sensors

MDPI

Article

Joint Content Placement and Storage Allocation Based on Federated Learning in F-RANs

Tuo Xiao [1,2], Taiping Cui [1,2,*], S. M. Riazul Islam [3] and Qianbin Chen [2]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Nan-An District, Chongqing 400065, China; S180131014@stu.cqupt.edu.cn
[2] Chongqing Key Labs of Mobile Communications, Chongqing 400065, China; chenqb@cqupt.edu.cn
[3] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Korea; riaz@sejong.ac.kr
* Correspondence: cuitp@cqupt.edu.cn; Tel.: +86-187-1628-5097

Abstract: With the rapid development of mobile communication and the sharp increase of smart mobile devices, wireless data traffic has experienced explosive growth in recent years, thus injecting tremendous traffic into the network. Fog Radio Access Network (F-RAN) is a promising wireless network architecture to accommodate the fast growing data traffic and improve the performance of network service. By deploying content caching in F-RAN, fast and repeatable data access can be achieved, which reduces network traffic and transmission latency. Due to the capacity limit of caches, it is essential to predict the popularity of the content and pre-cache them in edge nodes. In general, the classic prediction approaches require the gathering of users' personal information at a central unit, giving rise to users' privacy issues. In this paper, we propose an intelligent F-RANs framework based on federated learning (FL), which does not require gathering user data centrally on the server for training, so it can effectively ensure the privacy of users. In the work, federated learning is applied to user demand prediction, which can accurately predict the content popularity distribution in the network. In addition, to minimize the total traffic cost of the network in consideration of user content requests, we address the allocation of storage resources and content placement in the network as an integrated model and formulate it as an Integer Linear Programming (ILP) problem. Due to the high computational complexity of the ILP problem, two heuristic algorithms are designed to solve it. Simulation results show that the performance of our proposed algorithm is close to the optimal solution.

Keywords: fog radio access network; content placement; storage allocation; federated learning

Citation: Xiao, T.; Cui, T.; Islam, S.M.R.; Chen, Q. Joint Content Placement and Storage Allocation Based on Federated Learning in F-RANs. *Sensors* **2021**, *21*, 215. https://doi.org/10.3390/s21010215

Received: 8 December 2020
Accepted: 25 December 2020
Published: 31 December 2020

1. Introduction

Recently, the increasing popularity of intelligent devices such as wearable devices, smartphones and sensors in our daily life has triggered a surge in many distributed network devices, which results in massive amounts of heterogeneous data that need to be processed [1–3]. Due to such unprecedented amount of data with exponential growth trend [4], it becomes impractical to send all data to a remote cloud computing center for processing and is full of privacy issues [2]. In addition, some applications and services rely heavily on high-speed data rates and low latency transmission, which prompts the mobile network operators to rethink current network architectures and seek more complex and advanced technologies to bring content closer to end users with low latency and cost.

To satisfy the diverse multi-dimensional requirements of quality of service (QoS), such as low-latency transmission, enhanced broadband and ultra-reliability, a fog radio access network paradigm has been proposed as a promising evolution path for the future wireless network architecture [5,6]. By integrating fog computing into wireless networks, it enables the distribution of cloud computing power to the edge of the network, enabling context-aware services and applications to approach mobile users. With this location, fog devices provide a unique opportunity not only to implement edge caching, but also to

perform edge processing. Therefore, we can intuitively use fog computing resources to design a new intelligent content caching and distribution mechanism, which are more flexible and can meet the QoS requirements of various application scenarios.

Due to the limited storage capacity of edge nodes, the network performance can be effectively improved by predicting the content popularity and actively caching the most popular one. However, most existing caching schemes are designed for highly controlled environments where users need to upload local private data to a central server, which may pose privacy and security risks [7]. Furthermore, with the increase of the number of users and data, the unreliability and communication cost of wireless networks cannot be ignored. Therefore, it is necessary to study a new network architecture with low communication cost and high reliability. To improve the caching performance of the edge network, a federated learning framework [1,2,8,9] is introduced to effectively predict the distribution of content popularity in the network. As a data-level distributed learning paradigm, federated learning is seen as a promising approach to generate high-quality models without having to collect all the local data at the server [10]. In the federated learning framework, each client trains its model based on the local data, and updates the global model accordingly by uploading the results of the training to the fog server. The fog server then returns the improved global parameters to user so that a new round of local training can begin. Finally, through model-level collaboration between the client and server, an accurate learning model can be generated. The core benefit of federated learning is to spread content over a large number of devices rather than having to centralize training data [11,12]. By applying federated learning to demand prediction problems, the users preference can be accurately predicted [13]. The realization of federated learning requires network edge devices to have powerful computing capabilities and flexible collaboration. Due to the sufficient fog computing resources, the F-RANs paradigm can fully support this.

The rapid increase in mobile data traffic places a heavy burden on the fronthaul link which connects fog servers to remote cloud centers. By caching content at a fog server, the content delivery rate can be improved, the cost of network traffic transmission can be reduced [14] and the quality of data transmission can be guaranteed. When a user requests content, the fog server that caches the content can provide the data directly, rather than fetching the content from a remote cloud computing center. Therefore, how to place the content on which caches nodes in the network is critical. Furthermore, caching performance is highly correlated with the capacity of storage. If the fog server is allocated with less storage, only a limited amount of content can be cached, which can result in a lower quality of service than larger cache storage. Therefore, to maximize the use of storage resources, an effective caching strategy must be designed to distribute storage across different network cache nodes, and storage resource allocation determines how many storages should be allocated to each fog server.

In this paper, we investigate jointly optimizing storage resource allocation and content placement in a caching enabled hierarchical F-RAN architecture, with the goal of minimizing network traffic costs. Moreover, considering users' content requests and privacy security, we adopt a federated learning to make distributed prediction of user preferences in different F-APs and apply it to the design of cache policy. The proposed caching scheme can effectively improve the performance of network content caching.

The rest of this paper is organized as follows. We review related work in Section 2. The system model is described in Section 3. In Section 4, we describe the formulation of our optimization problem. A solution is provided in Section 5. Simulation results are discussed in Section 6, and the Conclusion and Future Work are drawn in Section 7.

2. Related Work

The core idea of F-RANs is to make full use of the rich computing resources, data sharing and storage capabilities of edge devices. Edge caching between Fog-computing Access points (F-APs) can bring content closer to mobile users, which effectively improves content delivery rate and reduces the heavy burden of link transmission.

Currently, some contributions focused on designing edge caching schemes or algorithms to improve the performance of F-RANs. The work of [15] summarized the latest progress in F-RANs performance analysis, which introduces advanced edge caching and adaptive model selection schemes to improve the spectrum and energy efficiency of F-RANs. Effective caching strategies for F-RAN was given in [16] where F-RAN refers to the cloud radio access network (C-RAN) architecture that utilizes distributed edge caching techniques [6,17]. The work of [18] discussed the impact of mobile social networks on the performance of F-RANs edge caching schemes from a perspective of users' social relationships. The work of [19] make use of social information and edge computing to reduce the end-to-end delay effectively, and the network content caching, mobility management and wireless access control has been studied. Although some researches have been done on caching in F-RAN, there are few researches on the joint optimization of resource allocation and content placement in networks. In this paper, we focus on the optimization of joint content caching and resource allocation in the F-RAN architectures to further improve the caching performance of F-RANs.

