

Article

Federated Learning over 5G/6G Networks: Dynamic Client Selection and Resource Allocation for Heterogeneous Edge Environments

Ahmed Lateef Salih Al-Karawi ¹ and Rafet Akdeniz ^{2,*}¹ Defne Telekomünikasyon A.Ş., 34485 Sarıyer, Türkiye; ahmedal-karawi@stu.aydin.edu.tr² Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Atlas University, Anadolu Caddesi No:40, 34408 Kağıthane, Türkiye

* Correspondence: rafet.akdeniz@atlas.edu.tr; Tel.: +90-5423-427-501

Abstract

Federated learning (FL) has emerged as a promising paradigm for privacy-preserving edge intelligence because it enables geographically distributed devices to collaboratively train a shared model without transferring raw data to a central cloud. This capability is particularly valuable for 5G and emerging 6G networks, where edge-native services are required to satisfy stringent latency, bandwidth, and privacy constraints while operating on highly heterogeneous devices and time-varying wireless channels. In practice, however, synchronous FL is often constrained by straggling clients with limited computation capability or unfavorable communication conditions, which increases round latency and reduces overall resource efficiency. To address this challenge, this study develops a rigorously structured framework for dynamic client selection and radio resource allocation in heterogeneous wireless edge environments. Each FL round is formulated as a latency-aware scheduling problem that jointly captures local computation time, uplink transmission time, minimum participation constraints, and resource block assignment. On this basis, we propose a Dynamic Client Selection and Resource Allocation (DCS-RA) method that integrates computation-aware, channel-aware, and fairness-aware scoring with greedy resource block allocation guided by marginal completion time reduction. The study further provides a clear methodological structure, workflow visualization, literature-grounded justification, dataset documentation, and uncertainty-aware result reporting. Under the reported simulation setting with 100 clients and 20 resource blocks, DCS-RA reduces the average round completion time from 1.92 s to 1.55 s on MNIST and from 2.02 s to 1.57 s on CIFAR-10, corresponding to improvements of 19.39% and 22.47%, respectively. Standard deviation reductions of 70.59% and 80.77% further indicate improved round-to-round stability and more reliable training behavior. These results support the central conclusion that lightweight joint scheduling can materially improve wall-clock FL efficiency in heterogeneous 5G/6G edge networks.



Academic Editor: Luis Alonso

Received: 28 April 2026

Revised: 27 June 2026

Accepted: 29 June 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: federated learning; client selection; resource allocation; edge intelligence; 5G; 6G; heterogeneous networks; wireless edge computing

1. Introduction

The migration of machine intelligence from centralized cloud infrastructures to the network edge is one of the most consequential architectural shifts shaping 5G and anticipated 6G systems. In this context, edge-native intelligence is no longer merely an

optimization option; rather, it is increasingly a functional requirement for latency-critical and privacy-sensitive services such as industrial sensing, vehicular analytics, mobile health, and smart city monitoring [1,2]. Contemporary survey studies have consistently identified the principal drivers of this transition: edge computing shortens the control loop, reduces backhaul congestion, and improves data locality, while FL enables collaborative model training without exposing raw user data [3,4]. Collectively, these characteristics make FL particularly well suited to future wireless environments that must simultaneously support massive connectivity, stringent delay requirements, and highly heterogeneous computing resources [5].

Although the proposed scheduling logic is applicable to both 5G and 6G-oriented edge systems, adoption is not identical across generations [6]. In a 5G deployment, DCS-RA would mainly operate above the existing base station or MEC control plane by using available channel quality, computation, and participation metadata to reduce synchronous FL stragglers. In a 6G-oriented deployment, the same scoring and allocation principle could be extended with richer sensing, AI-native orchestration, finer-grained radio resources, and predictive mobility or channel information. Therefore, the method does not depend on a 6G-only capability, but the quality, frequency, and granularity of the available network state information can affect how effectively it is deployed.

A concrete deployment scenario is autonomous vehicular edge learning, where vehicles and roadside units act as FL clients while an edge controller associated with a base station coordinates training. In such a scenario, client computation differs because vehicles may use different embedded processors, sensor suites, battery states, and background workloads. Wireless quality also varies because of mobility, blockages, handovers, interference, and distance from the serving infrastructure. These conditions create practical FL stragglers: some vehicles may complete local training quickly but upload slowly, while others may have good channels but limited computation resources. Dynamic client selection and resource allocation are therefore needed to prevent a small subset of weak or temporarily unfavorable clients from determining the duration of every synchronous round.

Recent studies further show that FL in 5G/6G environments is increasingly discussed together with secure network design, edge computing, blockchain-enabled trust, and Internet of Medical Things applications, rather than as an isolated learning algorithm [7,8]. This broader framing is important for the present study because DCS-RA is evaluated as a network efficiency mechanism operating under constrained communication and computation resources, not as a replacement for cryptographic, trust management, or privacy-preserving protocol layers.

Despite this strong conceptual alignment, practical FL over wireless edge networks remains difficult to realize efficiently. The central challenge is that FL operates over clients with non-uniform CPU frequency, local dataset size, battery budget, and radio quality. In synchronous training, the server must wait for all selected participants before aggregation can proceed. As a result, even a single weak client may determine the duration of the entire communication round. Recent surveys focused on resource-constrained IoT systems and client-side FL limitations show that device heterogeneity [9], communication inefficiency, fairness, and client availability continue to be major barriers to real-world deployment [10,11]. Thus, privacy preservation alone is insufficient for practical adoption; FL must also be effectively schedulable under realistic edge constraints [12,13].

This observation makes two decisions fundamental to overall system performance: The first is client selection, namely, determining which clients should participate in a given training round. The second is resource allocation, namely, deciding how limited communication resources should be assigned to the selected set. The recent survey literature on device and client selection shows that random participation becomes inadequate once

learning systems operate under heterogeneous radio conditions, non-IID data distributions, and resource asymmetry [12,14]. Consequently, the scheduling controller must account for both computation and communication conditions, and must do so with sufficiently low complexity to remain practical for round-by-round operation [15].

Several recent studies have proposed joint or hierarchical scheduling methods for wireless FL, demonstrating that latency [16] and energy consumption can be reduced when client participation is coordinated with resource assignment [17–19]. However, the literature also reveals an important practical gap: Many existing formulations pursue high performance through complex decomposition methods, hierarchical orchestration, or learning-based control. Although such strategies are scientifically valuable, they are not always attractive when a controller must make repeated round-level decisions under volatile edge conditions. A useful complementary direction is therefore the development of a method that remains analytically motivated, computationally lightweight, and directly deployable in realistic wireless edge environments.

The novelty claimed in this paper is therefore deliberately scoped: DCS-RA does not claim to be the first joint scheduling framework for wireless FL; its contribution is the integration of resource-aware client scoring, fairness-aware participation pressure, and marginal latency-based RB allocation into a single interpretable heuristic that can be executed repeatedly with low scheduling overhead. This positioning distinguishes the proposed method from heavier optimization-based joint schedulers and from client selection-only approaches while also avoiding an unsupported claim of universal superiority.

Against this background, this paper describes a Dynamic Client Selection and Resource Allocation (DCS-RA) framework for heterogeneous 5G/6G edge environments. The framework is structured around a latency-aware system model, an interpretable client-scoring rule, a greedy RB allocation mechanism, and an uncertainty-aware evaluation of the reported round completion times. The main contributions are as follows:

- We formulate FL scheduling in a heterogeneous wireless edge system by jointly modeling local computation latency, uplink transmission latency, minimum client participation, and resource block assignment constraints.
- We propose DCS-RA, a structured methodology that combines computation-aware, channel-aware, and fairness-aware client scoring with greedy resource allocation based on marginal round latency reduction. The method is presented as a deployment-oriented heuristic rather than a globally optimal solver.
- We evaluate DCS-RA on MNIST and CIFAR-10 based on round completion latency, latency dispersion, cumulative round behavior, and standard deviation-aware summary statistics. The evaluation focuses on scheduling efficiency and latency stability, and accuracy/loss convergence, explicit non-IID partition analysis, and stronger baseline benchmarking are treated as limitations and future validation directions.

The remainder of the paper is organized as follows: Section 2 synthesizes the most relevant review and research literature and defines the research gap. Section 3 presents the system model and optimization problem. Section 4 describes the proposed DCS-RA workflow and algorithm. Section 5 details the experimental design, datasets, and evaluation metrics. Section 6 reports the results and provides scientific interpretation. Section 7 concludes the paper.

2. Related Work and Research Gap

2.1. What the Review Literature Has Already Established

The broader FL literature has already established a strong consensus regarding the importance of edge-aware collaborative learning. Comprehensive surveys published in *IEEE*

Communications Surveys & Tutorials, *Information Processing & Management*, *High-Confidence Computing*, *Applied Sciences*, and *Sensors* consistently show that FL is increasingly regarded as a system-level enabler of privacy-preserving analytics in IoT and edge computing environments. At the same time, these studies also make clear that communication cost, device heterogeneity, and orchestration complexity remain fundamental barriers to practical deployment [2–5,9,10]. For 6G-oriented environments, Duan et al. further argue that ubiquitous intelligence will depend on the joint evolution of FL and edge computing, particularly in scenarios characterized by dense device participation and latency-sensitive services [1]. Complementary review studies on future wireless edge infrastructure similarly emphasize that 6G services will not rely on static cloud offloading models, but rather will require dynamic coordination between communication and computation resources [6].

A second stream of the review literature has focused specifically on the participation problem. Surveys published in *Computer Networks*, *SN Computer Science*, and *Artificial Intelligence Review* show that client or device selection is no longer a secondary implementation detail; instead, it directly influences convergence speed, statistical bias, fairness, and operational cost [12,14]. Recent IoT-centered review work has also emphasized that fairness and resource-aware participation are especially important in heterogeneous settings [11], where weak devices may otherwise be excluded from repeated training rounds [13]. Taken together, the review literature supports two robust conclusions: First, joint scheduling is a central requirement for efficient wireless FL. Second, lightweight yet principled client selection mechanisms remain underdeveloped relative to their practical significance.

