Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.069873

ARTICLE

FedCW: Client Selection with Adaptive Weight in Heterogeneous Federated Learning

Haotian Wu[1], **Jiaming Pei**[2] **and Jinhai Li**[3,*]

[1]School of Computer Science, Torrens University Australia, Sydney, NSW 2007, Australia
[2]School of Computer Science, The University of Sydney, Camperdown, Sydney, NSW 2006, Australia
[3]College of Economics and Management, Taizhou University, Taizhou, 225300, China

*Corresponding Author: Jinhai Li. Email: lijinhai@tzu.edu.cn

Received: 02 July 2025; Accepted: 29 September 2025; Published: 10 November 2025

ABSTRACT: With the increasing complexity of vehicular networks and the proliferation of connected vehicles, Federated Learning (FL) has emerged as a critical framework for decentralized model training while preserving data privacy. However, efficient client selection and adaptive weight allocation in heterogeneous and non-IID environments remain challenging. To address these issues, we propose Federated Learning with Client Selection and Adaptive Weighting (FedCW), a novel algorithm that leverages adaptive client selection and dynamic weight allocation for optimizing model convergence in real-time vehicular networks. FedCW selects clients based on their Euclidean distance from the global model and dynamically adjusts aggregation weights to optimize both data diversity and model convergence. Experimental results show that FedCW significantly outperforms existing FL algorithms such as FedAvg, FedProx, and SCAFFOLD, particularly in non-IID settings, achieving faster convergence, higher accuracy, and reduced communication overhead. These findings demonstrate that FedCW provides an effective solution for enhancing the performance of FL in heterogeneous, edge-based computing environments.

KEYWORDS: Federated learning; non-IID; client selection; weight allocation; vehicular networks

1 Introduction

Federated Learning (FL) [1] has emerged as a promising paradigm to enable distributed model training across multiple devices without the need to centralize data [2]. This characteristic is especially appealing in vehicular networks, where data privacy and communication costs are significant concerns [3,4]. With an increasing number of connected vehicles, FL can provide an efficient solution to collaboratively train models, such as those needed for traffic prediction, autonomous driving, and vehicular safety, while preserving individual data privacy [5]. The decentralized nature of FL reduces the risk of data breaches and ensures that sensitive information remains on local devices, which is particularly important for vehicular networks, where data privacy regulations are stringent, and the costs of data transmission are high [6]. However, the inherent heterogeneity in vehicular data and the variance in computational capabilities across different clients pose significant challenges to conventional FL algorithms [7–9].

A fundamental issue faced by traditional FL algorithms in vehicular networks is their inability to effectively address the heterogeneity of data and client capabilities. The variability in data distribution across clients, which is frequently non-independent and non-identically distributed (Non-IID), can significantly impede convergence and lead to biased global models [10–12]. In vehicular networks, data collected by

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

individual vehicles may differ markedly due to factors such as geographic location, driving behavior, and environmental conditions, further complicating the learning process. Additionally, the random selection of clients without accounting for their potential contributions, coupled with static weight allocation that disregards clients' relevance to the global model, exacerbates these challenges, particularly in resource-constrained vehicular contexts [13,14]. Consequently, the global model may converge slowly or fail to generalize effectively across all clients, resulting in suboptimal performance in real-world deployments. These challenges underscore a significant research gap in the domain of FL for vehicular networks. In Federated Learning, two central problems are client selection and the appropriate distribution of aggregation weights, particularly in heterogeneous environments where devices have diverse computational capacities and data distributions. The challenge of client selection lies in determining which edge devices should participate in each training round, as involving all clients can be computationally expensive and inefficient. Similarly, the aggregation of model updates must reflect the varying importance of contributions from different clients, considering that clients with more representative data or superior computational resources should have a greater influence on the global model. Fundamentally, both issues revolve around the optimal allocation of weights: client selection can be interpreted as assigning binary weights (0 or 1) to clients to determine their participation, while weight aggregation entails determining the proportional contribution of each client's update. Studies have shown that unbalanced client participation and inappropriate weighting can detrimentally impact model convergence and accuracy. However, existing methods often treat client selection and weight aggregation independently, resulting in suboptimal outcomes that fail to capitalize on the interconnected nature of these processes. A holistic approach that jointly optimizes both client selection and weight aggregation could yield more efficient and accurate federated learning in heterogeneous edge environments.