Due to the limited storage capacity of edge devices, the content which is most likely to be requested by the user must be placed at the local fog server. The traditional cache mechanism updates the cache content based on static rules such as first in first out (FIFO) [20], least recently used (LRU) [21] and least frequently used (LFU) [22]. However, the popularity of content in the network changes over time, making this approach impractical. Currently, many researches focused on developing of dynamic caching schemes based on content popularity. The works of [23–25] modeled the caching problem as a Multi-Arm Bandit (MAB) problem, and indirectly obtained the content popularity distribution according to the cumulative request rate of all content. However, since the content popularity prediction process requires online cache training for all content, the prediction is not real-time and computationally complex. The work of [26] modeled the content popularity prediction problem as a Contextual Multi-Arm Bandit (CMAB) problem. In order to improve the accuracy of prediction, the scene information of all requesting users is partitioned, and the online prediction method similar to literature [23] is used in different scene partitions. Although the prediction accuracy has improved, it has not improved in terms of real-time prediction and computational complexity. The work of [27,28] used Alternating Direction Method of Multipliers (ADMM) algorithm to found highly popular content for caching through dynamic iteration. However, due to the prediction of content popularity in dynamic calculation, the cache timeliness cannot be guaranteed.

In addition, due to the dynamic nature of content and the mobility of user, the content popularity in the unit changes dynamically over time. Therefore, many studies use machine learning to learn content popularity by observing users' historical content needs. Bastug et al. [29] proposed a small cellular network caching algorithm based on Collaborative Filtering (CF). However, CF algorithm has high computational complexity and is prone to cold-start when the data is sparse, which will affect the accuracy of content popularity prediction. The work of [30] proposed an active content caching mechanism based on transfer learning (TL), which was to minimize the content transmission cost of the system. It solves the problem of data sparseness, but if the similar content is migrated improperly, it will make the prediction accuracy worse. The work of [31] proposed using the Extreme Learning Machine (ELM) algorithm to build a model of the relationship between content features and user request information, and a random approximation algorithm is used for content feature selection design to improve the performance of the ELM algorithm. Finally, a trained model is used to predict future content popularity. However, the prediction algorithm cannot track the change of content popularity, and the algorithm accuracy still needs to be further improved.

Since most machine learning methods require the collection of individual user information at a central unit, which may cause privacy concerns for users. Local users have difficulty trusting the servers and are reluctant to upload their private data. In this context, federated learning as a distributed machine learning framework can effectively

address this problem. It can perform the learning process from the data spread across multiple users, thus protecting sensitive data. Applying the federated learning framework to the demand predicting problem can effectively predict the distribution of network content popularity [32,33]. The performance comparison of content popularity prediction methods is provided in Table 1, and the main contributions of this paper are as follows:

- We jointly considered storage resource allocation and content placement in the network to formulate an optimization problem to minimize the network traffic cost.
- Due to the dynamic change of content popularity in the network, the federated learning framework is applied to predict the content popularity accurately in the region to develop an efficient content caching strategy. To the best of our knowledge, the problem of federated learning-based joint content placement and storage allocation has not been well studied in previous works.
- Two heuristic algorithms are proposed, and the experimental results based on real-worlds datasets verify the performance superiority of our proposed algorithm.

Table 1. Performance Comparison.

Related Work	**[23–25]**	**[26]**	**[27,28]**	**[29]**	**[30]**	**[31]**	**This Work**
Online/Offline-Learning	Online	Online	Online	Offline	Offline	Offline	Online
High Computational	Yes	Yes	Yes	Yes	No	Yes	No
Accuracy	No	Yes	Yes	No	No	No	Yes
Real-Time	No	No	No	No	No	No	Yes
Privacy Protection	No	No	No	No	No	No	Yes

3. System Model

In this section, we first introduce a cache-enable F-RAN architecture and design a federated learning framework in F-RANs. Next, the content cache process is presented in details. Finally, the problem of content popularity prediction is formulated. Some key parameters are listed in Table 2.

Table 2. Key Parameters.

Notation	Definition
$\mathcal{N}$	Set of F-APs
N	Number of F-APs
$\mathcal{U}$	Set of mobile users
U	Number of mobile users
C	Storage budget of F-APs
$\mathcal{F}$	Library of popular contents
F	Total number of contents
s_f	Size of content f
p_f	Global content popularity
β	The skewness factor of Zipf
P_{nf}	Local content popularity
c_n	The storage capacity of F-AP n
$\mathbf{X}$	A binary content cache matrix
x_{nf}	Content cache decisions
Δ_u	Local dataset
α	Learning rate
$w_u(t)$	Local parameter vector
W^1,W^2,W^3	The traffic cost of wireless link, Fog-Fog link and fronthaul link, respectively

3.1. System Architecture

An illustrative network architecture of F-RANs is shown in Figure 1. We consider a cache-enabled F-RAN architecture that contains N F-APs which is equipped with the fog computing server (denoted as a set $\mathcal{N} = 1, 2, \ldots, N$) and U mobile users (denoted as a set $\mathcal{U} = 1, 2, \ldots, U$). F-APs exchange data with the cloud computing center through fronthaul links, and F-APs can communicate with each other and with a Cache Manager (CM) via X2 interface [5], to achieve content sharing. F-APs communicate with the users through wireless channels. We assume that each user can only download the requested content from the F-AP which is associated with it. Moreover, we consider allocating a certain amount of storage for each F-APs in the network, and the total caches of F-APs cannot exceed the upper limit C of the storage budget which is specified by the mobile network operator. CM can monitor all user-generated content requests [31] and is responsible for:

(1) Retrieve user's requested content from the cloud computing server;
(2) Maintain an index table for storing cached content locations in the network;
(3) Forward the user's content request to the neighboring F-APs that cache the content;
(4) Collect information about the requested content in F-APs;
(5) Decide when to update the entire content cache of F-APs, which can be refreshed at specific intervals or when content popularity changes significantly.

In the above-mentioned federated learning framework of F-RANs, each user can train a local model based on their own data, and then aggregate the local model at the fog computing server. The learning model is trained through the interaction between the fog computing server and the user until the model converges to a specific level of accuracy.

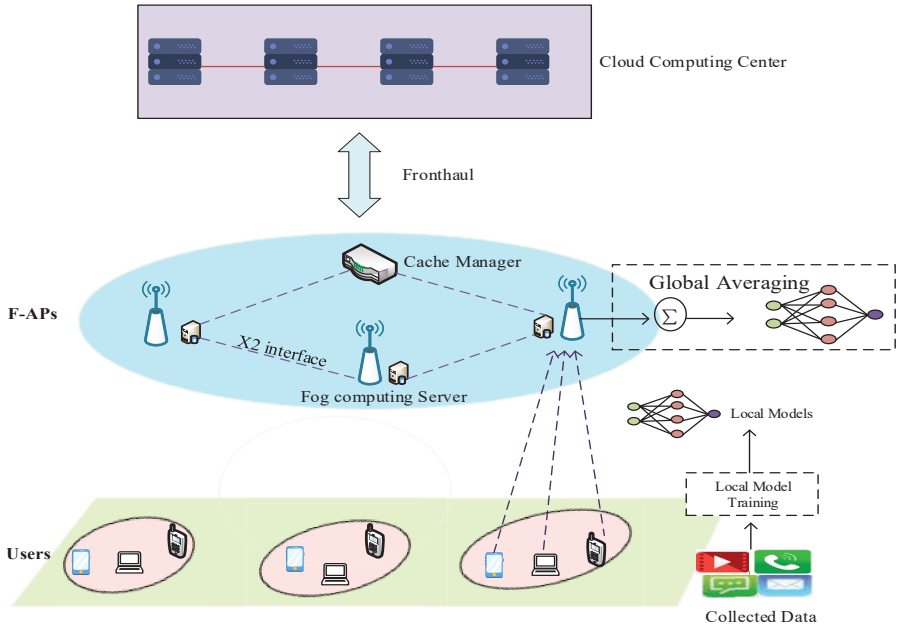

Figure 1. Federated learning paradigms in F-RANs.

