 and is currently pursuing a master's degree in Computer Technology at the same university, expected to graduate in 2024.

Na Xu was born on November 20, 1997, he began his postgraduate studies at Hebei University of Engineering in 2021. He specializes in UAV-assisted edge computing.

Jiatao Su was born in China,in August 1997, he was admitted as a postgraduate of Hebei University of Engineering in 2021. I specialize in unmanned systems and am skilled in path planning algorithms.

Wenting Su was born in June 1997 in China. She is now a second-year graduate student in the School of Information and Electrical Engineering, Hebei University of Engineering. Her research direction is UAV system.

Jiarong Cao was born in China, in October 1998.She is now a postgraduate student at Hebei University of Engineering. Her area of interest is UAV communication.