Device Selection and Data Offloading for Edge Caching in a FL Framework

1 Introduction

In this work, we aim to apply federated learning (FL) to the domain of content caching, where each device stores frequently requested content to reduce network load and latency. A key goal is to identify popular content across the network to optimize caching decisions. FL is used to collaboratively train a global model that predicts content popularity, with each device leveraging its local history of user requests [1].

To enhance the FL process, a system that combines intelligent device selection with data offloading is proposed. In each training round, only a subset of devices is selected based on their resource availability and the quality of their local datasets. Non-selected devices can offload part of their data to selected ones, improving data diversity and balance. This joint optimization improves both the accuracy of content popularity prediction and the overall efficiency of the FL framework.

2 System Model

Consider a set of N devices, $\mathcal{N}$, where each device $n \in \mathcal{N}$ has information about the content requested by the users in the nearby area. These content requests are converted into a binary request matrix $\bar{\mathbf{X}}_n \in \{0,1\}^{\bar{D}_n \times C}$, where $\bar{D}_n$ is the number of associated users and C is the total number of contents across devices.
The (i,j)th entry of $\bar{\mathbf{X}}_n$ is given by:

$$\bar{\mathbf{X}}_n(i,j) = \begin{cases} 1, & \text{if the } i\text{th user requested } j\text{th content in the past} \\ 0, & \text{if the content is either not requested or delivered by the device} \end{cases}$$

The total content request $\bar{z}_n$ on a device is $\sum_i \sum_j \bar{\mathbf{X}}_n(i,j)$.
A binary variable, s_n, represents whether a device is selected or not.

$$s_n = \begin{cases} 1, & \text{if device } n \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

The data is offloaded from a non-selected device to a selected device. The quantity of data samples at a device depends on the original data samples $\bar{D}_n$ and data samples

offloaded:

$$D_n = \bar{D}_n + (2s_n - 1) \begin{cases} \sum_{m \in \mathcal{N}} \phi_{m,n} \bar{D}_m, & s_n = 1 \\ \sum_{m \in \mathcal{N}} \phi_{n,m} \bar{D}_n, & s_n = 0 \end{cases}$$

where $\phi_{n,m}$ is the fraction of data samples offloaded from device n to m.

Similarly, the content request utilized during the training is given by:

$$z_n = \bar{z}_n + (2s_n - 1) \begin{cases} \sum_{m \in \mathcal{N}} \sum_{i=1}^{\phi_{m,n} \bar{D}_m} \bar{\mathbf{X}}_m^T(i)\mathbf{1} & s_n = 1 \\ \sum_{m \in \mathcal{N}} \sum_{i=1}^{\phi_{n,m} \bar{D}_n} \bar{\mathbf{X}}_n^T(i)\mathbf{1} & s_n = 0 \end{cases}$$

2.1 Latency and Energy Consumption

2.1.1 Global model download

The latency and energy consumption when a device downloads the updated global model from the central server are given by:

$$t_n^{\text{dn}} = \frac{w_S}{B_S^{\text{dn}} \log_2(1 + \gamma_n)}, \qquad e_n^{\text{dn}} = P_S t_n^{\text{dn}}$$

w_S	size of global machine learning model	B_s^{dn}	downlink bandwidth
γ_n	signal-to-noise ratio (SNR)	P_S	downlink transmission power

2.1.2 Local model update

The latency and energy consumption in local training using the device's data are:

$$t_n^{\text{comp}} = \frac{\mu_n C_n D_n}{f_n}, \qquad e_n^{\text{comp}} = \beta_n \mu_n C_n D_n f_n^2$$

μ_n	number of local epochs	C_n	number of CPU cycles to process a sample
D_n	number of data samples	f_n	CPU-cycle frequency
β_n	effective switched capacitance		

2.1.3 Local model upload

After training, the device sends its local model updates back to the central server. The latency and energy consumption in uploading the local model are given by:

$$t_n^{\text{up}} = \frac{w_n}{B_n^{\text{up}} \log_2(1 + \gamma_n)}, \qquad e_n^{\text{up}} = P_n t_n^{\text{up}}$$

w_n	size of local machine learning model
B_n^{up}	uplink bandwidth
P_n	downlink transmission power

The latency and energy of each device are given by:

$$t_n = t_n^{\text{dn}} + t_n^{\text{comp}} + t_n^{\text{up}}, \qquad e_n = e_n^{\text{dn}} + e_n^{\text{comp}} + e_n^{\text{up}}$$

