IEEE International Conference on Communications (ICC)

Energy Efficient User Scheduling for Hybrid Split and Federated Learning in Wireless UAV Networks

Authors:
Xiaolan Liu
Yansha Deng,
Toktam Mahmoodi

Hybrid Split and Federated Learning in Wireless UAV Networks

Wireless networks are evolving to include UAVs as essential nodes for data collection and processing, serving a wide range of applications from surveillance to disaster response.

Key Challenges in UAV Networks:

- High communication overhead in traditional learning methods.

- Energy constraints due to limited battery life of UAVs.

- Privacy concerns with transmitting large datasets.

- Variability and unreliability of wireless channels.

- Heterogeneous computational capabilities across UAVs.

- Efficient user scheduling and resource optimization.

Proposed Model

We propose a **Hybrid Split and Federated Learning (HSFL)** framework that allows users to select either **Split Training (ST)** or **Federated Training (FT)** method based on the characteristics of the users in wireless UAV networks.

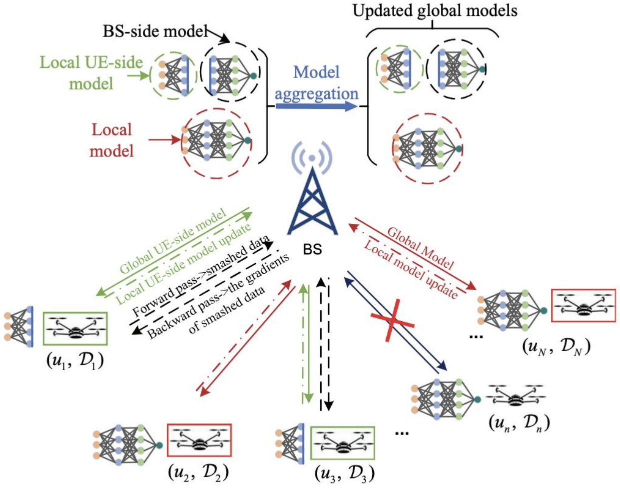


Fig. 1. The system model

Objectives of this paper

- Energy Efficiency
- Resource Optimization
- Communication Overhead
- User Scheduling
- Learning Under Unreliable Wireless Channels

Background

Previously we explored **Radio Resource Management** (**RRM**) using **Federated Edge Learning (FEEL)** to optimally allocate and manage radio spectrum resources—like bandwidth and transmitting power—among multiple edge devices.

The proposed strategy used **Federated Learning**:

- To collaboratively learn a shared model
- Minimize energy consumption and prolong battery life of edge devices
- Prioritize energy efficiency without compromising learning rate.

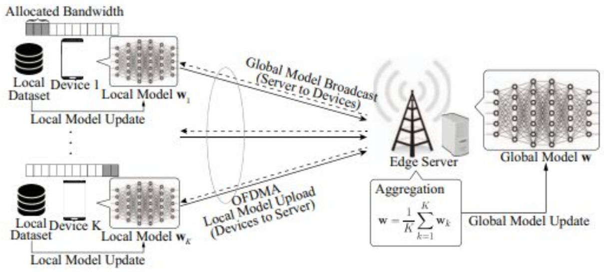


Figure 1. A framework for FEEL system.

System Model (Learning Model)

We adopt a DNN model in the wireless UAV network. The users and the Base Station collaboratively train an ML model to minimize a global loss function $F(\omega)$.

$$N = \{u_1, \ldots, u_N\}$$

Where each u_i is an Unmanned Aerial Vehicle (UAV), and each u_i has a dataset D_i.

$$\min_{\omega} \; F(\omega) \triangleq \sum_{i=1}^{N} \frac{D_i}{D} F_i(\omega), \quad D = \sum_{i=1}^{N} D_i \qquad (1)$$

System Model (Learning Model) (Cont.)

We adopt the model splitting structure of the SL by dividing the considered DNN model into two sub-models and distributing

---

Algorithm 1 Wireless HSFL Algorithm

Initialization The BS initializes global model ω_t, global UE-side model ω_t^l and global BS-side model ω_t^e, set $t = 0$.

Repeat

Each user sends their characteristic information to the BS

The BS selects a subset of users $\mathcal{K}$, and schedules each selected user with the ST or FT method.