In this paper, we aim to unify these two subproblems by presenting a new algorithm, FedCW (Federated Learning with Client Selection and Adaptive Weighting). This algorithm enhances Federated Learning in two key stages: client selection and weight allocation. we calculate the Euclidean distance between each client's local model and the global model, prioritizing clients with greater divergence for selection. By dynamically adjusting the number of clients involved in each round and allocating weights based on both data volume and model divergence, the algorithm ensures that clients contributing more significant updates have a larger impact on the global model. This integrated approach simultaneously addresses both challenges, optimizing the learning process and improving global model convergence in heterogeneous federated learning environments. To validate our approach, we conduct experiments on various datasets, including MNIST, CIFAR-10, ImageNet-100, and CelebA. We evaluate the performance of FedCW against state-of-the-art algorithms. The experimental results demonstrate that FedCW significantly improves global model accuracy and convergence speed, especially in non-IID data scenarios. In the following, we provide a summary of contributions:

- We integrate client selection and weight allocation into a unified optimization framework for Federated Learning. This approach effectively addresses the interdependent challenges of heterogeneous client environments, improving both the efficiency and accuracy of model training.
- We introduce a novel client selection mechanism that prioritizes clients with diverse updates based on their Euclidean distance from the global model. Simultaneously, FedCW dynamically adjusts aggregation weights according to both data volume and model divergence, enhancing model convergence and generalization in non-IID environments.
- We demonstrate that FedCW significantly outperforms existing algorithms through extensive experiments. Our method achieves faster convergence, higher model accuracy, and reduced communication overhead, particularly in non-IID federated learning scenarios.

2 Related Works

Client selection and weight allocation are considered open optimization problems in federated learning (FL), driven by the constraints of limited bandwidth and the necessity for selected participation in training. The problem of client selection is regarded as an ongoing optimization challenge within the FL process. Initial discussions on client selection, such as those by [15], introduced the concept of selecting clients based on their resource efficiency to enhance training convergence and model accuracy. This client selection paradigm has been foundational, sparking widespread research into optimizing FL with heterogeneous resources. However, choosing clients based merely on performance metrics such as speed or computational resources might lead to biases, potentially compromising data diversity. This challenge was addressed by [16], who provided a convergence analysis for client selection, advocating for a balanced approach that considers both computational and communication efficiencies. Further developments by [17] introduced a hybrid FL mechanism that aims to increase the number of participating clients and thus improve the accuracy of the aggregated model by enhancing the diversity of client data used during training. On the other hand, reference [18] explored dynamic data profiling to optimize client selection further, suggesting that understanding data characteristics dynamically could lead to more informed client choices in FL environments. Additionally, recent studies have proposed various frameworks to refine client selection further. For instance, reference [19]'s VFedCS framework focuses on optimizing client selection to manage the volatility and dynamic nature of client availability and resource allocation in federated settings. At the same time, recent vehicular/ITS surveys observe that most systems still *decouple* the decision of *who* participates from *how much* their updates should count during aggregation, which can entrench selection bias under non-IID and volatile participation [20,21].

Several algorithms have been developed to improve the aggregation phase in federated learning. Reference [22] proposed FedProx, which introduces a proximal term to the objective function, mitigating the impact of local updates diverging from the global model. This modification leads to more robust convergence in heterogeneous environments compared to FedAvg, particularly when devices perform a variable amount of local work. To tackle the client-drift problem caused by non-IID data, reference [23] introduced SCAFFOLD, which incorporates control variates to correct local updates. By reducing the variance of updates, SCAFFOLD improves the accuracy of the global model while requiring fewer communication rounds. In the domain of adaptive optimization, reference [24] presented FedOpt, which applies server-side optimization techniques such as Adam and Yogi during the aggregation process, leading to better performance in heterogeneous client environments. Additionally, reference [11] proposed FedNova, which normalizes local updates to eliminate objective inconsistency, ensuring that the global model converges to a solution of the correct objective function despite differences in local workloads. Reference [25] introduced FedMA, a novel aggregation technique that constructs a global model by matching and averaging neurons across layers of client models, effectively aligning the structure of local neural networks. This method enhances performance and reduces communication costs, particularly for complex neural network architectures such as CNNs and LSTMs. Nevertheless, these aggregation methods typically assume uniform or data-size-proportional weights and *random/greedy* participation, which *implicitly presumes* representativeness of the selected clients; vehicular studies indicate this assumption often breaks under mobility, skewed labels, and intermittent connectivity [20,21].