2.2. Representative Joint Scheduling Studies

The recent research literature confirms the need for tighter coupling between participation decisions and resource control. Yu et al. jointly optimized client selection and resource management for wireless IoT FL and showed that energy–latency trade-offs can be improved when CPU frequency and transmission power are co-managed [16]. Yang et al. proposed an asynchronous hierarchical FL framework with bandwidth allocation and client scheduling, highlighting the importance of adaptive scheduling under mobility and bandwidth variation [17]. Zhao et al. formulated a mixed-integer optimization problem for joint client and wireless resource selection in IoT FL [18]. Zhou et al. further extended this research direction by studying joint client selection and resource allocation under imperfect channel state information, thereby emphasizing the need to balance energy cost and learning quality in realistic radio environments [19]. Collectively, these studies provide strong evidence that random scheduling is inadequate for heterogeneous wireless FL systems.

FedCS is also directly relevant because it selects FL clients despite heterogeneous mobile edge resources and a completion time constraint [20]. Compared with such resource-aware selection, DCS-RA keeps the same practical concern for heterogeneous client capability but adds an explicit fairness counter and a separate marginal latency RB allocation step. The distinction is not that DCS-RA replaces FedCS-style methods, but that it combines resource-aware selection and post-selection RB assignment in a transparent scheduler targeted at repeated 5G/6G edge rounds [21,22].

MEC and privacy-aware architecture studies have further broadened this context. MEC resource allocation studies have demonstrated that communication and computation resources should be coordinated when latency and edge service efficiency are central design objectives [22,23]. Privacy-oriented 5G and sensor system architectures also explain why decentralized data processing is attractive when raw data transfer creates operational, security, or privacy exposure [24,25]. These works are not direct performance competitors for DCS-RA because several of them do not implement FL client scheduling; rather, they moti-

vate the coexistence of resource-aware learning, edge computing, and privacy-conscious system design in practical 5G/6G ecosystems.

Positioning of the present work relative to representative wireless FL scheduling studies:

Yu et al. [16]: The primary focus is energy–latency-aware resource management for wireless IoT FL. The study performs joint client selection and resource allocation, but it does not explicitly incorporate fairness. Its deployment orientation targets optimization-driven and deadline-based scheduling.

Yang et al. [17]: The primary focus is asynchronous hierarchical FL with bandwidth allocation and mobility-aware scheduling. The study also performs joint selection and allocation, but without an explicit fairness mechanism. Its deployment orientation targets a hierarchical architecture that introduces greater control state overhead.

Zhao et al. [18]: The primary focus is utility-maximizing joint client and radio resource optimization. The study includes joint selection and allocation, but it does not explicitly address fairness. Its deployment orientation targets mixed-integer optimization for wireless IoT settings.

Zhou et al. [19]: The primary focus is joint selection and allocation with imperfect channel state information and the energy–learning trade-off. The study performs joint selection and allocation, but again without explicit fairness regularization. Its deployment orientation targets more elaborate optimization under imperfect radio state knowledge.

This work: The primary focus is round latency minimization through fairness-aware lightweight scheduling. Like the studies above, it includes joint client selection and resource allocation, but it additionally includes explicit fairness. Its deployment orientation targets an interpretable score fusion strategy combined with greedy resource block assignment, making it more suitable for repeated round-wise operation in practical edge environments.

2.3. Research Gap

The research gap is therefore not the absence of scheduling studies, but rather the limited availability of a clearly structured and low-complexity method that integrates the following three requirements simultaneously: heterogeneous computation capability, wireless channel quality, and explicit fairness regularization. Many advanced optimization-based studies address joint scheduling effectively, but they often rely on control state estimation, decomposition procedures, or learning-based coordination mechanisms that may complicate repeated per-round deployment at the edge. Motivated by both the review literature and the research literature, this paper focuses on a method that is analytically transparent, operationally lightweight, and still faithful to the dominant bottlenecks identified in wireless FL. In this sense, the proposed DCS-RA framework is intended not to replace more complex optimization methods, but to provide a practically deployable alternative that preserves scientific rigor while improving applicability in dynamic heterogeneous 5G/6G edge environments. The paper therefore provides a novel integration and practicality contribution where computation-aware scoring, channel-aware scoring, fairness pressure, and RB allocation are combined in one interpretable round-level controller; however, stronger state-of-the-art benchmarking remains necessary before we can make broader comparative claims. Accordingly, DCS-RA should be read as a deployment-oriented bridge between very simple random participation and heavier optimization- or learning-based scheduling policies. Its specific novelty lies in the compact coupling of four operations within one round controller: state-aware scoring, fairness pressure injection, top client selection, and marginal-latency RB assignment.

2.4. Mathematical Notation Used in This Study

Table 1 defines the mathematical symbols used in the system model, optimization formulation, client-scoring function, and resource allocation rule. Introducing these symbols before the formal formulation improves readability and prevents ambiguity when the same variables are reused across computation, communication, and scheduling contexts.

Table 1. Summary of mathematical symbols used throughout the manuscript.

Symbol	Meaning	Usage/Context
$N, \mathcal{N}$	Number and set of candidate clients	Defines the available FL participants.
$K, \mathcal{K}$	Number and set of orthogonal resource blocks	Defines the uplink radio resource pool.
i, k	Client and resource block indices	Used in all scheduling and communication expressions.
$\mathcal{D}_i$	Private local dataset of client i	Used to compute local workload without transferring raw data.
x_i	Binary client selection variable	Equals one when client i participates in the current round.
$a_{i,k}$	Binary RB assignment variable	Equals 1 when RB k is assigned to client i .
$P_{i,k}, P_i^{\max}$	Transmit power on RB k and maximum client power budget	Used in uplink rate and power budget constraints.
$h_{i,k}$	Channel gain between client i and RB k	Captures wireless link quality.
B_k, N_0	RB bandwidth and noise power spectral density	Used in Shannon-based uplink rate calculation.
$R_{i,k}, R_i$	Per-RB and total uplink rates of client i	Determine communication delay.
S_m	Model update payload size	Used to compute uplink transmission time.
R	Number of FL communication rounds	Used in the expanded per-simulation complexity discussion.
$D_i = \mathcal{D}_i $	Number of local samples at client i	Represents client-level data-volume heterogeneity.
E, C_s, f_i	Local epochs, cycles per sample, and client CPU frequency	Used to compute local training time.
η	Local learning rate used by the local optimizer	Relevant to accuracy and convergence analysis; not used in the latency-only scheduling equations.
C_i	Local computation load of client i	Aggregates local epochs, data volume, and cycles per sample.
$T_i^{\text{comm}}, T_i^{\text{comp}}$	Communication and computation time of client i	Form the selected client completion time.
T_{round}	Synchronous FL round completion time	Optimization objective determined by the slowest selected client.
$M_{\min}, M$	Minimum and actual number of selected clients	Enforce participation and determine selected set size.
$S_i^{\text{comp}}, S_i^{\text{chan}}, S_i^{\text{fair}}$	Computation, channel, and fairness scores	Components of DCS-RA client ranking.
n_i	Consecutive skipped round counter for client i	Used to regulate fairness-aware participation.
w_1, w_2, w_3	Score fusion weights	Balance computation, channel, and fairness terms.
$\Delta_{i,k}, \hat{T}_i(\cdot)$	Marginal latency reduction and estimated completion time	Used by the greedy RB allocation rule.
$\mathcal{M}, \mathcal{K}_i$	Selected client set and RBs currently assigned to client i	Used during post-selection resource allocation.
q_i	Compact state report submitted by client i	Represents computation estimate, local workload proxy, channel state, and participation history used by the scheduler.
p_i	Participation count of client i over the evaluation horizon	Required for participation frequency variance, minimum/maximum participation ratio, and Jain-style fairness analysis.

3. System Model and Problem Formulation

3.1. Network and Learning Architecture

We consider a synchronous FL system deployed over a heterogeneous edge network. A coordinating server, located at a base station or edge controller, manages a set of N candidate clients denoted by $\mathcal{N} = \{1, 2, \dots, N\}$. Each client $i \in \mathcal{N}$ stores a private local dataset $\mathcal{D}_i$ and trains a local copy of the global model when selected. For notational compactness, $D_i = |\mathcal{D}_i|$ denotes the number of local samples held by client i .

The uplink spectrum is partitioned into K orthogonal resource blocks (RBs) collected in the set $\mathcal{K} = \{1, 2, \dots, K\}$. At the beginning of each communication round, the server must determine which clients will participate and how the available RBs will be assigned to the selected set. This decision must explicitly account for heterogeneous client computation capability and time-varying wireless conditions because both factors jointly determine whether the FL round proceeds efficiently or whether it is delayed by a small number of weak participants.

The network and learning architecture corresponding to this formulation is shown in Figure 1. The figure separates the coordinating server, resource-aware scheduler, 5G/6G wireless edge layer, and distributed clients to clarify the control and data flows assumed by the proposed model. This visualization also makes it explicit that raw data remain at the edge clients, while scheduling metadata, model updates, and global model parameters are exchanged through the wireless edge infrastructure.

In the autonomous vehicular scenario introduced earlier, the clients in Figure 1 can represent connected vehicles or roadside sensing units. The edge controller can be colocated with a base station or multi-access edge computing node. The state-reporting path conveys

only scheduling metadata, such as computation estimates, channel quality, workload proxies, and participation history, whereas the learning path carries model parameters and updates. This separation clarifies that DCS-RA is a scheduling mechanism and does not by itself implement secure aggregation, malicious client detection, or privacy-preserving metadata protection.