The BS distributes ω_t to the users $u_i \in \mathcal{K}_F$ with FT method, distributes ω_t^l to the users $u_i \in \mathcal{K}_S$ with ST method and at the same time assign the BS with ω_t^e.

for $u_i \in \mathcal{K}$ in parallel **do**

    if $u_i \in \mathcal{K}_F$ **then**

        user computes local model updates with FT method independently.

    else if $u_i \in \mathcal{K}_S$ **then**

        user computes local model updates collaboratively with the BS using ST method.

    end if

end for

The BS performs model aggregation with the weighted average technique of FedAvg [4].

Set $t = t + 1$.

Until the desired convergence performance is achieved or the final iteration arrives

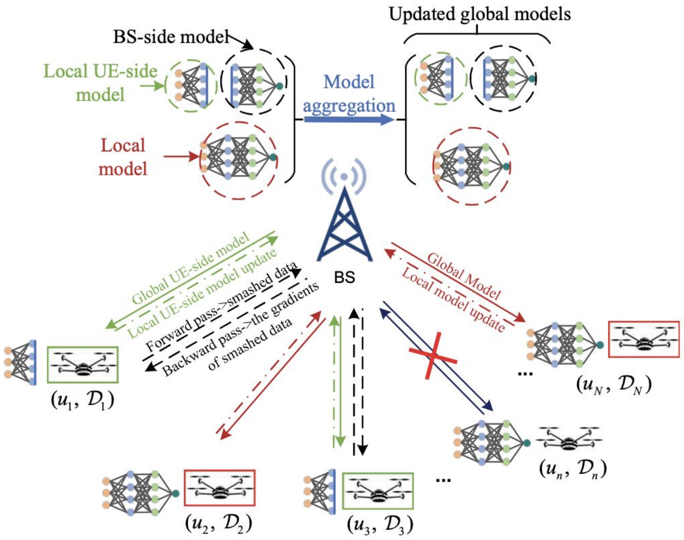


Fig. 1. The system model

System Model (Computation Model)

FT Method:

The computation time of user ui computing its local model updates is:

$$\tau_{iF}^{tr} = \frac{e_i C_{iF} D_i}{f_i}, \forall i \in K_F$$

- τ_{iF}^{tr}: The computation time of user u_i computing its local model updates.
- e_i: The number of local training iterations at user u_i.
- C_{iF}: The number of CPU cycles required for computing one sample data at user u_i.
- D_i: The dataset of user u_i.
- f_i: The computation capacity of user u_i, measured in CPU cycles per second.
- K_F: The set of users scheduled with the FT method.

The energy consumption of computing its local model updates is:

$$E_{iF} = k e_i C_{iF} D_i f_i^2$$

- E_{iF}: The energy consumption of computing local model updates at user u_i.
- k: The effective switched capacitance that depends on the chip architecture.

System Model (Computation Model) (Cont.)

ST Method:

The computation time of user ui computing its local model updates is:

$$\tau_{iS_l}^{tr} = \frac{e_i C_{iS} D_i}{f_i}, \quad \forall i \in K_S$$

$$\tau_{iS_e}^{tr} = \frac{e_i C_{iB} D_i}{f}, \quad \forall i \in K_S$$

Where:

- $\tau_{iS_l}^{tr}$ and $\tau_{iS_e}^{tr}$ are the computation times for the UE-side and BS-side model updates, respectively.
- e_i is the number of local training iterations for user u_i.
- C_{iS} and C_{iB} are the CPU cycles required for computing one sample of data at the user equipment (UE) and base station (BS), respectively.
- D_i is the dataset of user u_i.
- f_i is the computation capacity of the user equipment (UE), and f is that of the base station (BS).
- K_S is the set of users scheduled with the ST method.

The computation time of user ui computing its local model updates is:

$$E_{iS} = k e_i C_{iS} D_i f_i^2$$

- E_{iS}: The energy consumption for computing the local model updates of the UE-side model at user u_i.
- k: The effective switched capacitance that depends on the chip architecture.