3.2. Caching Process

The mobile user connects a F-APs, and the connected F-AP is responsible for serving the user's content requests. If a requested content is in the cache of the connected F-AP, the request is served immediately with no additional load placed on the fronthaul link, which reduces network traffic. On the other hand, if the F-AP does not cache the content requested by a local user, the request is forwarded to CM. The CM checks whether the contents requested in the lookup table are cached in neighbor F-APs. If the content is cached in the neighbor F-APs, CM will perform all necessary signaling to retrieve the content from the neighbor F-APs. Content provided by neighbor F-APs incurs lower downloading latency and reduces network traffic. Finally, if CM cannot find the requested content in any cache, it forwards the request to the remote cloud computing center for the content. Since dividing the content into small pieces and caching them at different levels will increase the complexity of the system, thus we assume that each piece of data is indivisible and can be cached on a F-AP as a whole.

Given the dynamic nature of network traffic, the content cached in the fog server should be updated regularly (e.g., an hour). At the beginning of each period, CM first optimizes content caching decisions and storage allocation strategies. If the reoptimization strategy is different from the previous phase, the cache can be updated and the cache storage can be reallocated accordingly.

3.3. Content Popularity

The set of popular content libraries requested by users in the network is represented as $\mathcal{F} = \{1, 2, \ldots, F\}$, and the average size of content is expressed as s_f. The status information from all users and each user requested content is defined as follows.

3.3.1. Global Content Popularity

The global content popularity in the network is defined as P_f, which represents the probability distribution of content requested by all mobile users in the system. The popularity of the f-th content can be calculated as the ratio of the number of requests for f content to the number of requests for all content in the network. The common preferences

of all users in the network can be expressed by the global content popularity, which usually follows a Zipf distribution model [34,35]:

$$P_f = \frac{(f)^{-\beta}}{\sum_{j=1}^{F}(j)^{-\beta}} \qquad ,\forall f \in \mathcal{F} \tag{1}$$

where β is the skewness factor. The higher the value of β, the higher the number of requests concentrated on a few (popular) contents.

3.3.2. Federated Learning Prediction

Due to different content preferences in different F-APs, and the probability that users from F-AP n requesting content f is defined as P_{nf}. User preferences can be predicted in advance or on a regular basis (e.g., hourly, daily, or weekly) through systematic learning and analysis of user social behavior [36,37]. In this paper, considering the privacy security of users, we adopt the federated learning method [7] to accurately predict the content popularity in the region.

As shown in Figure 2, the federated learning framework includes the user's device, which is responsible for local data training and uploading updates to the fog server. In general, the datasets used for local model training are generated based on the user's device usage, such as the user's web browsing and video playing in daily life. Different time and place, different activities, and even different types of mobile devices [26] may cause users to request different content. Therefore, the historical request information of users under different circumstances constitutes a part of the local training dataset. On the fog server, the global learning model is improved by merging and aggregating the local model updated from the user's device. Finally, the fog server sends the improved model parameters back to the client, and this step is termed as a round of communication. The details of our designed FL communication process consist of the following steps:

① Model Download:

As shown in Figure 2, step ①, a set of users U are selected to participate in FL training for the t-th communication round. The selected users then download the global model from the fog computing server and train the model with their own local data. Therefore, they download the parameters w_u of the global model from the fog computing server.

② Local Model Training:

The second step in our proposed FL is to train the model by utilizing local data at user devices, as shown in Figure 2, step ②. And at each round of algorithm iteration of t, the users participating in the training process are a subset of the entire user set. Each user u involved in the training process and updates its local parameter vector $w_u(t)$, implicitly built on the basis of its local dataset Δ_u, in accordance with the following rule [2]:

$$w_u(t) = \hat{w}_u(t-1) - \alpha \nabla F_u(\hat{w}_u(t-1)) \tag{2}$$

where α is the learning rate and $\hat{w}_u(t-1)$ represents the term $w_u(t-1)$ after global aggregation.

③ Upload Updated Model:

After completing the local model training, the users upload the local model parameters $w_u(t)$ to the fog computing server, as shown in Figure 2 step ③. In order to reduce communication costs and save the upload time, the model can be compressed before being uploaded to the fog computing server, as the uplink speed is slower than the download speed [38].

④ Weighted Aggregation:

After uploading their models, the last step is to generate the new global model $w(t)$ by computing a weighted sum of all received local models $w_u(t)$, as shown in Figure 2,

where t denotes the communication rounds in FL. The new constructed global model is used for the next training round. The fog server provides the weighted average suggested in [8], which is expressed as:

$$w(t) = \frac{\sum_{u\in\{1,\ldots,U\}} |\Delta_u| w_u}{\sum_{u\in\{1,\ldots,U\}} |\Delta_u|} \tag{3}$$

where $|\Delta_u|$ indicates the cardinality of Δ_u, i.e. the number of elements in Δ_u.

The distributed data training of the algorithm proposed above has some advantages in terms of user privacy and content exchange. In fact, client is trained on local data, which allows users to protect their sensitive information. In addition, for each round of algorithm iteration, only a portion of the user set is involved, ensuring reduced messaging between the client and server. Finally, it should be emphasized that by considering the perspective of a user device, the gradient descent algorithm is used for optimization without excessive resource consumption. Therefore, after training a shared global model, each F-APs can predict local content popularity and then use it for cache content placement.

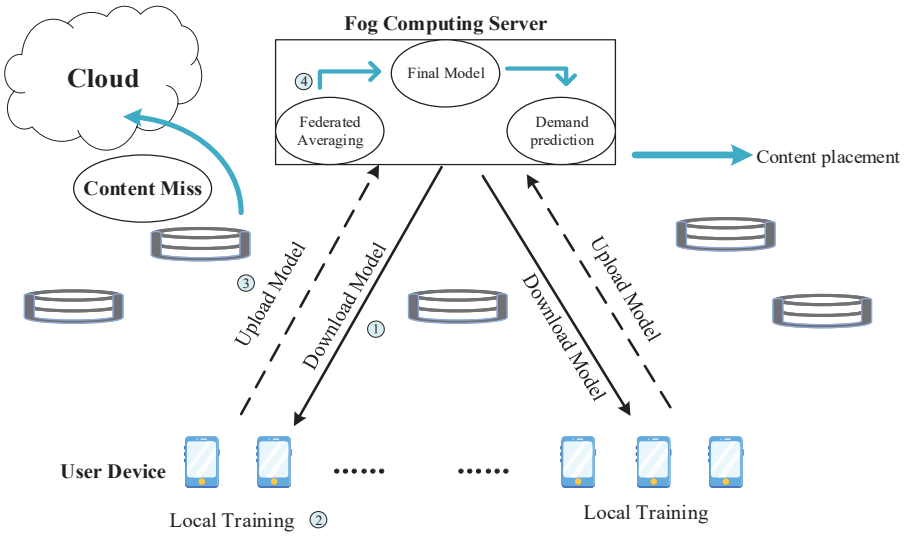

Figure 2. Federal learning prediction framework.