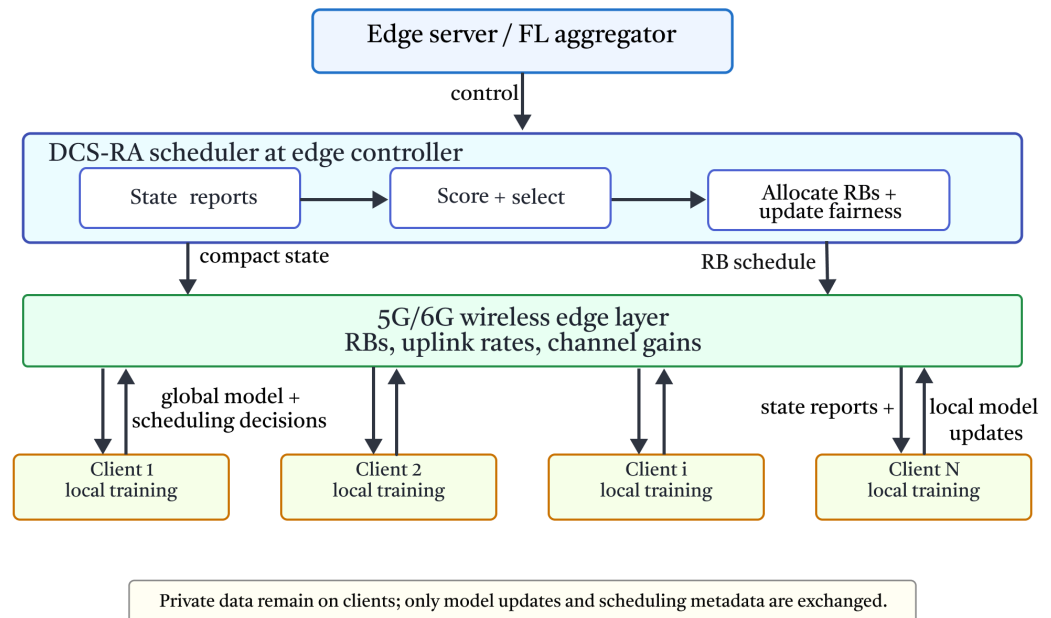


Figure 1. Deployment-oriented network and learning architecture of the proposed DCS-RA-enabled FL system. The edge server or FL aggregator receives compact state reports from heterogeneous clients, performs resource-aware scoring and greedy RB allocation through the 5G/6G edge layer, distributes the global model and scheduling decisions, and aggregates returned local updates while raw data remain on the clients.

3.2. Communication Model

Let $a_{i,k} \in \{0, 1\}$ indicate whether RB k is assigned to client i , and let $P_{i,k}$ denote the corresponding transmit power. The achievable uplink rate of client i over RB k is modeled using Shannon's expression $R_{i,k} = a_{i,k} B_k \log_2 \left(1 + \frac{P_{i,k} h_{i,k}}{N_0 B_k} \right)$, where B_k is the RB bandwidth, $h_{i,k}$ is the channel gain, and N_0 is the noise power spectral density.

The total uplink rate of client i is then $R_i = \sum_{k \in \mathcal{K}} R_{i,k}$.

If the model update payload size is S_m bits, the communication time becomes $T_i^{\text{comm}} = \frac{S_m}{R_i}$.

This formulation directly explains why radio-aware scheduling is necessary. For a fixed model update size, clients with a poor effective uplink rate incur disproportionately large transmission delays. In synchronous FL, such delays are especially harmful because they propagate from the affected client to the completion time of the entire global round.

3.3. Computation Model

Each selected client performs local training for E epochs. Let f_i denote the client CPU frequency in cycles/s, and let C_s represent the required number of cycles per data sample. The local computation load is $C_i = E |\mathcal{D}_i| C_s$, and the corresponding local computation time is $T_i^{\text{comp}} = \frac{C_i}{f_i}$.

This abstraction is standard in systems-oriented FL because it links device heterogeneity directly to straggling risk. Clients with lower CPU capability or larger local workloads

require more time to complete local updates, and this can significantly slow synchronous aggregation if those clients are admitted without discrimination.

3.4. Round Latency Objective

Let $x_i \in \{0, 1\}$ indicate whether client i is selected in the current round. Since the server waits for all selected clients in synchronous FL, the round completion time is determined by the slowest selected participant: $T_{\text{round}} = \max_{i \in \mathcal{N}: x_i=1} (T_i^{\text{comp}} + T_i^{\text{comm}})$.

The scheduling problem is therefore formulated as minimizing T_{round} over the client selection variables $\{x_i\}$, the RB assignment variables $\{a_{i,k}\}$, and the transmit power variables $\{P_{i,k}\}$, subject to the following constraints:

First, the number of selected clients must satisfy the minimum participation requirement: $\sum_{i=1}^N x_i \geq M_{\min}$.

Second, each RB can be assigned to at most one client: $\sum_{i=1}^N a_{i,k} \leq 1$ for all $k \in \mathcal{K}$.

Third, the total transmit power allocated to each client must not exceed its maximum power budget: $\sum_{k=1}^K P_{i,k} a_{i,k} \leq P_i^{\max}$ for all $i \in \mathcal{N}$.

Fourth, RBs can only be assigned to clients that are selected: $a_{i,k} \leq x_i$ for all $i \in \mathcal{N}$ and all $k \in \mathcal{K}$.

Finally, both x_i and $a_{i,k}$ are binary decision variables.

The resulting formulation is a mixed-integer nonlinear problem because client participation, RB allocation, and achievable transmission rates are jointly coupled. Solving this problem exactly at every FL round is generally impractical in edge environments, where decisions must be taken quickly and repeatedly. This motivates the development of a structured low-complexity heuristic that remains aligned with the original optimization objective while being operationally feasible.

4. Proposed Dynamic Client Selection and Resource Allocation Method

4.1. Methodological Design Principles

The proposed method is built around four scientific design principles: First, client selection should explicitly reduce the probability of computation-side stragglers. Second, channel-aware scheduling should prioritize clients that can return model updates efficiently under current wireless conditions. Third, fairness should be treated as a design variable rather than a post hoc discussion point, because persistent exclusion of weaker clients can reduce long-run participation and data representativeness [11,14]. Fourth, the controller must remain lightweight enough to operate on a round-by-round basis in realistic edge deployments.

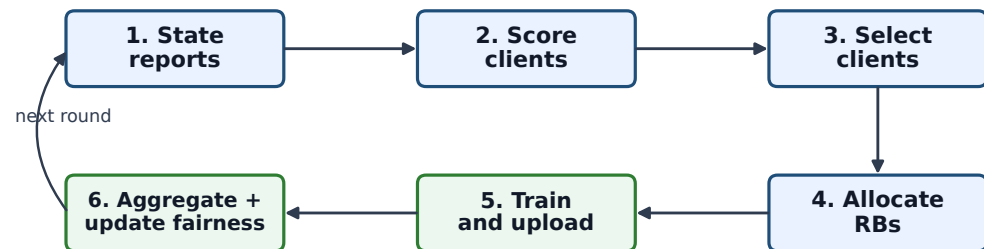
These principles are important because a practically useful FL scheduler must balance efficiency and inclusiveness. A policy that focuses only on fast devices may reduce latency in the short term, but it can weaken representativeness and long-term participation. Conversely, a policy that ignores system heterogeneity may preserve inclusiveness yet produce repeated straggling and poor resource utilization. The proposed method is therefore designed to strike a structured balance between these competing requirements. This design differs from client selection-only policies because the selected set is followed by RB assignment, and it differs from exact joint optimization because the controller intentionally uses score fusion and greedy marginal allocation to reduce runtime and implementation burden.

4.2. Workflow Overview

Figure 2 summarizes the proposed DCS-RA workflow. The server first gathers the minimum state required for scheduling, namely, device computation capability, a local workload proxy, channel gains, and a fairness counter that tracks the number of rounds in which a client has been skipped. It then computes an interpretable score for each client,

selects the top M candidates, and allocates RBs greedily according to the marginal reduction in estimated completion time. After local training and model upload, the server aggregates the received updates and refreshes the fairness counters.

DCS-RA workflow at the edge controller



Inputs: resource state and participation history; outputs: selected clients, RB assignment, and aggregated model.

Figure 2. Implementation workflow of the proposed DCS-RA method at an edge controller. The workflow starts with client state reporting, proceeds through score computation, client ranking, top- M selection, greedy RB allocation, local training, model upload, aggregation, and fairness history updating, and then repeats in the next communication round.

The inclusion of Figure 2 is scientifically important because it makes the logic of the proposed method operationally explicit. In scheduling-oriented FL research, performance gains are meaningful only when the decision path from the observed system state to client participation and radio assignment is clearly interpretable. The workflow figure therefore does more than illustrate the method; it demonstrates that DCS-RA relies only on lightweight, directly relevant scheduling signals and that each stage of the controller maps to a specific systems objective. In particular, the figure clarifies that fairness is not appended after optimization but integrated into the decision loop itself, while RB allocation is explicitly linked to round latency reduction rather than generic throughput maximization.

4.3. Client State Collection and Secure Reporting Considerations

DCS-RA assumes that the edge controller receives a compact state report q_i from each available client before scheduling. In the simulated formulation, this report includes or enables estimation of CPU frequency f_i , local workload size D_i , channel quality $h_{i,k}$ or its aggregate equivalent, and participation history n_i . These values are used only for scheduling and are not raw training samples. Nevertheless, scheduling metadata can still reveal system-level information, such as device capability or link quality, and malicious clients may attempt to falsify reported resource metrics to increase or avoid selection. The present framework does not include cryptographic attestation, secure aggregation of scheduling metadata, malicious client filtering, or privacy-preserving state reporting. These protections are therefore identified as deployment-layer extensions rather than existing properties of DCS-RA.

A practical implementation could reduce this risk by using network-measured channel quality indicators, controller-side timing observations, bounded self-reports, trusted execution support, anomaly detection on reported metrics, or privacy-preserving quantized state signals. These mechanisms are not evaluated in the present simulation, so the security and privacy discussion is intentionally framed as a limitation and deployment requirement rather than an implemented contribution.

4.4. Client Scoring

For each available client i , DCS-RA constructs three components:

Computation score

Clients with higher CPU capacity and lighter effective local workloads should be favored. We define $S_i^{\text{comp}} = \frac{f_i}{E|\mathcal{D}_i|\mathcal{C}_s}$.

This score increases when the client has stronger computation capability or a lower effective workload and therefore directly captures the likelihood that the client can complete local training without becoming a computation-side straggler.

Channel score

Because a client's transmission efficiency depends on its radio state across the available RBs, we use the mean channel gain $\bar{h}_i = \frac{1}{K} \sum_{k=1}^K h_{i,k}$ and define $S_i^{\text{chan}} = \bar{h}_i$.

This term reflects the client's expected communication quality in a simple but effective manner. Clients with stronger average channel conditions are more likely to upload their model updates quickly, thereby reducing the communication component of round latency.

Fairness score

Let n_i denote the number of consecutive rounds in which client i has not been selected. To gradually increase the participation probability of overlooked clients, we define $S_i^{\text{fair}} = \sqrt{n_i + 1}$.

The square root form grows monotonically but sublinearly, which avoids over-correcting in favor of very weak clients in a single round while still ensuring that repeated exclusion is progressively discouraged. This term should be interpreted as a fairness encouragement mechanism rather than a formal guarantee of equal participation, because per-client participation frequencies and Jain's fairness index are not reported in the present evaluation.