System Model (Transmission Model)

We assume all the users transmit their model parameters to the BS via frequency domain multiple access (FDMA) scheme.

$$r_i = b_i B_w \log_2 \left(1 + \frac{g_i P_i}{N_0 b_i B_w}\right), \quad \forall i \in \mathcal{N}$$

Where:

- r_i: The uplink transmission rate from the user u_i to the base station (BS).
- b_i: The ratio of allocated bandwidth to user u_i.
- B_w: The total bandwidth.
- g_i: The channel gain between user u_i and the BS.
- P_i: The transmit power of user u_i.
- N_0: The power spectral density of the Gaussian noise.
- $\mathcal{N}$: The set of all users.
- $\mathcal{K}$: The selected subset of users, which must satisfy $\sum_{i \in \mathcal{K}} b_i \leq 1$ due to the limited system bandwidth.

System Model (Transmission Model) (Cont.)

For the FT method:

Transmission time: $\tau_{iF}^{ul} = \frac{m_i^g}{r_i}$

- This represents the transmission time for uploading the local model updates to the BS.
- m_i^g is the model size for user u_i, which determines the communication overhead.
- r_i is the uplink transmission rate from the user u_i to the BS.
- Due to high transmit power at the BS and substantial available bandwidth for data broadcast, downlink transmission time is considered negligible.

For the ST method:

Transmission time: $\tau_{iS}^{ul} = \frac{m_i^l + m_i^a}{r_i}$

- Here, the user uploads both the output activations of the cut layer and the local model updates to the BS.
- m_i^l indicates the size of the user-side model.
- m_i^a represents the size of the activations and is a product of a constant a and the local dataset size $|D_i|$.
- r_i remains the transmission rate from the user to the BS.

System Model (Transmission Model) (Cont.)

For the downlink transmission:

Downlink rate: $r_s = B_s \log_2(1 + \frac{P_s g}{N_s B_s})$

- r_s: Achievable downlink data rate.
- B_s: Bandwidth allocated to each UE.
- P_s: Transmit power for downlink.
- g: Channel gain for downlink.
- N_s: Noise power spectral density.
- Downlink transmission time: $\tau_{iS}^{dl} = \frac{m_i^g}{r_s}$.

Latency of the HSFL algorithm:

Total Latency: $T = \max(\alpha_{iF}\tau_{iF} + \alpha_{iS}\tau_{iS})$

- $\tau_{iF} = \tau_{iF}^{tr} + \tau_{iF'}^{ul}$, for $i \in K_F$.
- $\tau_{iS} = \tau_{iS}^{tr} + \tau_{iS}^{ul} + \tau_{iS}^{dl}$, for $i \in K_S$.
- α_{iF} and α_{iS} are indicators that equal 1 if user u_i is scheduled with the FT or ST method, respectively; they are 0 if the user is not selected in this round.

System Model (Diversity Index)

The Diversity Index (I_i) for a given user (u_i) is calculated as a weighted sum of four parameters, representing different aspects of user and data diversity:

- Local Dataset Diversity
- Age-of-Update
- Computation Capacity
- Local Dataset Size

The Diversity Index for a user u_i is calculated as:

$$I_i = \sum_n v_{i,n}\gamma_{i,n}$$

Where:

- $v_{i,n}$ are the normalized values for each characteristic of user diversity.
- $\gamma_{i,n}$ are the weights assigned to each characteristic.

The weights are collectively represented as:

$$\Phi = \{\gamma_{i,1}, \ldots, \gamma_{i,n}\}$$

This index could be used to make strategic decisions in a system, such as prioritizing users for updates, allocating bandwidth, or distributing computational tasks in a federated learning scenario.

Problem Formulation

- **Objective**: To minimize the total energy consumption of users while maximizing user diversity in a wireless network, under a specific latency constraint.
- **Energy and Diversity**: The problem considers both the computation and communication energy of the users, promoting diversity among the scheduled users.
- **Multiple-Choice Knapsack Problem (MCKP):** The user scheduling problem is modeled as an MCKP, where each user selects a training method, optimizing for energy efficiency and diversity.

Constraints

Constraints:

- Each user's training method selection and bandwidth allocation must not exceed the predefined latency and bandwidth limits.
- Each user can only be scheduled with one training method at a time.
- Binary decision-making is used for selecting the training method for each user.

Solution Approach: Due to the complexity of the problem, a linear relaxation and a greedy algorithm are proposed for finding a near-optimal solution.