4. Problem Formulation

In this section, the problem is represented as a joint optimization of content caching and resource allocation with the objective of minimizing network traffic costs. The storage allocated for caching at F-AP n is denoted as c_n. A binary content cache matrix that is denoted as $\mathbf{X} = \{x_{nf} | n \in \mathcal{N}, f \in \mathcal{F}\}$ indicates whether content f is placed at the local cache of server n. $x_{nf} = 1$ if content f is precached in fog server n and $x_{nf} = 0$ otherwise. The content popularity in the region is expressed as P_{nf}, that is, the probability of content f requested by the user of F-AP n, which can be predicted by using the federated learning method. Therefore, the problem is formulated as follows:

$$\begin{aligned}
&\min_{x,c} \sum_{f=1}^{F} \sum_{n=1}^{N} \{P_{nf} \cdot U_n \cdot s_f [W^1 x_{nf} + (W^1 + W^2)(1 - x_{nf}) x_{mf} \\
&+ (W^1 + W^3)(1 - x_{nf})(1 - x_{mf})]\} \\
\text{s.t.} \quad &\text{C1} : \sum_{n=1}^{N} c_n \leq C \\
&\text{C2} : \sum_{f=1}^{F} x_{n,f} s_f \leq c_n, \quad \forall n \in \mathcal{N} \\
&\text{C3} : x_{nf} \in \{0,1\}, \quad \forall n \in \mathcal{N}, \quad \forall f \in \mathcal{F} \\
&\text{C4} : x_{mf} \in \{0,1\}, \quad \forall m \in \mathcal{N}, \quad \forall f \in \mathcal{F}
\end{aligned} \tag{4}$$

where the first item of objective function (4) denotes that the content f is cached in the local F-AP n and constitutes the traffic through the wireless channel between user and F-APs. The second term indicates that content f is cached in neighbor F-APs m and constitutes the traffic through the Fog-Fog links and wireless channels. The third item represents that content f is requested from the cloud computing center and the traffic comes from the wireless channel and fronthaul links. The constraint C1 means that the cache allocated in all F-APs should not exceed the storage budget C. The constraint C2 means that all data in each F-APs should not surpass its storage capacities. The constraints C3 and C4 represent the caching decisions of fog servers in the network. Due to the product term in (4), the problem is nonlinear and difficult to solve. In this case, we introduce another binary decision variable z_{nf} to enable $z_{nf} = x_{nf} x_{mf}$. In order to ensure that the transformed problem is equivalent to the original problem, the condition C5–C7 needs to be satisfied. Therefore, the converted problem can be expressed as follows:

$$\begin{aligned}
&\min_{x,c,z} \sum_{f=1}^{F} \sum_{n=1}^{N} \{P_{nf} \cdot U_n \cdot s_f [(W^1 + W^3) - W^3 x_{nf} + (W^2 - W^3) x_{mf} - (W^2 - W^3) z_{nf}]\} \\
\text{s.t.} \quad &\text{C5} : z_{nf} \leq x_{nf}, \quad \forall n \in \mathcal{N}, \quad \forall f \in \mathcal{F} \\
&\text{C6} : z_{nf} \leq x_{mf}, \quad \forall m \in \mathcal{N}, \quad \forall f \in \mathcal{F} \\
&\text{C7} : z_{nf} \geq x_{nf} + x_{mf} - 1, \quad \forall n, m \in \mathcal{N}, \quad \forall f \in \mathcal{F}
\end{aligned} \tag{5}$$

Transformation problem (5) is an integer linear programming (ILP) problem, which can be solved by exhaustive search algorithm, but with the high computational complexity and poor system performance. Therefore, two low complexity sub-optimal algorithms are designed to improve the performance of the system in the next section.

5. Problem Solution

Due to the high complexity of joint optimization problem calculation, in this section, we propose two sub-optimal heuristic algorithms to solve the problem, which can effectively improve the time efficiency. We decompose the joint optimization problem of storage allocation and content caching into two sub-problems. We first address the allocation of storage resources and then use it for the placement of cached content.

5.1. Storage Resource Allocation Problem

For storage allocation problems, storage resources are allocated to each F-APs based on the total F-APs storage budget. Therefore, the algorithm should be designed according to the different traffic requirements of F-APs to maximize the utilization of fog server storage resources. Traffic demand in the network is related to the popularity of the content, the number of users and the size of the content. For F-APs with high traffic requirements, more cache storage should be allocated. Therefore, we propose a traffic-based allocation

algorithm, which allocates storage proportionally according to different traffic requirements. The Algorithm 1 in detail is shown as follows:

Algorithm 1: Traffic-based allocation Algorithm

Input: U_n , P_{nf} , s_f , C;
Output: storage allocation strategy c;
Initialize the total F-APs' traffic $Tra = 0$;
for *each F-AP n* **do**
  Calculate traffic demand T_n;
  $T_n = P_{nf} s_f U_n$;
end
Calculate the total F-APs' traffic Tra;
$Tra = Tra + T_n$;
for *each F-AP n* **do**
  $c_n = \frac{T_n}{Tra} * C$;
end
return c;

5.2. Cache Content Placement Problem

The content placement problem determines which content should be cached on each F-AP to minimize the traffic costs. Here two heuristics algorithms are proposed to address the problem of content placement.

5.2.1. Greedy Algorithm based on Global Content Popularity

Due to the importance of content popularity in cache policy design, caching content with high popularity performs better. Greedy algorithm is adopted to cache as many popular content as possible on each cache entity. Specifically, the greedy algorithm based on global content popularity caches the most popular content in each F-APs until reaching the cache storage capacity limit. Algorithm 2 shows the process in detail. From the practical perspective, as different F-APs have their own preferences, the shortcoming of this algorithm is that it does not consider the content preference of regional users and the resource utilization is insufficient.

Algorithm 2: Greedy Algorithm based on Global Content Popularity

Input: N , F , U_n , P_f , s_f , W^1 , W^2 , W^3 , c;
Output: Traffic cost P , content placement decisions $\mathbf{X}$;
for *each F-AP n* **do**
  Use all available storage resources c_n;
  for *contents in descending order of content popularity* P_f **do**
    Cache the most popular content in each F-AP n until the storage capacity is full;
  end
  Get content placement decision x_{nf};
end
x_{nf} is the element of $\mathbf{X}$;
Substitute N , F , U_n , P_f , s_f , W^1 , W^2 , W^3 , c, $\mathbf{X}$ into (5) to calculate traffic cost P;
return $P, \mathbf{X}$;

5.2.2. Local Popularity Knapsack Algorithm based on Federated Learning

In this section, in order to reduce network traffic, a local popularity knapsack algorithm based on federated learning is proposed. The algorithm considers the local content popularity of each F-APs and avoids the deficiency of Algorithm 2. Firstly, the content that

each fog server needs to cache depends on the content popularity P_{nf} and the content size s_f. Therefore, the content caching decision of fog server n can be expressed as:

$$\begin{aligned} &\max_{x} \sum_{f=1}^{F} P_{nf} s_f x_{nf} \\ &\text{s.t.} \quad \text{C8}: \sum_{f=1}^{F} x_{nf} s_f - c_n \leq 0 \end{aligned} \tag{6}$$

where C8 means that all content cached in F-AP n should not exceed its capacity limit. According to Equation (6), we prefer to cache high popularity and large data length content in each fog server. It is observed that Equation (6) is a 0–1 knapsack problem [39], where $x_{nf} \in [0, 1]$ is content placement decision, $x_{nf} = 1$ means fog server n cache content f, otherwise $x_{nf} = 0$, s_f is the weight of content item f, c_n is the knapsack capacity, and $P_{nf} s_f$ is the value of each item. Therefore, we can use dynamic programming [39] to solve the 0–1 knapsack problem.