The overall client score is $S_i = w_1 S_i^{\text{comp}} + w_2 S_i^{\text{chan}} + w_3 S_i^{\text{fair}}$, where $w_1, w_2, w_3 \in [0, 1]$ and $w_1 + w_2 + w_3 = 1$. In implementation, the score components are evaluated from the current round state and fused using weights selected to balance computation efficiency, communication efficiency, and fairness. The server then ranks all available clients and selects the top M candidates, where $M \geq M_{\min}$. The reported setting $(w_1, w_2, w_3) = (0.4, 0.4, 0.2)$ is interpreted as a balanced engineering choice that gives equal priority to computation and channel quality while assigning a smaller but nonzero weight to fairness pressure. It should not be interpreted as universally optimal. The formulation assumes that score components are placed on comparable scales before fusion; otherwise, the absolute numerical scale of one component could dominate the weighted sum. A sensitivity analysis over alternative weight triples remains necessary to quantify the robustness of this heuristic configuration. Thus, the selected weights are best understood as a reproducible heuristic configuration: w_1 and w_2 give equal emphasis to computation-side and communication-side straggler reduction, while w_3 keeps fairness active without allowing the fairness reward to dominate the latency objective.

This scoring strategy is scientifically meaningful because it integrates the three dominant scheduling dimensions identified by the literature: device capability, communication condition, and participation equity. As a result, the selected client set is expected to be both more efficient and more operationally robust than a random or purely greedy alternative. The wording is intentionally cautious: the available evidence supports improvement over the implemented random baseline, not superiority over all FedCS-style, utility-aware, deadline-aware, or fairness-aware schedulers.

4.5. Marginal Latency-Aware Resource Allocation

After fixing the selected client set $\mathcal{M}$, DCS-RA allocates RBs greedily. Importantly, the allocation target is not raw throughput alone. Instead, the method estimates how much

assigning one additional RB to a selected client would reduce its round completion time. For each candidate pair (i, k) , the marginal benefit is $\Delta_{i,k} = \hat{T}_i(\mathcal{K}_i) - \hat{T}_i(\mathcal{K}_i \cup \{k\})$, where $\mathcal{K}_i$ is the set of RBs currently assigned to client i and $\hat{T}_i(\cdot)$ is the estimated completion time under the resulting rate. At each step, the scheduler assigns the next available RB to the client–RB pair with the largest positive $\Delta_{i,k}$. This rule is directly aligned with the optimization objective because it preferentially reduces the strongest sources of straggling.

To preserve deployment simplicity, DCS-RA uses uniform transmit power allocation across the RBs assigned to each client, subject to the client power budget. This approximation is deliberate. Although more sophisticated water filling or reinforcement learning-based control is possible, the contribution of this paper lies in a lightweight joint scheduler rather than a high-overhead optimization framework. The adopted allocation rule is therefore justified by the need to retain low runtime complexity while still making latency-oriented resource decisions. This allocation step is the main difference between DCS-RA and methods that stop after ranking clients: selected clients are not merely admitted, but are subsequently assigned radio resources according to their estimated contribution to reducing the synchronous round bottleneck.

4.6. Operational Complexity and Practicality

The per-round scheduling cost can be decomposed into four operations: First, computing the three score components for all clients requires one pass over the available client state and has time complexity $\mathcal{O}(N)$ with $\mathcal{O}(N)$ temporary storage for scores and counters. Second, ranking clients by the fused score requires $\mathcal{O}(N \log N)$ time under standard sorting, after which extracting the top M clients is linear in the selected set size. Third, greedy RB allocation depends on how marginal gains are updated. A direct implementation that recomputes all selected client/RB marginal gains after each RB assignment has a worst-case time complexity $\mathcal{O}(MK^2)$. If marginal gains are cached and updated through a priority structure, the allocation step can be implemented closer to $\mathcal{O}(MK + K \log(MK))$ after initialization. Fourth, fairness counter updates require $\mathcal{O}(N)$ time per round.

Therefore, for R communication rounds, a conservative direct implementation scales as $\mathcal{O}(R(N \log N + MK^2))$ time and $\mathcal{O}(N + MK)$ space, excluding the cost of the local neural network training itself. A more optimized implementation can reduce the allocation term, but DCS-RA does not claim formal optimality for the greedy scheduler. The complexity advantage of DCS-RA should be interpreted qualitatively: it avoids solving the original mixed-integer nonlinear problem at every round and uses operations that are compatible with repeated online scheduling in heterogeneous edge environments. Experimental scalability over larger client sets and varying RB budgets remains necessary. Multiple values of N and K , such as client counts of 50, 75, 100, 125, and 150 with alternative resource block budgets, would be required before making empirical scalability claims beyond the current 100-client, 20-RB setting.

5. Experimental Setup

5.1. Datasets and Workload Characteristics

The experimental workloads use MNIST and CIFAR-10, two widely adopted image classification benchmarks with different levels of visual complexity. MNIST contains grayscale handwritten digits and is commonly used to evaluate lightweight learning pipelines, whereas CIFAR-10 contains small natural color images with substantially greater intra-class variability and richer visual structure [21]. Their joint use is therefore appropriate for testing whether a scheduling method remains effective across both relatively simple and more communication-demanding FL workloads.

Dataset characteristics used to motivate the evaluation workloads are summarized as follows:

MNIST [26]: input size is 28×28 grayscale with 10 classes and a canonical split of 60,000 training images and 10,000 test images. In the present study, MNIST serves as a lightweight benchmark for examining communication round efficiency on relatively simple visual features with lower representational complexity.

CIFAR-10 [21]: input size is 32×32 RGB with 10 classes and a canonical split of 50,000 training images and 10,000 test images. In the present study, CIFAR-10 provides more heterogeneous image content and greater visual complexity, making it a stronger workload for evaluating latency robustness under more demanding FL updates.

The joint use of both datasets strengthens the evaluation because it reduces the risk that the observed scheduling benefit is tied to a single workload type. Instead, it allows the study to assess whether the proposed method retains its system-level advantage across tasks with different levels of data complexity.

5.2. Dataset Partitioning and Non-IID Status

The simulation explicitly represents systems heterogeneity through unequal local dataset sizes, CPU frequencies, transmit power budgets, and channel gains. However, the present evaluation does not fully characterize statistical heterogeneity across clients. In particular, it does not report a class-wise client partition table, a Dirichlet concentration parameter, or separate label-skew, quantity-skew, and feature-skew settings. This distinction is important because real FL deployments often combine system heterogeneity with non-IID data, and statistical heterogeneity can influence both convergence and fairness. Consequently, the reported results should be interpreted primarily as system heterogeneity results rather than a full non-IID FL validation.

5.3. Simulation Environment

The evaluation uses a controlled simulation setting designed to emulate a heterogeneous wireless edge environment. A custom simulator models client computation, client selection, and radio resource allocation over multiple FL rounds. The purpose of the experiments is to evaluate system efficiency, specifically, round completion latency and stability, rather than to demonstrate a complete end-to-end benchmark against all modern FL optimizers.

The simulation parameters used in the evaluation are as follows:

Number of clients N : 100.

Number of resource blocks K : 20.

Minimum selected clients per round $M_{\min}$: 10.

Number of FL rounds: 50.

Model update size S_m : 10 MB (80 Mbits).

Local epochs per round E : 5.

CPU cycles per sample C_s : 1000 cycles/sample.

Client CPU frequency f_i : uniformly distributed in $[1, 5]$ GHz.

Maximum client transmit power $P_i^{\max}$: uniformly distributed in $[0.1, 1.0]$ W.

Resource block bandwidth B_k : 180 kHz.

Noise power spectral density N_0 : 10^{-20} W/Hz.

Channel gain $h_{i,k}$: uniformly distributed in $[10^{-10}, 10^{-8}]$.

DCS-RA weights (w_1, w_2, w_3) : (0.4, 0.4, 0.2).

Datasets: MNIST and CIFAR-10.

Client dataset sizes are heterogeneous and randomly generated to emulate uneven local data volumes. This modeling choice is appropriate because the targeted issue of

the paper is heterogeneity-induced straggling. The random baseline is Random Client Selection with Uniform Resource Allocation, in which the controller selects the minimum required number of clients uniformly at random and distributes RBs uniformly among them. This baseline is intentionally simple and therefore serves as a lower-bound deployment benchmark against which the benefit of structured scheduling can be assessed. However, this baseline is not sufficient to establish superiority over state-of-the-art schedulers. Future benchmarking should include FedAvg with random participation under identical training curves, FedCS-style resource-aware selection, Oort-style utility-aware selection, deadline-aware scheduling, joint communication–computation optimization, and fairness-aware participation scheduling. The reported weights are likewise fixed for reproducibility in the present simulation and should be validated through sensitivity analysis in extended experiments.

5.4. Assumption Justification and Realism Limits

The uniform distributions used for CPU frequency, transmit power, and channel gain are simplified abstractions for controlled simulation rather than complete models of 5G/6G radio behavior. They are useful for isolating the effect of heterogeneous computation and communication resources, but they do not fully capture fading, mobility, shadowing, interference, traffic bursts, handovers, or client dropout. Consequently, the reported results should be interpreted as evidence under a controlled heterogeneous simulation, not as proof of behavior for all realistic vehicular, industrial, or healthcare deployments.

More realistic validation would require replacing or augmenting the uniform channel model with mobility-aware and fading-aware models, such as Rayleigh or Rician fading, path loss and shadowing profiles, interference variation, trace-driven channel quality indicators, and stochastic client availability. The present study does not report such channel dynamics experiments; therefore, our deployment claims are limited to the controlled heterogeneous simulation setting evaluated here.

5.5. Hardware and Software Environment

The reported round completion times are obtained from the analytical simulation model of computation and uplink communication latency rather than from direct wall-clock execution on a hardware testbed. Accordingly, hardware characteristics are not used as explanatory variables in the present numerical results. A hardware-in-the-loop or real MEC testbed implementation should report the CPU, memory, accelerator, operating system, software versions, simulator settings, and random-seed protocol used for runtime profiling.

Figure 3 combines the three key heterogeneity indicators used in the simulation: panel (a) shows the distribution of client CPU frequency, panel (b) shows the distribution of local dataset size per client, and panel (c) shows the distribution of maximum client transmit power.