Initial Problem (OP1)

$$
\begin{aligned}
\mathcal{OP}_1 \quad & \min_{b_i,\, \{\alpha_{ij}\}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij}(E_{ij} - I_{ij}) \\
s.t. \quad & C_1 : \sum_{j \in \mathcal{J}} \alpha_{ij}\tau_{ij} \leq T, \\
& C_2 : \alpha_{ij} \in \{0, 1\},\ \forall j \in \mathcal{J},\ \forall i \in \mathcal{N}, \\
& C_3 : \sum_{j \in \mathcal{J}} \alpha_{ij} \leq 1,\ \forall i \in \mathcal{N}, \\
& C_4 : \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij} b_i \leq 1, \\
& C_5 : 0 \leq b_i \leq 1,\ \forall i \in \mathcal{N},
\end{aligned}
$$

Continued

- **Constraints**:
  - $C1$: The total computation time must not exceed the maximum round latency T.
  - $C2$: α_{ij} are binary decision variables.
  - $C3$: Each user can only choose one training method.
  - $C4$: The sum of the bandwidth allocated to all users must not exceed the total available bandwidth.
  - $C5$: Bandwidth allocation ratios are between 0 and 1.

Transformation to OP2 (Standard MCKP Problem):

$$\mathcal{OP}_2 \quad \min_{\{\alpha_{ij}\}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij}(E_{ij} - I_{ij})$$

$$s.t. \;\; C_2, \; C_3, \; C_4, \; C_5$$

$$\sum_{j \in \mathcal{J}} \alpha_{ij}\tau_{ij} \le T, \quad \forall i \in \mathcal{N}$$

Continued

- **Simplification**: The time constraint $C1$ is initially ignored to simplify the problem.
- **Objective**: Same as OP_1, but without the time constraint.
- **Constraints**: Same as OP_1, excluding $C1$.

Final Problem OP3 (Simplified MCKP Problem)

$$\mathcal{OP}_3 \quad \max_{\{\alpha_{ij}\}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij} p_{ij}$$

$$s.t. \quad C_2, \; C_4,$$

$$C_3 : \sum_{j \in \mathcal{J}} \alpha_{ij} = 1, \; \forall i \in \mathcal{N}$$

$$0 \leq \alpha_{ij} \leq 1, \; i \in \mathcal{N}, \; j \in \mathcal{J}.$$

Continued

- **Maximization Objective:** Instead of minimization, the problem is reframed to maximize the profit of selecting training methods.
- **Profit Definition:** $p_{ij} = C + (I_{ij} - E_{ij})$, where C is a large constant ensuring profits are non-negative.
- **Relaxed Decision Variables:** α_{ij} can now take any value between 0 and 1 instead of being binary, making the problem linear (LMCKP).
- **Constraints**: $C2$ and $C4$ remain the same, $C3$ ensures that each user can be assigned to only one training method or not be selected at all (due to the inclusion of $m_j = 0$ as a possible 'non-selection' method).

Solution

Then, the linear **MCKP (LMCKP)** problem can be solved in O($nlogn$) time through the following two steps:

1. Remove the dominated training mechanisms to reduce the scale of the given **LMCKP.**

2. Apply the greedy algorithm.

Removing Dominated Training Mechanisms

- **Efficiency Assessment**: Evaluate training methods based on a profit-to-weight ratio to identify efficiency.
- **Dominance Criteria**: A training method is dominated and can be removed if there is another method that offers higher profit for the same or lower weight.
- **Cost-Benefit Analysis**: Compare the cost (weight) and benefit (profit) of all available training methods for each user.
- **Simplification Process**: Eliminate dominated methods to reduce complexity and focus on the most efficient options.
- **Preparation for Greedy Algorithm**: After removing less efficient methods, prepare a refined set of training options for the subsequent greedy selection process.