Security-oriented ITS works adopt FL for intrusion detection in in-/inter-vehicle networks, confirming feasibility but also revealing robustness and privacy tensions when participation is static and weights are near-uniform [26,27]. Resource-latency studies in vehicular edge computing (VEC) emphasize that client scheduling/offloading co-determines both spectral efficiency and learning dynamics; misaligned aggregation

can offset scheduling benefits [28,29]. Communication-efficient FL with gradient quantization and DRL-based compression reduces bandwidth, yet injects additional noise/bias that calls for *adaptive* weighting and selection to prevent oscillation under harsh non-IID [30]. Application-driven ITS systems—traffic management and mobility/location prediction—report gains with FL but still rely on static participation and weighting, limiting resilience to churn and distribution drift [31,32].

Across client selection, aggregation, and vehicular deployments, three gaps emerge: (i) *selection-aggregation decoupling*—who participates is decided without calibrating how strongly their updates should influence the global model; (ii) *sensitivity to non-IID and volatility*—static policies amplify bias when participation is skewed or data distributions drift; (iii) *engineering blind spots*—scheduling/offloading and compression alter update statistics, yet server-side aggregation remains oblivious to these shifts.

Although the aforementioned works focus on client selection or weight allocation, they treat these as separate issues rather than addressing them as a unified problem. In this paper, we integrate both subproblems into a global optimization framework and propose the FedCW algorithm, which simultaneously tackles both challenges. Through extensive experiments, we compare our method with several of the aforementioned algorithms and demonstrate that FedCW effectively improves global accuracy and accelerates convergence.

3 Preliminaries

Training Process

FL involves multiple clients that train on locally generated data and periodically communicate with a global server to aggregate models. Specifically, the training process in each round comprises the following steps:

- **Initialize (Round 0):** The global model is initialized on the server side and then distributed to all participating clients.
- **Client Selection and Model Distribution:** The server selects the client according to the provided client model and information, and sends the aggregated global model to the selected client.
- **Local Training:** Each selected client trains the received global model on their local data. This step involves running several epochs of training to adjust the model parameters based on the client's unique data distribution.
- **Model Transmission:** each client sends their updated local model back to the server after local training.
- **Model Aggregation:** The server receives the successfully returned model and aggregates the updates to the new global model based on this algorithm.
- **Repeat Process:** The new round of training begins with the client selection step, repeating steps 2 to 5 until the desired convergence or accuracy is achieved.

To graphically understand FL's training process, we refer the reader to the Fig, which demonstrates each step of the training process through examples.

4 Problem Formulation

Global Objective

In the standard FL framework, our objective is to minimize the global loss function $F(w)$, which for a fixed set of clients over T rounds. It can be expressed as:

$$\min_{w} \sum_{k=1}^{K} p_k F_k(w) \tag{1}$$

where $F_k(w)$ is the local loss function of the k-th client, $p_k = (|D(k)|]/D)$ which is the weight of the k-th client's data relative to the entire dataset, and w represents the parameters of the global model.

In our proposed method, the weights p_k are no longer fixed or uniformly distributed among the clients. Instead, they are dynamically adjusted based on the Euclidean distance d_k between the local model parameters w_k of the k-th client and the global model parameters w. The modified weight p_k is defined as:

$$p_k = \frac{\exp(-\lambda d_k)}{\sum_{j=1}^{K} \exp(-\lambda d_j)} \tag{2}$$

where λ is a hyperparameter controlling the rate of exponential decay, and d_k is the Euclidean distance between the local model of the k-th client and the global model. This formulation ensures that clients with models that are further from the global model have a larger influence on the model update, potentially accelerating convergence by incorporating more diverse updates into the global model. The updated global objective is then:

$$\min_{w} \quad F(w) = \sum_{k=1}^{K} \left(\frac{\exp(-\lambda d_k)}{\sum_{j=1}^{K} \exp(-\lambda d_j)} \right) F_k(w) \tag{3}$$

where:

- $F_k(w)$ is the local loss function for the k-th client.
- w represents the parameters of the global model.
- d_k is the Euclidean distance between the local model parameters w_k of the k-th client and the global model parameters w.
- λ is a hyperparameter that controls the rate of exponential decay, influencing how the distances affect the weights.