The inclusion of Figure 3 is methodologically critical because it verifies that the evaluation environment genuinely reflects the heterogeneity assumed in the system model. Panel (a) confirms that the clients are not computation homogeneous, which means that local training times can differ substantially even before radio effects are considered. Panel (b) shows that local workloads are also uneven, creating additional variability in computation latency. Panel (c) further demonstrates that the communication side is heterogeneous as well, because the available transmit power budget differs from client to client. Without these three forms of heterogeneity, the scheduling problem would be artificially simplified and the practical value of DCS-RA would be difficult to justify. The figure therefore provides direct visual support for the central premise of the paper: heterogeneous system conditions generate uneven completion times, and those uneven completion times create the need for an intelligent joint scheduler.

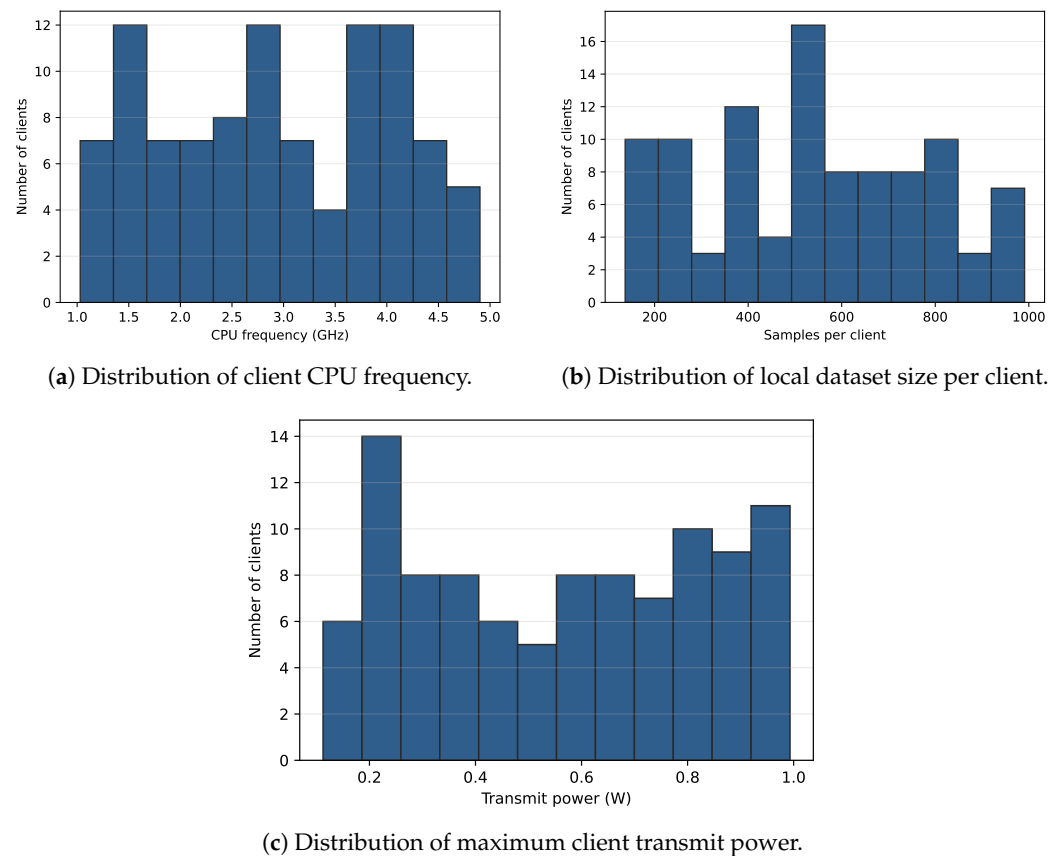


Figure 3. System heterogeneity in the simulated edge environment: panel (a) visualizes computation diversity, panel (b) visualizes workload imbalance across clients, and panel (c) visualizes communication resource asymmetry through heterogeneous power budgets. Together, the three panels show that the simulated FL clients differ in both computation-side and communication-side resources.

5.6. Evaluation Metrics and Their Justification

Three metrics are emphasized:

- **Round completion time:** The maximum selected client completion time in a global round. This is the most relevant systems metric because synchronous FL aggregation is blocked by the slowest selected client.
- **Average round completion time:** The average round completion time across all rounds. This captures long-run training throughput and provides a stable summary of system efficiency.
- **Selection stability and fairness:** Evaluated indirectly through the fairness term in the scheduling rule and through the dispersion of round completion times. Because per-client participation frequency vectors are not reported, this evidence should be interpreted as an indirect stability indicator rather than a complete quantitative fairness proof.

These metrics are appropriate because the paper addresses a scheduling problem rather than a pure model accuracy problem. Accordingly, the evaluation prioritizes wall-clock efficiency, latency predictability, and participation behavior, which are the most direct indicators of whether the proposed method improves synchronous FL operation in heterogeneous wireless environments. A complete learning-oriented FL evaluation would also report global model accuracy, training loss, convergence curves over communication rounds, rounds needed to reach a target accuracy, client participation distribution, communication cost, training time per round, end-to-end convergence time, Jain's fairness index, and participation frequency variance. These metrics are identified as necessary extensions because the present study evaluates the scheduling layer rather than full model training quality.

6. Results and Discussion

6.1. Main Latency Results

The reported performance of DCS-RA relative to the random baseline is summarized as follows:

MNIST: DCS-RA achieves an average round completion time of 1.55 s, whereas the random baseline requires 1.92 s. This corresponds to an absolute saving of 0.37 s, an improvement of 19.39%, and a standard deviation of 0.05 s for DCS-RA versus 0.17 s for the baseline.

CIFAR-10: DCS-RA achieves an average round completion time of 1.57 s, whereas the random baseline requires 2.02 s. This corresponds to an absolute saving of 0.45 s, an improvement of 22.47%, and a standard deviation of 0.05 s for DCS-RA versus 0.26 s for the baseline.

Table 2 reports the latency results as the mean plus or minus the standard deviation. These standard deviations are computed from the round completion time series and therefore quantify round-to-round variability within the simulation record, not confidence intervals across independent repeated experimental runs.

Table 2. Uncertainty-aware latency summary for DCS-RA and the random baseline. Values are reported as mean $\pm$ standard deviation in seconds.

Dataset	Method	Mean $\pm$ SD (s)	Coefficient of Variation	Mean Latency Improvement
MNIST	Random baseline	1.92 $\pm$ 0.17	8.85%	Reference
MNIST	DCS-RA	1.55 $\pm$ 0.05	3.23%	19.39%
CIFAR-10	Random baseline	2.02 $\pm$ 0.26	12.87%	Reference
CIFAR-10	DCS-RA	1.57 $\pm$ 0.05	3.18%	22.47%

These results show that the proposed method reduces the mean round completion time on both workloads. The absolute savings are operationally meaningful in synchronous FL because each saved fraction of a second reduces the waiting time of the aggregation server and increases the number of rounds that can be executed within a fixed training budget. Interpreted through coefficients of variation, the round time dispersion decreases from 8.85% to 3.23% on MNIST and from 12.87% to 3.18% on CIFAR-10, confirming that DCS-RA improves predictability as well as speed. The evidence should be interpreted as latency and stability evidence only; it does not demonstrate improved model accuracy, faster convergence to a target accuracy, or better final training quality, because those metrics are outside the reported scheduling-layer evaluation.

Figure 4 visualizes the same mean round completion times with standard deviation error bars so that the uncertainty associated with the reported round values is visible in the graph rather than only in the text.

Figure 4 is important because it makes the central empirical result immediately visible: the figure shows that the latency advantage of DCS-RA is not confined to a single dataset; rather, it is preserved across both a simpler workload and a more demanding one. The error bars additionally show that DCS-RA has a lower reported round time dispersion than the random baseline on both datasets. This directly addresses the uncertainty in the reported values and supports the claim that the proposed scheduler improves latency stability, not only the average. This cross-dataset consistency strengthens the systems interpretation of the method. A scheduler that provides gains only under one workload type would have limited practical value, whereas a method that improves both cases is more likely to be robust under realistic deployment diversity.

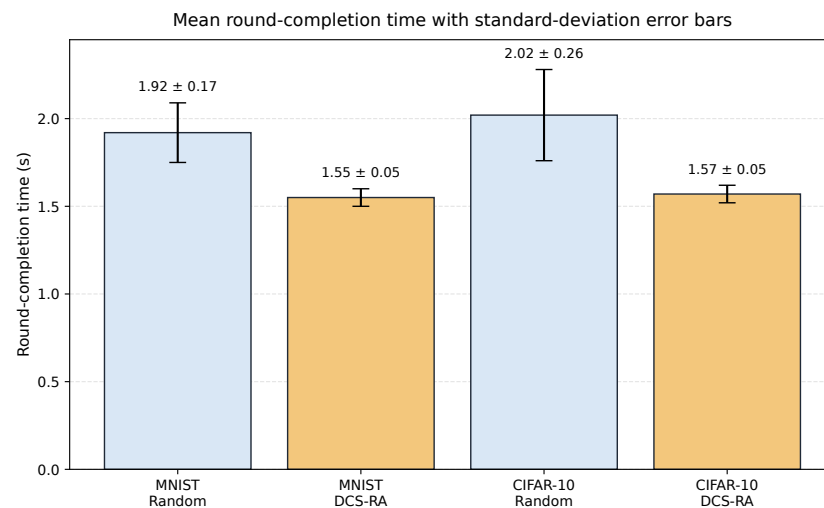


Figure 4. Average round completion time of DCS-RA and the random baseline on MNIST and CIFAR-10 with standard deviation error bars. Light-blue bars denote the random baseline, orange bars denote DCS-RA, and black error bars indicate the standard deviation across runs. The figure highlights both the mean latency advantage and the lower round-to-round uncertainty of the proposed method across both workloads.

6.2. Scalability Analysis

Scalability is an important dimension for wireless FL scheduling because the number of candidate clients may vary substantially across edge deployments. The present numerical evaluation reports the $N = 100$ case and does not provide a full node-scaling curve for $N = 50, 75, 100, 125, 150$. Therefore, the reported results are interpreted only for the evaluated configuration. A complete scalability assessment requires repeated evaluation of DCS-RA and the baseline under consistent resource block, workload, and selection settings across multiple values of N .