Continued

In LMCKP, the training method m_{is} is considered to be dominated by the training method m_{ir} and m_{it} if the three methods satisfy the following conditions:

$$w_{ir} \leq w_{is} \leq w_{it}, \; p_{ir} \leq p_{is} \leq p_{it}, \tag{11a}$$

$$\lambda_{i,r\to s} \leq \lambda_{i,r\to t}, \tag{11b}$$

$$\lambda_{i,r\to t} = \frac{p_{it} - p_{ir}}{w_{it} - w_{ir}}, \; \lambda_{i,r\to s} = \frac{p_{is} - p_{ir}}{w_{is} - w_{ir}}, \tag{11c}$$

Where $\lambda_{i,r\to t}$ is defined as the update efficiency. Here, (11b) means switching from m_{ir} to m_{is} is less efficient than switching from m_{ir} to m_{it}

The Greedy Algorithm

Algorithm 2 LMCKP-greedy User Scheduling Algorithm

Initialization:

Input: The diversity index $I_i, i \in \mathcal{N}$, energy consumption of each learning mode $E_{ij}, i \in \mathcal{N}, j \in \mathcal{J}$

Set $T_{th} = 10$

Learning:

1: **for** $t \leq T$ **do**

2: The profit is calculated as $\mathbf{p}_{ij}$, the weight matrix is initialized as $b_{ij} = \frac{1}{N}$, if $j = \{S, F\}$, otherwise $b_{ij} = 0$.

3: $\mathcal{N}_{candi}$ = ***LMCKPgreedy*** *(profit, weight)*

4: **for** $u_i \in \mathcal{N}_{candi}$ **do**

5: **if** $u_i \in \mathcal{K}_F$ and $\tau_i \leq T_{th}$ **then**

6: update local model updates with FL method.

7: **else if** $u_i \in \mathcal{K}_S$ and $\tau_i \leq T_{th}$ **then**

8: update local model updates with SL method.

9: **end if**

10: **end for**

11: The BS perfroms model aggregation with the received local model updates.

12: **end for**

LMCKPgreedy:

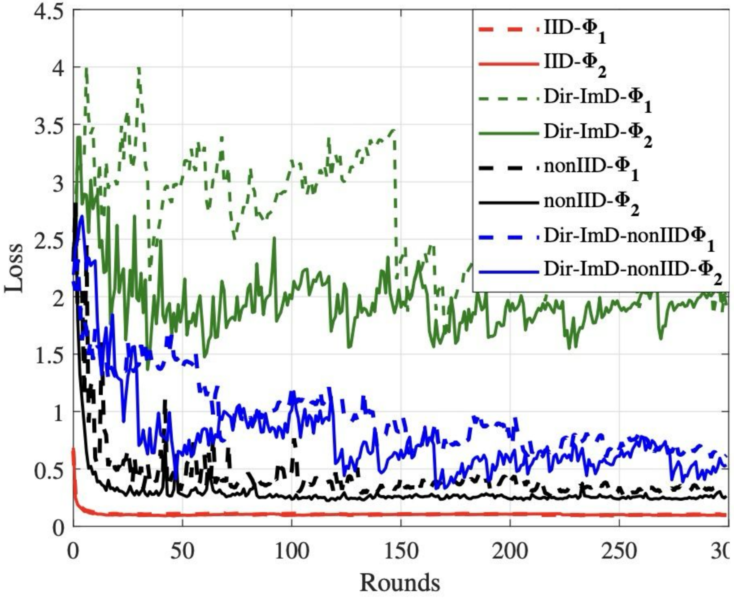
13: Remove dominated training mechanisms according to (11), set $\alpha_{ij} = 1$

14: Sort $\lambda_{i,r \to t}$ in non-decreasing order, and update the selection variable α_{ij} for each training mechanism; then update $b = b - b_{ij} + b_{ik}$; repeat until violates the constraint C_4.

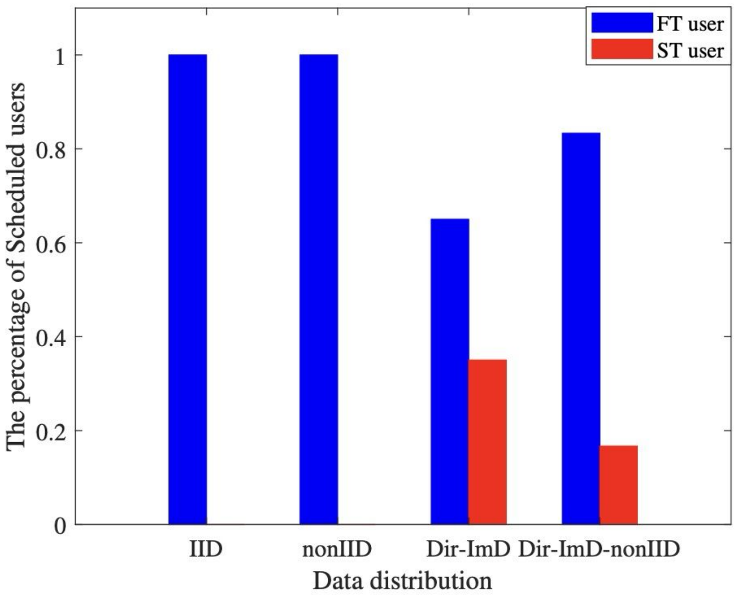
15: Obtain the MCKP feasible solution $\mathcal{N}_{candi}$.