The principle of dynamic programming is to divide the original problem into several subproblems and solve the subproblems by looking for the recurrence relation between the original problem and the subproblems, and finally achieve the effect of solving the original problem. In order to decompose the original problem into subproblems, a matrix v is constructed. If the knapsack capacity of the cached content item $\{1, 2, \ldots, f\}$ is j, then $v(f, j)$ represents the maximum target value that can be obtained. Therefore, the optimal solution is $v(F, c_n)$, and the relationship between the original problem and the subproblem is:

$$v(f, j) = \begin{cases} v(f-1, j), & if \quad j < s_f \\ max\{v(f-1, j), v(f-1, j-s_f) + P_{nf} s_f\}, & otherwise \end{cases} \tag{7}$$

If the storage capacity j is less than the size s_f of the content item f, the fog server cannot cache the content. Therefore, we can remove content f in our solution and only consider caching data from $\{1, 2, \ldots, f-1\}$, that is $v(f, j) = v(f-1, j)$. Otherwise, we choose the optimal one between caching or not caching the data for the content item f. The first item in curly bracket indicates that the content item f is not cached in F-APs and does not have any impact on the original problem or take up any storage. The second item means that the content item f is cached in F-AP, so the $P_{nf} s_f$ value is added to the original problem and occupies the storage capacity of s_f.

Algorithm 3 describes the knapsack algorithm process of local content popularity based on federated learning. With this algorithm, we can calculate all the entries of v and trace back the entries of v according to the optimal solution to determine the contents cached in F-APs n.

Algorithm 3: Local Popularity Knapsack Algorithm based on Federated Learning

Input: N , F , U_n , P_{nf} , s_f , W^1 , W^2 , W^3 , c;
Output: Traffic cost P , content placement decisions $\mathbf{X}$;
for *each F-AP* n **do**
  Initialize v as a $(F+1) \times (c_n+1)$ zero matrix;
  for $f = 1 \rightarrow F$ **do**
    for $j = 1 \rightarrow c_n$ **do**
      if $s_f > j$ **then**
        $v(f+1, j+1) = v(f, j+1)$;
      else
        $v(f+1, j+1) = \max\{v(f, j+1), v(f, j+1-s_f) + P_{nf} s_f\}$;
      end
    end
  end
  $V = v(F+1, c_n+1), f = F, j = c_n$;
  while $V > 0$ **do**
    while $v(f+1, j+1) == V$ **do**
      $j = j - 1$;
    end
    $f = f + 1$;
    $x_{nf} = 1$;
    $j = j - s_f$;
    $f = f - 1$;
    $V = v(f+1, j+1)$;
  end
end
x_{nf} is the element of $\mathbf{X}$;
Substitute N , F , U_n , P_{nf} , s_f , W^1 , W^2 , W^3 , c, $\mathbf{X}$ into (5) to calculate traffic cost P;
return $P, \mathbf{X}$;

6. Simulation Results

In this section, the experimental results of the proposed algorithms are investigated, and the performance of three other algorithms, that is Oracle, No Storage Allocation and Random, are provided as references.

6.1. Simulation Parameters

In our simulation, we set the number of F-APs $N = 30$. Since more users are sharing a fronthaul link than a Fog-Fog link in the network, the fronthaul link is more likely to cause traffic congestion. Therefore, the traffic cost of wireless link, Fog-Fog link and the fronthaul link are W^1, W^2 and W^3 respectively, where $W^3 > W^2 > W^1$. According to the parameters used in [40], in our simulation experiment, the values of W^1, W^2 and W^3 are set to 1, 2, and 4 per MB, respectively. Mobile users are randomly distributed among different F-APs, where the number of users $U = 1000$. The average size of content is s_f, and the actual size of all content in the network is randomly selected from $0.8s_f$ to $1.2s_f$. That global content popularity in the network can be represented by Zipf distribution with $\beta = 0.56$, which agrees with the used models in [34]. Since different F-APs have their own preferences, we represent the probability of user request f in F-AP n as P_{nf}. Considering the privacy security of users, federated learning framework is adopted to accurately predict content popularity in F-APs. The parameters used in the simulation experiment are shown in Table 3.

Table 3. Simulation Parameters.

Parameter Name	Value
Number of F-APs	$N = 30$
Number of users	$U = 1000$
The traffic cost of wireless link	$W^1 = 1$ MB
The traffic cost of Fog-Fog link	$W^2 = 2$ MB
The traffic cost of fronthaul link	$W^3 = 4$ MB
The total storage budgets of F-APs	$C = 1500$ MB
The average content size	$s_f = 10$ MB
Zipf distribution skewness parameter	$\beta = 0.56$

6.2. Datasets

In our experiment, we used real-world datasets—MovieLens [41]. The MovieLens dataset contains ratings data for multiple movies by multiple users, as well as movie metadata information and user attribute information. The MovieLens 1M dataset contained 1,000,209 ratings for 3706 movies participated by 6040 users, while the MovieLens 100K dataset had 100,000 ratings for 1682 movies from 943 users. Each user reviews at least 20 movies, and the user rating is based on a five-star scale, that is, from 0 to 5. In this paper, to simulate the process of users requesting content, we assume that the movie participating in the rating is the content requested by the user, and each movie rating corresponds to a content download. The paper of [26,42] adopt a similar method to simulate the process of users requesting content.

6.3. Evaluation and Discussion

The evaluation is based on two different sizes of datasets: MovieLens 1M and 100K. And the algorithms proposed in this paper are compared with the following three algorithms:

(i) Oracle: The algorithm has a priori knowledge of content popularity and provides optimal cache performance.
(ii) No storage allocation (NoStrgAlloc): The content popularity follows the Zipf distribution and does not consider the storage resource allocation of the fog computing server.
(iii) Random: The random algorithm randomly selects F content for caching, which provides the lowest caching performance.

Figures 3 and 4 depict the cache hit rate (HR) against the number of federated communication rounds with different numbers of participated users. The results indicate that the more users participated in the learning process, which an accurate result can be trained to achieve better cache performance. In addition, fewer communication rounds are needed with more users or larger datasets. Therefore, the model can be updated by increasing algorithm rounds to improve system performance and make HR reach a higher value.

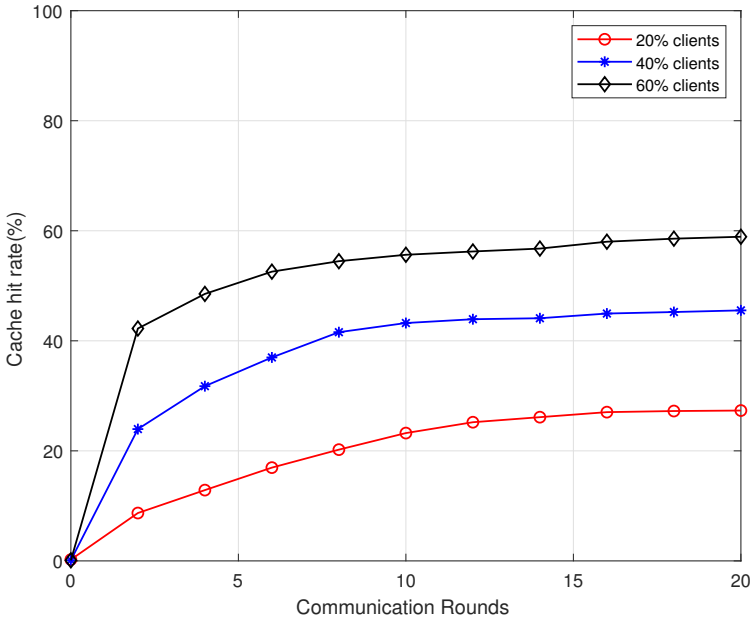

Figure 3. Cache hit rate vs communication rounds for datasts of MovieLens 100K.