6.3. Federated Learning Accuracy and Convergence Metrics

The present evaluation supports round latency analysis but does not report global model accuracy, training loss, convergence curves, rounds to target accuracy, or end-to-end convergence time measurements. For this reason, the empirical claim is limited to scheduling efficiency and latency stability. Demonstrating learning quality benefits would require per-round accuracy and loss traces for DCS-RA and the relevant baselines together with a clearly defined target accuracy threshold and total convergence time.

6.4. Fairness Evidence and Participation Analysis

The fairness term in DCS-RA increases the score of clients skipped in previous rounds, but the present results do not report per-client participation counts, missed-round distributions, or fairness indices. Therefore, the fairness claim is stated conservatively as an encouragement for more balanced participation rather than a fairness guarantee. A complete fairness evaluation should report per-client selection frequency p_i , minimum and maximum participation ratios, participation frequency variance, missed-round distributions, and Jain's fairness index for both DCS-RA and baseline methods.

6.5. Discussion of the Performance Improvement

The observed gain is consistent with the structure of the proposed method. The computation score reduces the probability of selecting clients whose local training time is intrinsically large because of weak CPU capability or heavy local workload. The channel score reduces the probability of admitting clients that would require a disproportionate up-

link time to return model updates. The fairness term, meanwhile, is intended to discourage repeated exclusion of weaker devices, although the present simulation record does not yet quantify the resulting participation distribution. This combined design is scientifically justified by recent survey work showing that communication heterogeneity, resource scarcity, and fairness are intertwined rather than separable FL design issues [10–12,14].

The greedy RB allocation step further strengthens this logic. Uniform allocation treats all selected clients as equally urgent, which is rarely appropriate in synchronous FL. By allocating RBs to maximize the marginal reduction in estimated completion time, DCS-RA shifts the resource controller from a throughput-centric perspective to a latency-centric one. This is an important distinction because the true bottleneck in synchronous aggregation is not aggregate throughput itself, but the slowest completion time among the selected clients.

6.6. Runtime Uncertainty and Adaptation

Runtime uncertainty is a central deployment issue for wireless FL. Rapid channel fluctuations can make a pre-round channel estimate obsolete, client dropout can reduce the number of returned updates, delayed uploads can create aggregation stalls, and changing CPU availability can alter local training time after scheduling has already been performed. DCS-RA partially addresses such uncertainty by recomputing scores and allocations at every communication round using the most recent available state. This round-level adaptation is consistent with the lightweight nature of the method.

However, DCS-RA is a round-level scheduler rather than a fully adaptive intra-round controller. It does not implement mid-round incremental RB reallocation, dropout recovery, stale update acceptance, asynchronous aggregation, or predictive mobility tracking. These mechanisms are important future extensions for highly mobile vehicular networks and volatile industrial wireless environments. The claim is therefore limited to round-level adaptive scheduling under the simulated state model.

6.7. 5G and 6G Deployment Implications

The proposed method is intentionally described for 5G/6G edge networks because the scheduling bottleneck appears in both generations; however, the deployment interface can differ. In 5G systems, DCS-RA can be interpreted as an MEC-side or base station-adjacent scheduler that uses available channel quality indicators, uplink resource information, and coarse device computation reports to avoid stragglers during synchronous FL rounds. In this case, the method is constrained by the granularity and update frequency of the radio and edge computing information available to the controller.

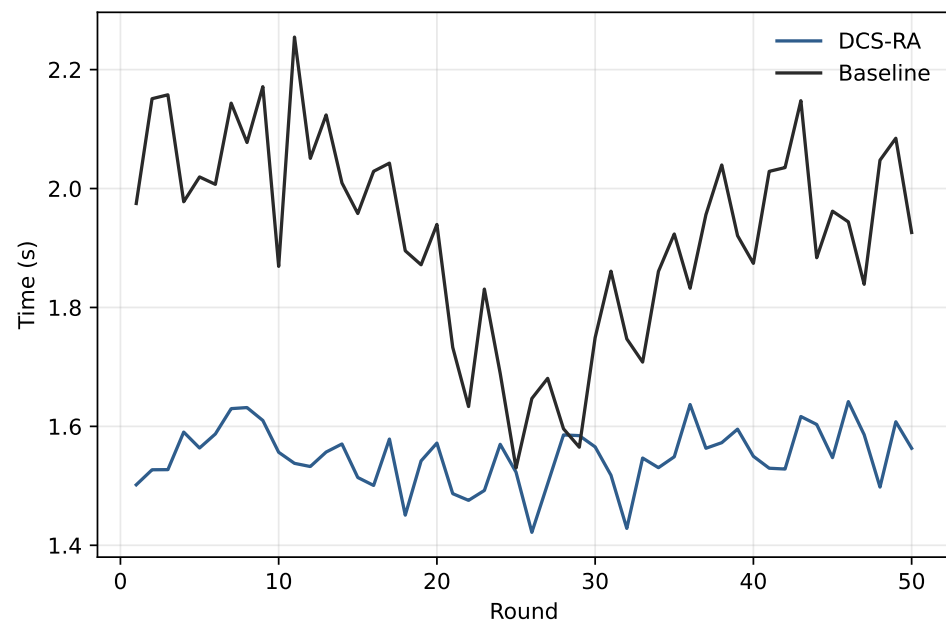
In 6G-oriented systems, the same principle may benefit from richer network state observability, AI-native control loops, integrated sensing and communication, cell-free or ultra-dense access, and predictive mobility/channel estimation. These capabilities could make client scoring and RB allocation more responsive, but they are not required by the current simulation model. Therefore, the adoption of DCS-RA does depend on the network generation in terms of control plane information, update frequency, radio resource granularity, and mobility awareness. The core algorithmic contribution remains generation agnostic, while the practical effectiveness may improve as the network exposes richer and more timely state information.

6.8. MNIST Robustness

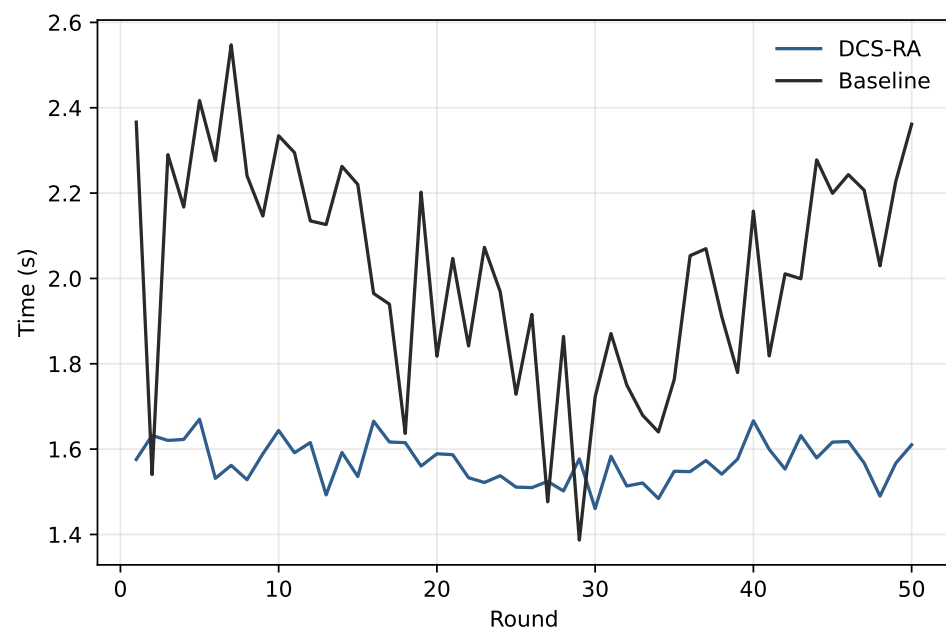
The MNIST results provide the first robustness check for the proposed scheduler under a relatively lightweight workload. Although MNIST is less visually complex than CIFAR-10, the scheduling problem remains nontrivial because the latency bottleneck arises from heterogeneous client resources rather than from input semantics alone. On MNIST, DCS-RA reduces the average round completion time from 1.92 s to 1.55 s and sharply

reduces dispersion from 0.17 s to 0.05 s. This indicates that the method does not merely accelerate the mean case; it also suppresses the variability associated with unfavorable client selections and inefficient RB assignment.

To make the robustness behavior explicit at the round level, Figure 5 compares the per-round completion times of DCS-RA and the random baseline for both datasets, while Figure 6 compares the corresponding cumulative averages. The placement of these figures here is deliberate. MNIST serves as the simpler workload, and it therefore provides a useful reference point from which the stronger effects observed later for CIFAR-10 can be interpreted more clearly.



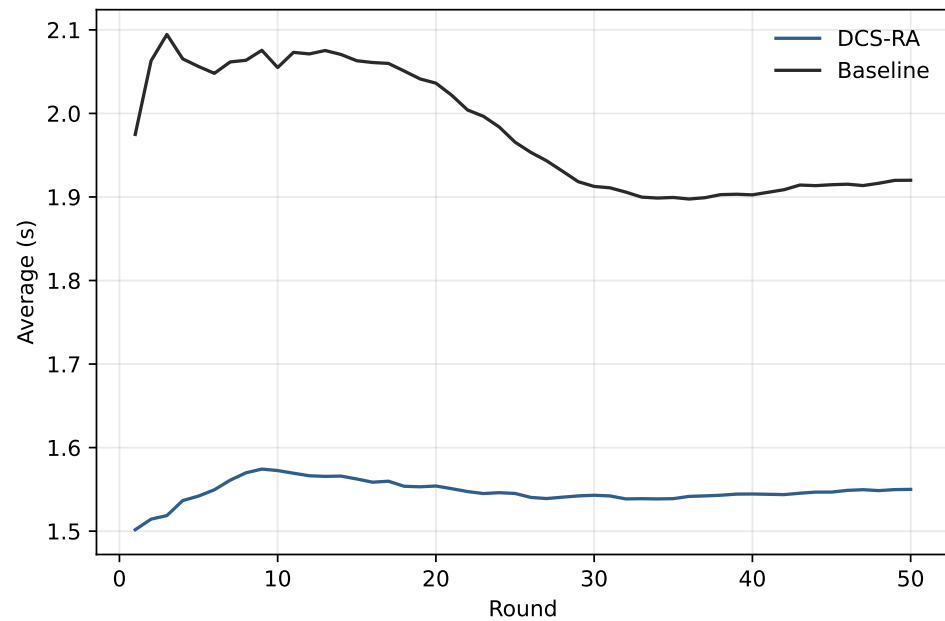
(a) Per-round completion time on MNIST.