The latency of the FL framework is decided by the slowest selected devices and is given by:

$$t = \max_{n \in \mathcal{N}} \{s_n t_n\}$$

2.2 Dataset Similarity

To assess the suitability of the device for participation in FL, a cosine similarity measure is used. The cosine similarity is defined as:

$$\rho_n = \frac{1}{D_n^2} \sum_{\mathbf{y}_i, \mathbf{y}_j \in X_n} \frac{\mathbf{y}_i^T \mathbf{y}_j}{\|\mathbf{y}_i\| \|\mathbf{y}_j\|}$$

where $X_n = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \dots & \mathbf{y}_{D_n}^T \end{bmatrix}^T$ and $\mathbf{y}_i$ is the request vector of user i.
Devices with higher cosine similarity indicate more consistent data and fewer outliers, enhancing the quality of local training.

3 Problem Formulation

The problem formulation aims to optimize FL performance by intelligently selecting a subset of devices and determining how much data to offload from non-selected to selected devices. The goal is to maximize the combined effect of content requests and data quality (similarity) on selected devices while addressing the latency, energy consumption, data volume, and system capacity.

$$\max_{\mathbf{s}, \mathbf{\Phi}} \sum_{n \in \mathcal{N}} s_n z_n \rho_n \tag{1}$$

$$\begin{aligned}
\text{subject to, } & s_n \in \{0,1\} && \forall n \in \mathcal{N} && \\
& \phi_{n,m} \in [0,1] && \forall n, m \in \mathcal{N} && \text{(1a)} \\
& s_n z_n + (1 - s_n) z_{\min} \geq z_{\min} && \forall n \in \mathcal{N} && \text{(1b)} \\
& (1 - s_n) z_n + \alpha_n t \leq z_{\max} && n \in \mathcal{N} && \text{(1c)} \\
& s_n D_n \leq D_{\max} && \forall n \in \mathcal{N} && \text{(1d)} \\
& t \leq t_{\max} && && \text{(1e)} \\
& s_n e_n \leq e_{\max} && \forall n \in \mathcal{N} && \text{(1f)} \\
& S_{\min} \leq \sum_{n \in \mathcal{N}} s_n \leq S_{\max} && && \text{(1g)} \\
& \sum_{m \in \mathcal{N}} \phi_{m,n} \leq 1 && \forall n \in \mathcal{N} && \text{(1h)} \\
& (1 - s_n)(1 - s_m) \phi_{n,m} = 0 && \forall n, m \in \mathcal{N} && \text{(1i)} \\
& s_n \phi_{n,m} = 0 && \forall n, m \in \mathcal{N} && \text{(1j)}
\end{aligned}$$

α_n data acquisition rate

(1) Maximize content request and data similarity over selected devices.
(1a) Range of decision variables s_n and $\phi_{n,m}$.
(1b) Ensures that the device is selected only if it has a minimum $z_{\min}$ number of content requests.
(1c) Constraints the number of requests a device can store.
The total content request is the sum of contents on the device and the number of contents it will accumulate over time while the current federated learning round is going on.
(1d) Limits the total number of data samples a selected device can process.
(1e) Latency of the FL round must be below a predefined threshold.
(1f) Selected devices must operate within their energy budget during each training round.
(1g) Limits the number of devices selected.
(1h) Prevents the device from offloading data more than it contains.
(1i) Ensures that two non-selected devices do not offload data to each other.
(1j) Prohibits selected devices from offloading data to other devices.

3.1 Datasets

MovieLens 1M

3.2 Simulation Parameters

N	10	w_S	75 MB	B_S^{dn}	150 MHz
P_S^{dn}	1 W	μ_n	1	C_n	$[40, 60]$
D_n	$[20, 50]$	f_n	10^9	β_n	10^{-27}
w_n	$[50, 100]$ MB	B_n^{up}	100 MHz	P_n	$[0.01, 0.05]$ W
α_n	$[18, 20.5] \times 10^5$	$z_{\min}$	800	$z_{\max}$	7000
$D_{\max}$	40	$t_{\max}$	1 ms	$e_{\max}$	1 mJ
$S_{\min}$	2	$S_{\max}$	$5, 10$		

References

1. Kushwaha, Deepali, et al. "Device Selection for Resource-Efficient Edge Caching in a Federated Learning Framework." ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2025.