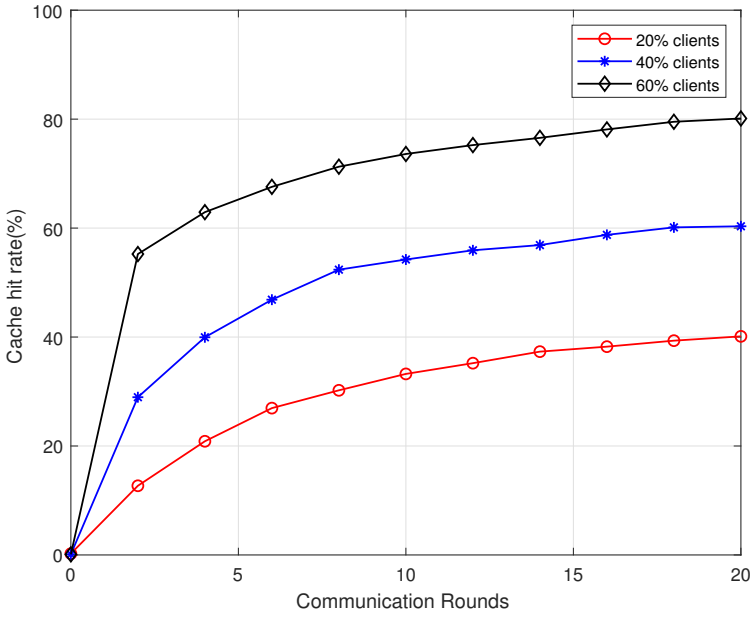

Figure 4. Cache hit rate vs communication rounds for datasts of MovieLens 1M.

The performances of GA and FL algorithm under different average content sizes are evaluated. The total storage capacity of all F-APs in the network is set to $C = 1500$ MB. Figures 5 and 6 show the traffic cost with different average content sizes ranging from 6 MB to 12 MB. As can be seen from Figures 5 and 6, the traffic cost increases with the average content size. Since the larger the content size is, the more network traffic will be generated. Oracle algorithms provide the lowest traffic cost because it has a perfect prior knowledge of future user requirements. The random algorithm has not considered the content popularity and the allocation of storage resources, thus resulting in the highest traffic cost. The algorithm without considering storage resource allocation in F-APs produces the second highest traffic cost. FL algorithm performs better than GA because FL considers local

popularity instead of global popularity, avoiding the disadvantage we discussed earlier that only considered global popularity.

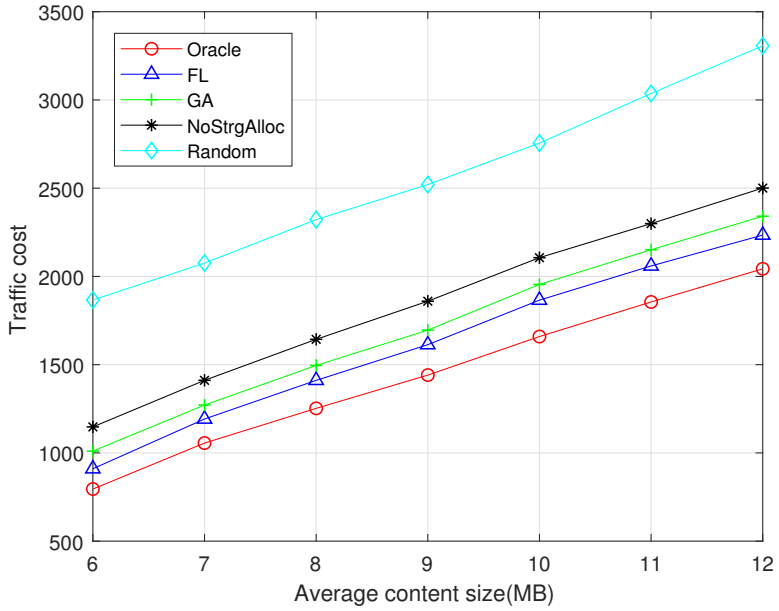

Figure 5. Traffic cost vs average content size (MovieLens 100K).

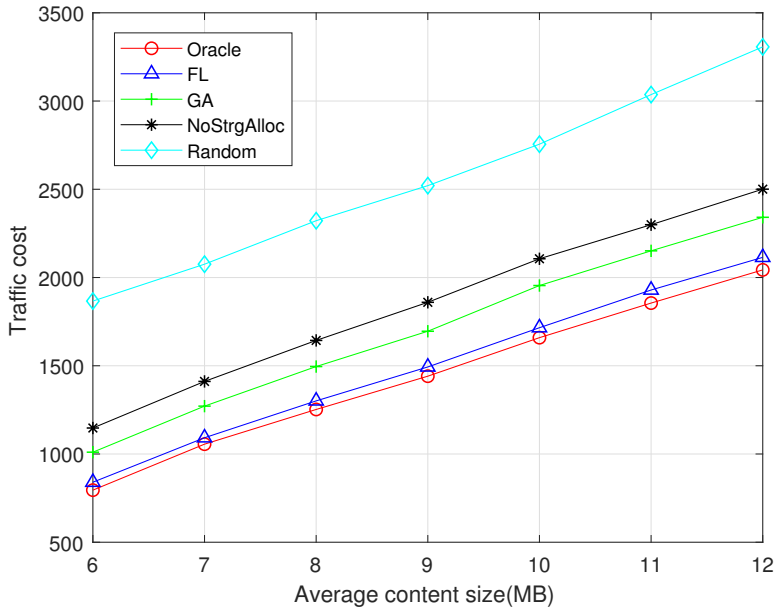

Figure 6. Traffic cost vs average content size (MovieLens 1M).

In comparing Figure 6 with Figure 5, observe that the performance of FL algorithm is closer to the Oracle (optimal) when the datasets is 1M rather than 100K, because FL algorithm has a better training effect when the datasets is larger and can predict the content popularity more accurately in the region. Due to the limited storage capacity of edge network nodes, pre-cached popular content can serve more user requests, thus reducing the network traffic costs effectively.

The storage budget is an important metric that should be considered in designing caching strategies. Figures 7 and 8 show the relationship between traffic costs and the F-APs cache budget. We compare the impact of F-AP cache budget C on traffic cost in different

algorithms, and the average content size is set to 10 MB. In both Figures 7 and 8, it is observed that the traffic cost decreases with F-APs cache budgets because more popular content can be cached in F-APs and hence less traffic is incurred to both the fronthaul and Fog-Fog links. The performance of Oracle algorithm is best, the random algorithm has the worst performance, and the cache without storage allocation is the second worst. FL performs better than GA because FL consider local popularity instead of global popularity. Compared with Figure 7, Figure 8 shows that the more datasets involved in model training, the better prediction can be achieved and the performance of the proposed algorithm is closer to the optimal one.

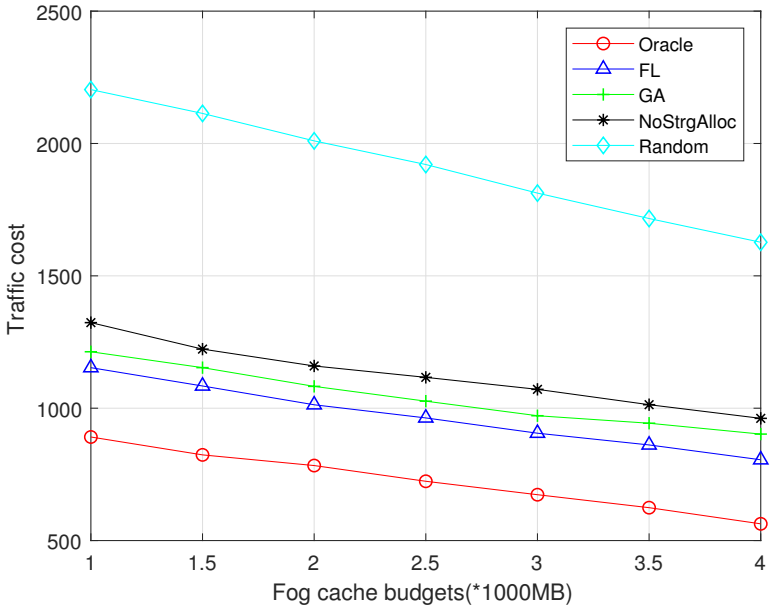

Figure 7. Traffic cost vs Fog cache budgets (MovieLens 100K).