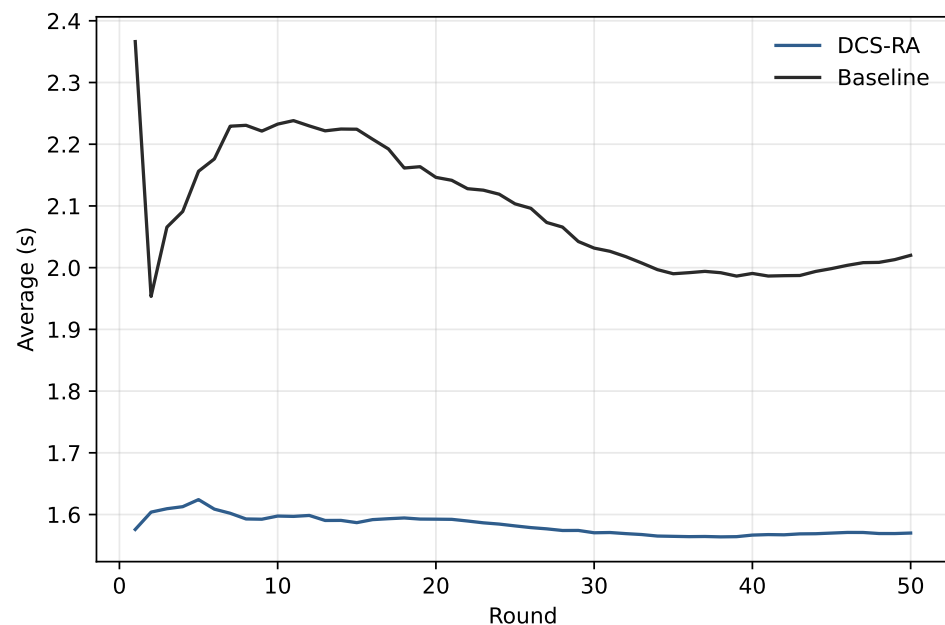
(b) Per-round completion time on CIFAR-10.

Figure 5. Round-wise robustness of DCS-RA relative to the random baseline. Panel (a) reports MNIST behavior and panel (b) reports CIFAR-10 behavior. In both cases, the proposed method remains consistently below or less volatile than the baseline, indicating sustained round-level scheduling benefit.

For MNIST specifically, panel (a) of Figure 5 shows that DCS-RA remains in a narrow latency band across rounds, whereas the random baseline exhibits visibly larger fluctuations. Panel (a) of Figure 6 strengthens this interpretation by showing that the cumulative average of DCS-RA stabilizes quickly and remains below that of the baseline throughout the run. This is scientifically important because it demonstrates that the benefit of the proposed method is not produced by a few isolated favorable rounds; rather, it arises from repeated and consistent scheduling decisions that prevent persistent straggling.



(a) Cumulative average round time on MNIST.



(b) Cumulative average round time on CIFAR-10.

Figure 6. Cumulative robustness of DCS-RA relative to the random baseline. Panel (a) shows MNIST and panel (b) shows CIFAR-10. The persistent separation between the two curves indicates that the proposed scheduler achieves sustained rather than isolated gains over the training horizon.

6.9. CIFAR-10 Robustness

The CIFAR-10 results provide a stronger robustness test because the workload is more demanding and therefore more sensitive to inefficient client–channel combinations. On CIFAR-10, DCS-RA reduces the average round completion time from 2.02 s to 1.57 s and reduces dispersion from 0.26 s to 0.05 s. The larger absolute and relative gain on this dataset suggests that the scheduler becomes even more valuable as workload difficulty and effective system heterogeneity increase.

This interpretation is visually supported by panel (b) of Figure 5 and panel (b) of Figure 6. In the round-wise view, DCS-RA remains substantially more stable than the random baseline, which experiences pronounced oscillations and repeated high-latency rounds. In the cumulative view, the separation between the two curves remains persistent, indicating that the improvement is not transient. These figures therefore justify a stronger conclusion than a simple average comparison alone: under a more demanding workload, DCS-RA not only improves efficiency but also regularizes the overall training process. This is precisely the type of behavior required in latency-sensitive wireless FL systems, where volatile round durations complicate scheduling, resource provisioning, and service-level assurance.

6.10. Stability and Cross-Dataset Interpretation

Figure 7 summarizes the relative improvement of DCS-RA across datasets in terms of latency reduction and variability reduction.

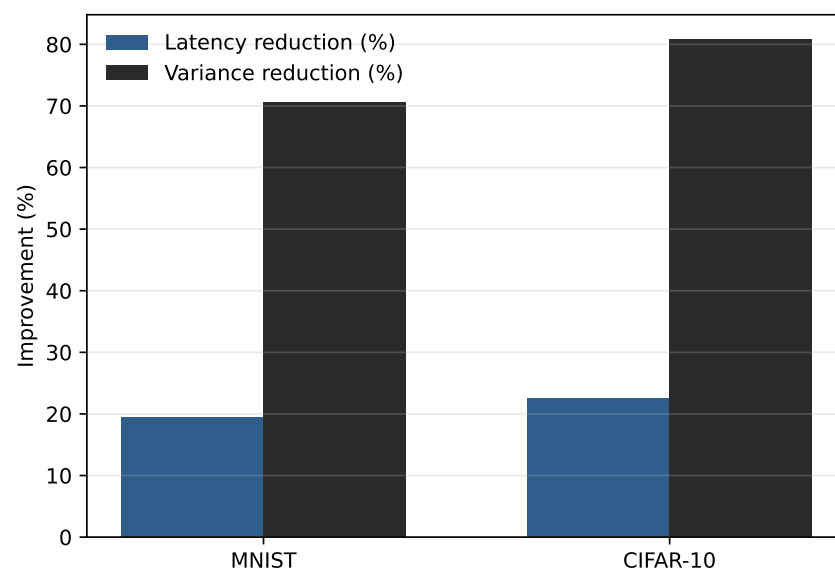


Figure 7. Relative improvement summary of DCS-RA on MNIST and CIFAR-10. The figure highlights that the method improves both mean latency and reported round time variability, with especially strong stabilization effects on the more demanding workload.

Figure 7 is especially valuable because it condenses the central empirical message of the paper into a single comparative view. The figure shows that the benefit of DCS-RA is multi-dimensional: it improves average round latency and also reduces round-to-round dispersion. This distinction matters. A scheduler that reduces only the mean while leaving variability high would still create operational uncertainty for edge services. By contrast, the present results indicate that DCS-RA simultaneously improves speed and regularity. The stronger improvement on CIFAR-10 further suggests that the proposed scheduler becomes more useful as the workload and system conditions become more challenging, which is an important argument for its relevance in heterogeneous 5G/6G edge environments.

6.11. Relation to Existing Work

The present results are directionally aligned with the broader wireless FL literature. Joint scheduling studies have repeatedly shown that coupling client participation with radio resource decisions improves training efficiency relative to naive baselines [16–19]. The present paper does not therefore claim that its contribution provides universal superiority over all existing methods; rather, it occupies a valuable position in the design space by offering an interpretable, lower-overhead policy that still captures the main sources of wireless FL delay. This positioning is important for edge deployment, where a scheduling policy must often be understandable, easy to calibrate, and computationally modest in addition to being effective. Because the experimental comparison includes only the random selection/uniform allocation baseline, comparisons with stronger prior methods remain qualitative rather than experimental. Consequently, the results should not be interpreted as evidence of state-of-the-art superiority over untested schedulers.

Table 3 compares DCS-RA with related FL, MEC resource allocation, and privacy-aware architecture studies. The comparison separates direct wireless FL scheduling studies from broader MEC and privacy-aware architecture studies so that motivational references are not incorrectly presented as performance baselines.

Table 3. Direct comparison with relevant FL, MEC resource allocation, and privacy-aware architecture studies.

Reference	Focus and Context	Resource Awareness	Privacy Role	Relation to This Work
[20]	FedCS resource-aware client selection in mobile edge FL	Client resource and deadline awareness	Raw data remain local under FL	Stronger baseline category; DCS-RA additionally couples fairness scoring with RB assignment.
[16]	Wireless IoT FL with joint client selection and resource management	CPU and transmission resource control	Raw data remain local under FL	Direct scheduling-related category.
[18]	Wireless IoT FL with joint client and wireless resource optimization	Explicit mixed-integer resource formulation	Raw data remain local under FL	Closest optimization-oriented category to DCS-RA.
[19]	Federated edge learning with imperfect CSI	Channel and resource uncertainty	Raw data remain local under FL	Highlights radio uncertainty; DCS-RA keeps a simpler interpretable rule.
[7]	Secure 5G/6G survey covering FL, blockchain, and digital twins	Discusses edge resource challenges at survey level	Security and trust are central	Motivates coexistence with security layers, not a direct scheduler.
[8]	FL and 5G/6G-enabled IoMT review	Discusses deployment and communication constraints	Healthcare privacy is central	Supports 5G/6G FL motivation beyond generic IoT.
[22]	DRL-based mobile edge resource allocation	Adaptive computing and network resource allocation	Not an FL privacy mechanism	Motivates dynamic edge resource control, not FL scheduling.
[23]	Task offloading and resource allocation in multi-server MEC	Joint offloading and resource optimization	Not an FL privacy mechanism	Supports latency-aware edge coordination.
[24,25]	Privacy-aware healthcare sensor and 5G architectures	Architecture-level deployment constraints	Privacy is central	Motivational references for privacy-aware decentralized design.
This work	Heterogeneous 5G/6G wireless FL with fairness-aware selection and RB allocation	Computation, channel, fairness pressure, and RB assignment	Raw data remain local under FL	Interpretable low-complexity scheduler; experimental comparison with stronger baselines remains required.

The full figure set strengthens this positioning further: Figure 2 explains how the scheduler operates, Figure 3 shows why such a scheduler is necessary, and Figures 4–7 demonstrate that the resulting policy materially improves system behavior. This progression from the system condition to the scheduling mechanism to empirical results substantially improves the explanatory strength of the manuscript.