16: **Return** $\mathcal{N}_{candi}$.

Simulation Results

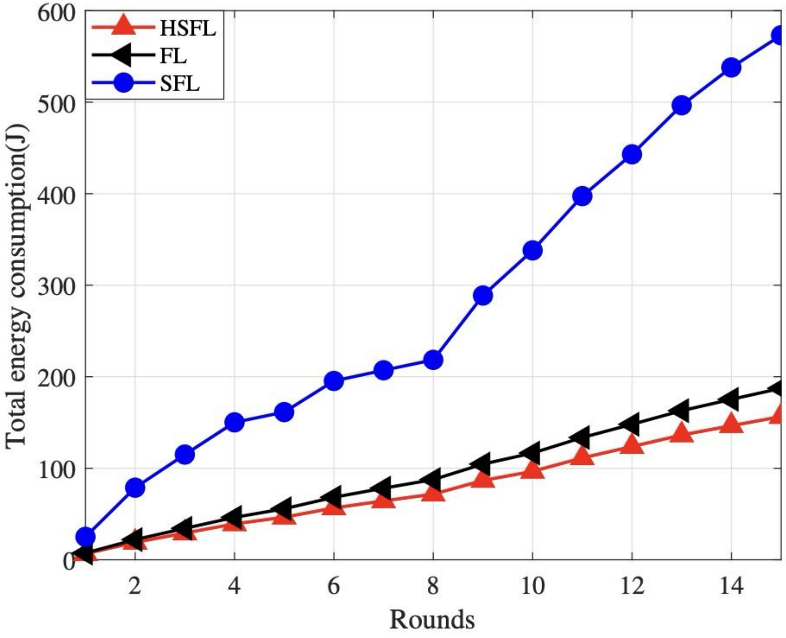


(a) Convergence performance

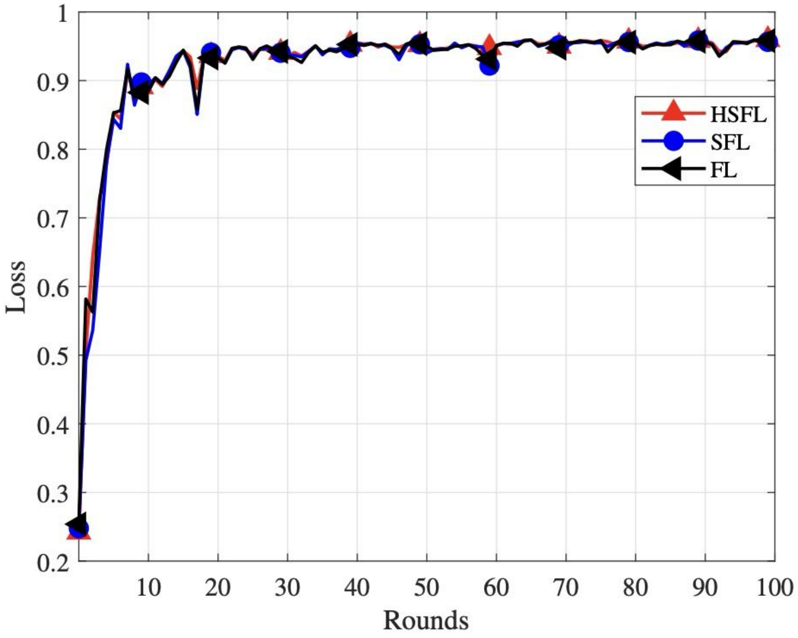


(b) Scheduled user percentage

Continued



(c) Energy consumption comparison



(d) Test accuracy

Thank You