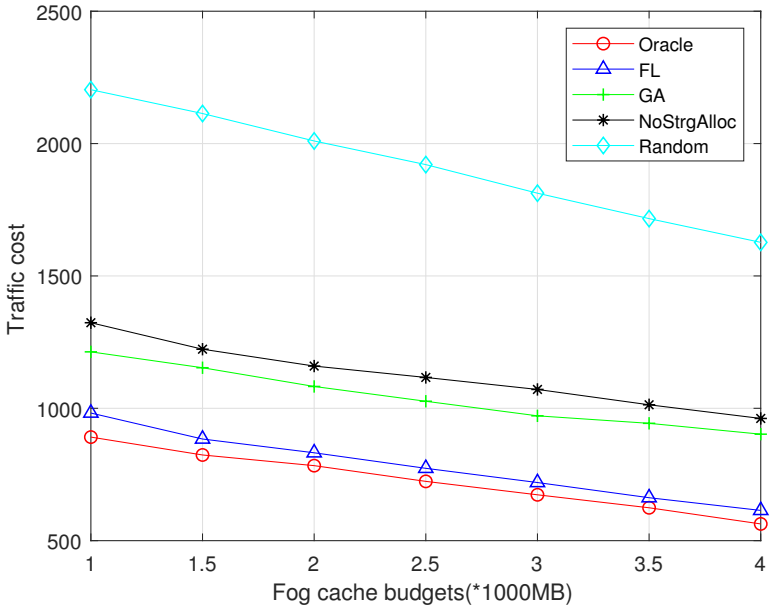

Figure 8. Traffic cost vs Fog cache budgets (MovieLens 1M).

Figures 9 and 10 show the relationship between traffic costs and the number of F-APs. Both figures show that as the number of F-APs increases (from 18 to 30), the cost of traffic

will decrease. Because the more F-APs there are, the more popular content can be cached in the local fog server, and the less traffic is injected into the network. The performance of our proposed FL and GA is superior to the existing random algorithm and no storage allocation algorithm. Moreover, the performance of FL algorithm is closer to the optimal one when the training dataset is 1M than that of 100K.

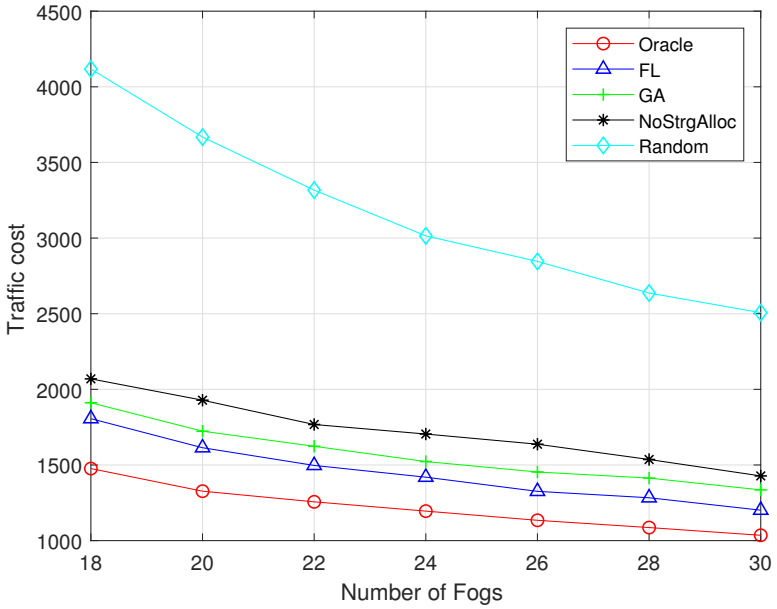

Figure 9. Traffic cost vs number of Fogs (MovieLens 100K).

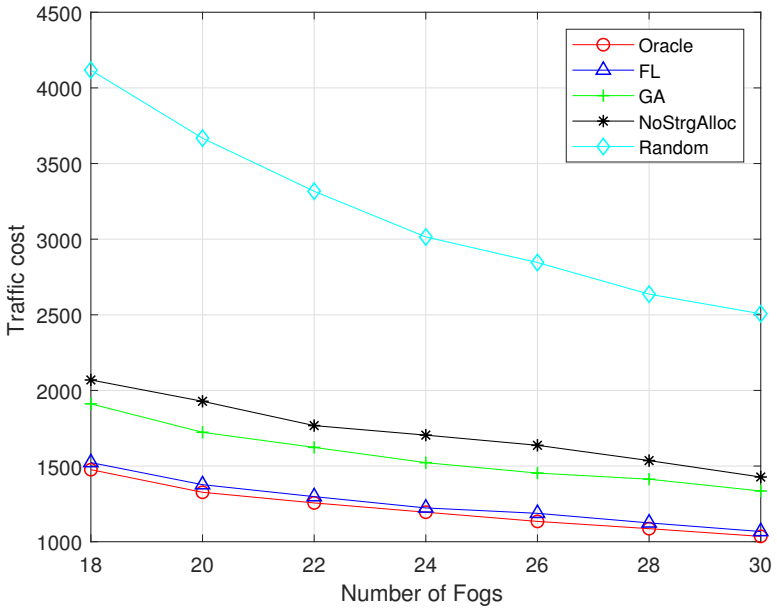

Figure 10. Traffic cost vs number of Fogs (MovieLens 1M).

As is discussed above, the simulation results based on the real-world datasets show that the more users and datasets participate in the training, the better the prediction effect of content popularity. Therefore, it can better realize the allocation of storage resources and the popular content placement in the network, thus reducing the network traffic cost effectively.

7. Conclusions and Future Work

In this paper, a federated learning-based intelligent F-RANs cache architecture is investigated. In the F-RAN architecture, we presented joint optimization of content caching and resource allocation to minimizing the traffic costs under F-APs storage budget constraints. In addition, considering the user's content request and privacy security, we adopt federated learning to make a distributed prediction of content popularity in different F-APs and apply it to the design of cache policy. The proposed caching scheme performs both efficient cache deployment and content caching. Due to the high computational complexity of the ILP model, and as the size of the problem increases, its scalability is not good. To reduce the computational complexity, two heuristic algorithms, that is greedy algorithm and FL algorithm, are introduced to provide approximate optimal solutions with lower computational complexity. Simulation results based on real-world datasets show that the proposed algorithm has better performance than existing algorithms and can obtain approximate optimal solutions.

Although the federated learning paradigm provides an efficient solution for implementing network edge smart caching in F-RANs, some key challenges remain. Due to the dynamic environment of network, mobile users may go offline or fall behind in the process of federated learning, which leads to poor accuracy of the training model. In future work, we will explore proactive content caching schemes based on fully asynchronous federated learning to better cope with highly dynamic network environments.