6.12. Figure and Table Readability

Each figure and table is introduced close to its placement and interpreted in terms of the metric it reports and the claim it supports. Figures 1 and 2 use simplified vector-style layouts with larger labels, fewer visual elements, and clearer directional flow. The comparison table separates direct FL scheduling references from broader MEC and privacy-aware architecture references. The main latency graph reports standard deviation error

bars, and Table 2 reports the corresponding numerical uncertainty values. Confidence intervals are not reported because independent repeated-run statistics are not part of the present evaluation.

6.13. Limitations and Responsible Interpretation

The interpretation of the findings is limited to the evidence directly supported by the simulations. The experimental results support system-level latency improvement, but they do not include complete accuracy or loss trajectories for constructing a convergence plot. For this reason, the result figures focus on the metrics that are directly supported by the available statistics, namely, mean round time, variability, per-round robustness, and cumulative performance. The current evaluation also lacks experimental comparisons with stronger baselines such as FedCS-style resource-aware selection, utility-aware participant selection, deadline-aware scheduling, joint communication–computation allocation, and fairness-aware participation schedulers. In addition, the evidence does not establish non-IID robustness, weight sensitivity behavior, or quantitative fairness through participation frequency statistics. The simulated scheduler also assumes honest and timely state reporting; it does not evaluate malicious metric manipulation, scheduling metadata leakage, secure reporting, client dropout, or mid-round channel shocks. Future work should extend the evaluation by adding explicit model accuracy curves, ablation analysis of the fairness weight, non-IID data partitions, and comparisons against additional optimization-based or reinforcement learning baselines. These limitations do not weaken the present systems result; rather, they clarify the scope of the evidence and strengthen the scientific credibility of the paper.

7. Conclusions

This paper presented DCS-RA, a lightweight framework for dynamic client selection and resource block allocation in synchronous FL in heterogeneous 5G/6G edge environments. The method addresses a specific systems challenge: reducing round completion latency caused by uneven client computation capability, local workload variation, and wireless channel diversity. By combining computation-aware, channel-aware, and fairness-aware scoring with marginal latency-based RB allocation, DCS-RA provides an interpretable scheduling mechanism that is easier to deploy repeatedly than solving the full mixed-integer nonlinear formulation at every round.

Under the reported simulation setting with 100 clients and 20 resource blocks, DCS-RA reduced the average round completion time by 19.39 percent on MNIST and 22.47 percent on CIFAR-10 relative to the random baseline. The uncertainty-aware values were 1.92 ± 0.17 s versus 1.55 ± 0.05 s on MNIST and 2.02 ± 0.26 s versus 1.57 ± 0.05 s on CIFAR-10. DCS-RA also reduced reported round time variability by 70.59 percent and 80.77 percent, respectively. These results support the narrower and evidence-based conclusion that DCS-RA improves wall-clock FL efficiency and latency stability within the evaluated simulation conditions.

The study remains subject to several limitations. The current evidence is simulation-based, uses two benchmark datasets, and does not yet include node-scaling results from 50 to 150 clients, real testbed validation, hardware runtime profiling, explicit non-IID partition analysis, accuracy/loss convergence curves, weight sensitivity analysis, per-client fairness metrics, or comparisons with stronger optimization and reinforcement learning baselines. In addition, although the core scheduling logic can be used in both 5G and 6G-oriented edge systems, the practical adoption path differs because 6G may provide richer sensing, mobility prediction, and radio state observability than current 5G deployments. Future work should therefore extend the evaluation to larger heterogeneous networks, realistic fading and

mobility-induced channel dynamics, dropout-aware runtime adaptation, energy-aware objectives, secure and privacy-preserving scheduling metadata, stronger client selection and resource allocation baselines, and real 5G/6G edge testbeds. These extensions are necessary before making deployment-level claims beyond the controlled simulation environment reported here.

Author Contributions: A.L.S.A.-K. designed the research and methods, developed the software artifacts, performed the analytic work, and wrote the initial version of the paper. R.A. coordinated supervision, reviewed the methodology and analysis, and contributed to manuscript refinement. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the co-author.

Acknowledgments: The authors would like to express their gratitude to Defne Telekomünikasyon A.Ş. for its valuable technical support. The authors also acknowledge Istanbul Atlas University Laboratory for providing a research environment that facilitated this study. The authors further thank Oğuz Haliloğlu, Ercan Fincan, and Hatice Cerşit of Defne Telekomünikasyon A.Ş. for their valuable support, insightful contributions, and assistance throughout this research. During the preparation of this manuscript, the authors used ChatGPT by OpenAI (GPT-5.5 Thinking, accessed in 2026) for language editing, proofreading, grammar correction, and improving the clarity and readability of selected manuscript text. The authors reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: Ahmed Lateef Salih Al-Karawi is affiliated with Defne Telekomünikasyon A.Ş., which provided technical support acknowledged in this manuscript. The remaining author declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest

References

1. Duan, Q.; Huang, J.; Hu, S.; Deng, R.; Lu, Z.; Yu, S. Combining Federated Learning and Edge Computing toward Ubiquitous Intelligence in 6G Network: Challenges, Recent Advances, and Future Directions. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2892–2950. [\[CrossRef\]](#)
2. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Poor, H.V. Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1622–1658. [\[CrossRef\]](#)
3. Banabilah, S.; Aloqaily, M.; Alsayed, E.; Malik, N.; Jararweh, Y. Federated Learning Review: Fundamentals, Enabling Technologies, and Future Applications. *Inf. Process. Manag.* **2022**, *59*, 103061. [\[CrossRef\]](#)
4. Brecko, A.; Kajati, E.; Koziolek, J.; Zolotova, I. Federated Learning for Edge Computing: A Survey. *Appl. Sci.* **2022**, *12*, 9124. [\[CrossRef\]](#)
5. Abreha, H.G.; Hayajneh, M.; Serhani, M.A. Federated Learning in Edge Computing: A Systematic Survey. *Sensors* **2022**, *22*, 450. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Ergen, M.; Saoud, B.; Shayea, I.; El-Saleh, A.A.; Ergen, O.; Inan, F.; Tuysuz, M.F. Edge Computing in Future Wireless Networks: A Comprehensive Evaluation and Vision for 6G and Beyond. *ICT Express* **2024**, *10*, 1151–1173. [\[CrossRef\]](#)
7. Asim, M.; Ateya, A.A.; Wani, M.A.; Ali, G.; ElAffendi, M.; Abd El-Latif, A.A.; Siyal, R. A Comprehensive Survey on Blockchain-Enabled Techniques and Federated Learning for Secure 5G/6G Networks: Challenges, Opportunities, and Future Directions. *Comput. Mater. Contin.* **2026**, *86*, 3. [\[CrossRef\]](#)
8. Ahad, A.; Ahmed, K.I.; Ullah, F.; Sheikh, M.A.; Tahir, M.; Hayajneh, M.; Pires, I.M. Federated Learning and 5G/6G-Based Internet of Medical Things (IoMT): Applications, Key Enabling Technologies, Open Issues and Future Research Directions. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2026**, *16*, e70065. [\[CrossRef\]](#)
9. Xia, Q.; Ye, W.; Tao, Z.; Wu, J.; Li, Q. A Survey of Federated Learning for Edge Computing: Research Problems and Solutions. *High-Confid. Comput.* **2021**, *1*, 100008. [\[CrossRef\]](#)
10. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 1–24. [\[CrossRef\]](#)

11. Shanmugarasa, Y.; Paik, H.-y.; Kanhere, S.S.; Zhu, L. A Systematic Review of Federated Learning from Clients' Perspective: Challenges and Solutions. *Artif. Intell. Rev.* **2023**, *56*, 1773–1827. [\[CrossRef\]](#)
12. Mattoo, A.; Jain, N.; Gandhi, C. Device Selection Methods in Federated Learning: A Survey. *SN Comput. Sci.* **2024**, *5*, 763. [\[CrossRef\]](#)
13. Dritsas, E.; Trigka, M. Federated Learning for IoT: A Survey of Techniques, Challenges, and Applications. *J. Sens. Actuator Netw.* **2025**, *14*, 9. [\[CrossRef\]](#)
14. Li, J.; Chen, T.; Teng, S. A Comprehensive Survey on Client Selection Strategies in Federated Learning. *Comput. Netw.* **2024**, *251*, 110663. [\[CrossRef\]](#)
15. Salehi, M.; Hossain, E. Federated Learning in Unreliable and Resource-Constrained Cellular Wireless Networks. *IEEE Trans. Commun.* **2021**, *69*, 5136–5151. [\[CrossRef\]](#)
16. Yu, L.; Albelaihi, R.; Sun, X.; Ansari, N.; Devetsikiotis, M. Jointly Optimizing Client Selection and Resource Management in Wireless Federated Learning for Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 4385–4395. [\[CrossRef\]](#)
17. Yang, J.; Zhou, Y.; Wen, W.; Zhou, J.; Zhang, Q. Asynchronous Hierarchical Federated Learning Based on Bandwidth Allocation and Client Scheduling. *Appl. Sci.* **2023**, *13*, 11134. [\[CrossRef\]](#)
18. Zhao, J.; Ni, Y.; Cheng, Y. Joint Client and Resource Optimization for Federated Learning in Wireless IoT Networks. *Appl. Sci.* **2024**, *14*, 542. [\[CrossRef\]](#)
19. Zhou, S.; Wang, L.; Wu, W.; Feng, L. Joint Client Selection and Resource Allocation for Federated Edge Learning with Imperfect CSI. *Comput. Netw.* **2025**, *257*, 110914. [\[CrossRef\]](#)
20. Nishio, T.; Yonetani, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In Proceedings of the 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7. [\[CrossRef\]](#)
21. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
22. Wang, J.; Zhao, L.; Liu, J.; Kato, N. Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1529–1541. [\[CrossRef\]](#)
23. Tran, T.X.; Pompili, D. Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 856–868. [\[CrossRef\]](#)
24. Raso, E.; Bianco, G.M.; Bracciale, L.; Marrocco, G.; Occhiuzzi, C.; Loreti, P. Privacy-Aware Architectures for NFC and RFID Sensors in Healthcare Applications. *Sensors* **2022**, *22*, 9692. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Liyanage, M.; Salo, J.; Braeken, A.; Kumar, T.; Seneviratne, S.; Ylianttila, M. 5G Privacy: Scenarios and Solutions. In Proceedings of the 2018 IEEE 5G World Forum (5GWF), Santa Clara, CA, USA, 9–11 July 2018; pp. 197–203. [\[CrossRef\]](#)
26. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.