Author Contributions: Conceptualization, T.X.; Funding acquisition, T.C. and S.M.R.I.; Project administration, T.C.; Supervision, T.C.; Validation, T.X. and S.M.R.I.; Writing—original draft, T.X.; Writing—review & editing, Q.C. and T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Innovation Project of the Common Key Technology of Chongqing Science and Technology Industry (cstc2018jcyjAX0383), the special fund of Chongqing key laboratory (CSTC), and the Funding of CQUPT (A2016-83, GJJY19-2-23, A2020-270).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Smith, V.; Chiang, C.; Sanjabi, M.; Talwalkar, A.S. Federated multi-task learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
2. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018; pp. 63–71.
3. Fortino, G.; Rovella, A.; Russo, W.; Savaglio, C. Towards cyberphysical digital libraries: Integrating IoT smart objects into digital libraries. In *Management of Cyber Physical Objects in the Future Internet of Things*; Springer: Cham, Switzerland, 2016; pp. 135–156.
4. Chiang, M.; Zhang, T. Fog and IoT: An overview of research opportunities. *IEEE Internet Things J.* **2016**, *3*, 854–864. [CrossRef]
5. Peng, M.; Yan, S.; Zhang, K.; Wang, C. Fog-computing-based radio access networks: issues and challenges. *IEEE Netw.* **2016**, *30*, 46–53. [CrossRef]
6. Park, S.-H.; Simeone, O.; Shitz, S.S. Joint optimization of cloud and edge processing for fog radio access networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 7621–7632. [CrossRef]
7. Yu, Z.; Hu, J.; Min, G.; Lu, H.; Zhao, Z.; Wang, H.; Georgalas, N. Federated Learning Based Proactive Content Caching in Edge Computing. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018; pp. 1–6.
8. McMahan, H.B.; Moore, E.; Ramage, D.; y Arcas, B.A. Federated learning of deep networks using model averaging. *arXiv* **2016**, arXiv:1602.05629.
9. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *Acm Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]

10. Mcmahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
11. Wang, X.; Han, Y.; Wang, C.; Zhao, Q.; Chen, X.; Chen, M. In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. *IEEE Netw.* **2019**, *33*, 156–165. [CrossRef]
12. Asad, M.; Moustafa, A.; Yu, C. A Critical Evaluation of Privacy and Security Threats in Federated Learning. *Sensors* **2020**, *20*, 7182. [CrossRef]
13. Fantacci, R.; Picano, B. Federated learning framework for mobile edge computing networks. *Caai Trans. Intell. Technol.* **2020**, *5*, 15–21. [CrossRef]
14. Han, T.; Ansari, N. Network utility aware traffic load balancing in backhaul-constrained cache-enabled small cell networks with hybrid power supplies. *IEEE Trans. Mobile Comput.* **2017**, *16*, 2819–2832. [CrossRef]
15. Peng, M.; Zhang, K. Recent Advances in Fog Radio Access Networks: Performance Analysis and Radio Resource Allocation. *IEEE Access* **2016**, *4*, 5003–5009. [CrossRef]
16. Tandon, R.; Simeone, O. Cloud-aided wireless networks with edge caching: Fundamental latency trade-offs in fog radio access networks. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 10–15 July 2016; pp. 2029–2033.
17. Fan, X.; Zheng, H.; Jiang, R.; Zhang, J. Optimal Design of Hierarchical Cloud-Fog & Edge Computing Networks with Caching. *Sensors* **2020**, *20*, 1582.
18. Wang, X.; Leng, S.; Yang, K. Social-aware edge caching in fog radio access networks. *IEEE Access* **2017**, *5*, 8492–8501. [CrossRef]
19. Hung, S.; Hsu, H.; Lien, S.; Chen, K. Architecture Harmonization Between Cloud Radio Access Networks and Fog Networks. *IEEE Access* **2015**, *3*, 3019–3034. [CrossRef]
20. Aggarwal, C.; Wolf, J.L.; Yu, P.S. Caching on the world wide Web. *IEEE Trans. Knowl. Data Eng.* **1999**, *11*, 94–107. [CrossRef]
21. Ahlehagh, H.; Dey, S. Video caching in radio access network: Impact on delay and capacity. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC), Paris, France, 1–4 April 2012; pp. 2276–2281.
22. Wang, X.; Chen, M.; Taleb, T.; Ksentini, A.; Leung, V.C.M. Cache in the air: Exploiting content caching and delivery techniques for 5G systems. *IEEE Commun. Mag.* 2014, 52, 131–139. [CrossRef]
23. Müller, S.; Atan, O.; van der Schaar, M.; Klein, A. Smart caching in wireless small cell networks via contextual multi-armed bandits. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–7.
24. Song, J.; Sheng, M.; Quek, T.Q.S.; Xu, C.; Wang, X. Learning-Based Content Caching and Sharing for Wireless Networks. *IEEE Trans. Commun.* **2017**, *65*, 4309–4324. [CrossRef]
25. Jiang, W.; Feng, G.; Qin, S.; Yum, T.S.P.; Cao, G. Multi-Agent Reinforcement Learning for Efficient Content Caching in Mobile D2D Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 1610–1622. [CrossRef]
26. Muller, S.; Atan, O.; van der Schaar, M.; Klein, A. Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1024–1036. [CrossRef]
27. Abboud, A.; Baştuğ, E.; Hamidouche, K.; Debbah, M. Distributed caching in 5G networks: An Alternating Direction Method of Multipliers approach. In Proceedings of the 2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Stockholm, Sweden, 28 June–1 July 2015; pp. 171–175.
28. Hassine, N.B.; Milocco, R.; Minet, P. ARMA based popularity prediction for caching in Content Delivery Networks. In Proceedings of the 2017 Wireless Days, Porto, Portugal, 29–31 March 2017; pp. 113–120.
29. Bastug, E.; Bennis, M.; Debbah, M. Living on the edge: The role of proactive caching in 5g wireless networks. *IEEE Commun. Mag.* **2014**, *52*, 82–89. [CrossRef]
30. Hou, T.; Feng, G.; Qin, S.; Jiang, W. Proactive Content Caching by Exploiting Transfer Learning for Mobile Edge Computing. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
31. Tanzil, S.M.S.; Hoiles, W.; Krishnamurthy, V. Adaptive Scheme for Caching YouTube Content in a Cellular Network: Machine Learning Approach. *IEEE Access* **2017**, *5*, 5870–5881. [CrossRef]
32. Yu, Z.; Hu, J.; Min, G.; Zhao, Z.; Miao, W.; Hossain, M.S. Mobility-Aware Proactive Edge Caching for Connected Vehicles Using Federated Learning. *IEEE Trans. Intell. Transp. Syst.* **2020**. [CrossRef]
33. Wang, X.; Wang, C.; Li, X.; Leung, V.C.M.; Taleb, T. Federated Deep Reinforcement Learning for Internet of Things With Decentralized Cooperative Edge Caching. *IEEE Internet Things J.* **2020**, *7*, 9441–9455. [CrossRef]
34. Blasco, P.; Gündüz, D. Learning-based optimization of cache content in a small cell base station. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 1897–1903.
35. Zhang, J.; Hu, X.; Ning, Z.; Ngai, E.C.H.; Zhou, L.; Wei, J.; Cheng, J.; Hu, B.; Leung, V.C.M. Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching. *IEEE Internet Things J.* **2019**, *6*, 4283–4294. [CrossRef]
36. Golrezaei, N.; Shanmugam, K.; Dimakis, A.G.; Molisch, A.F.; Caire, G. FemtoCaching: Wireless video content delivery through distributed caching helpers. In Proceedings of the IEEE International Conference on Computer Communications, Orlando, FL, USA, 25–30 March 2012; pp. 1107–1115.
37. Chen, B.; Yang, C. Caching policy optimization for D2D communications by learning user preference. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 4–7 June 2017; pp. 1–7.
38. Leighton, T. Improving performance on the Internet. *Commun. ACM* **2009**, *52*, 44–51. [CrossRef]

39. Kellerer, H.; Pferschy, U.; Pisinger, D. *Knapsack Problems*; Springer: Heidelberg, Germany, 2004.
40. Li, X.; Wang, X.; Leung, V.C.M. Weighted network traffic offloading in cache-enabled heterogeneous networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6.
41. Harper, F.; Konstan, J. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 12. [CrossRef]
42. Li, S.; Xu, J.; van der Schaar, M.; Li, W. Popularity-driven content caching. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.