5 Client Selection with Adaptive Weight in FL

In this section, we introduce an enhanced FL algorithm. It can be conceptually divided into two stages: client selection and weight allocation. Algorithm 1 illustrates the operational procedure of our method. In the following, we will elucidate how Euclidean distances influence client selection and weight Allocation; we will also discuss how these elements integrate to form a cohesive whole.

Algorithm 1: Enhanced federated learning with euclidean distance based client selection

Input: Total number of clients N, number of rounds T, initial global model w^0, hyperparameters λ (decay rate), β (weight parameter), α (initial selection fraction), min_clients (minimum number of participants)
Output: Final global model w^T
if $t = 0$ **then**
    Initialize: All clients participate
    $w_k^0 \leftarrow$ Train on all clients
    $w^1 \leftarrow \frac{1}{N} \sum_{k=1}^{N} w_k^0$
    Compute Euclidean distances d_k for all k
    Sort clients by d_k descending
    Initialize hyperparameters λ, β, α, min_clients

(Continued)

Algorithm 1 (continued)

end if
for $t = 1$ to T **do**
  Client Selection:
  if $t > 0$ **then**
    Select clients based on sorted d_k with participation fraction $\alpha_t = \alpha \times \exp(-\lambda t)$ and minimum min_clients
    $w_k^t \leftarrow$ Train on selected clients
    Aggregate: $w^{t+1} = \sum_{k=1}^{K} a_k^t \, w_k^t$ where $a_k^t \propto n_k \cdot \exp(\beta d_k)$
    Post-Aggregation:
    Compute distances d_k for clients updated this round
    Sort clients by d_k descending
  end if
  Repeat: Go to Client Selection
end for

5.1 Euclidean Distance Calculation

The server typically possesses significantly greater computational power and better resource allocation than individual clients. This enables efficient execution of complex mathematical computations, especially when involving multiple clients. Furthermore, performing these calculations on the server side also reduces the communication overhead by eliminating the need to transmit the computed distances. Therefore, starting from Round 1, server will compute the Euclidean distance from each client model to the current global model once the server receives and aggregates the models from all participating clients. The calculation is formulated as follows:

$$d_i = \sqrt{\sum_k (w_k - w_{i,k})^2} \tag{4}$$

The equation defined above calculates the Euclidean distance d_i between the global model and the model parameters of the i-th client. Here is a breakdown of the components within the equation:

- w_k represents the parameter vector of the global model. These parameters are aggregated values derived from the previous round of updates received from all or a subset of clients.
- $w_{i,k}$ denotes the parameter vector of the i-th client's model following local training on its dataset.

After the calculation is complete, the server will sort the d_i from largest to smallest. We will explain why later.

5.2 Client Selection

In this approach, we select clients based on a descending order of their Euclidean distance from the global model. Intuitively, selecting clients with the greatest divergence from the global model (i.e., the largest Euclidean distance) can introduce more information and data diversity. This is beneficial for the global model as it aids in learning and adapting to a broader data distribution, thereby reducing the risk of over-fitting. By prioritizing clients whose models significantly differ from the global model, each iteration can incorporate more substantial updates, potentially allowing the global model to adapt more quickly to the entire data distribution. We introduce a decay function to dynamically adjust the number of clients selected in each round. The decay function allows the algorithm to adjust the number of participating clients dynamically

based on the progress of training or the performance of clients. This method can specifically increase or decrease the number of clients involved in training at different stages to meet the needs of the model during the training process. Given an initial selection fraction α and a decay rate λ, the client selection fraction in round t is:

$$\alpha_t = \alpha \times \exp(-\lambda t) \tag{5}$$

Adjusting the number of participating clients can more effectively utilize limited computational and communication resources. Initially, more clients may need to be involved to enhance the diversity of the model, while later in training, the number might be reduced to concentrate resources on optimizing a model that is nearing convergence. This method, compared to randomly selecting a fixed number of clients, improves the algorithm's adaptability to different training stages and network conditions, particularly when facing non-IID data distributions. Let N be the total number of clients. The number of clients selected in round t is:

$$n_t = \max(\lceil N \times \alpha_t \rceil, \text{min\_clients}) \tag{6}$$

where min_clients is a hyperparameter representing the minimum number of clients participating in each round.

To ensure the robustness and efficacy of the algorithm, especially when the total number of clients N is large and the proportion α_t might be set relatively small. This parameter ensures sufficient model update diversity: it prevents inadequate model updates or excessive bias due to too few participating clients. It also ensures that enough clients participate to maintain the continuity and stability of model training, even if some clients may be unavailable due to technical issues.

5.3 Weight Allocation

In the FedAvg algorithm, the aggregation formula for updating the global model is traditionally given by:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t \tag{7}$$

This formula ensures that each client's contribution to the global model is weighted by the proportion of data they hold relative to the total dataset. However, the sheer volume of data does not always reflect a client's potential impact on the model. Therefore, in our algorithm, the weight allocation formula is modified as follows:

$$w_k = \frac{n_k \cdot \exp(\beta d_k)}{\sum_{i=1}^{K} n_i \cdot \exp(\beta d_i)} \tag{8}$$

- n_k is the size of the local dataset of client k.
- d_k is the Euclidean distance between the local model of client k and the global model.
- β is an adjustment parameter that controls the influence of the Euclidean distance.
- K is the number of participating clients.

Then the global model parameters are updated as follows:

$$w_{t+1} = \sum_{k=1}^{K} w_k w_k^{(t)} \tag{9}$$

This weight allocation formula not only considers the volume of data each client contributes but also the divergence of each client's model from the global model. It introduces a corrective mechanism that allows models with greater Euclidean distances (i.e., more divergent client models) to have a more significant voice in the aggregation process. By employing an exponential function regulated by β, we can flexibly adjust the impact of the Euclidean distance. This provides a method to balance the respect between data-rich clients and those clients whose model parameters significantly differ. The denominator $\sum_{i=1}^{K} n_i \cdot \exp(\beta d_i)$ acts as a normalization factor, ensuring that the sum of all client weights equals 1, thus maintaining stability in the model update steps. By combining data volume and model divergence, the algorithm can balance the influence of different clients on the global model, utilizing the information provided by data-rich clients while preventing the model from deviating from the optimization path due to the excessive influence of outlier clients.

When β is set to a positive value, it implies that as the Euclidean distance d_k between a client's local model and the global model increases, so does their weight in the model update. In this setting, as d_k increases (i.e., the greater the divergence of the client model from the global model), the greater their influence on the global model. This approach prioritizes clients with substantial differences from the global model, theoretically assisting the global model to quickly adapt and learn from the unique data patterns represented by a minority of clients. Applicable scenarios: This is particularly suitable in situations where the global model needs to learn from significant differences in client models, especially in cases of highly non-independent and identically distributed (Non-IID) data, helping to enhance the model's diversity and generalization capabilities.

When β is set to a negative value, the weights decrease as d_k increases. This means that clients with smaller differences from the global model gain greater weight, aiding the global model to learn more robustly towards the central tendency of the data, reducing fluctuations caused by extreme or deviating clients.

This method of weight allocation can enhance the robustness and generalization ability of the global model in federated learning. It ensures that all relevant characteristics, such as data volume and model uniqueness, are appropriately considered in weighting each client's contribution. This method is particularly effective in heterogeneous environments, as client data distributions and system characteristics often vary greatly in reality. We will demonstrate this in the experiments.

5.4 Synergy of Client Selection and Weight Allocation

In this algorithm, client selection and weight allocation are two interdependent components that together determine the efficiency and quality of the global model updates. They are less effective when existing alone while these components can be understood independently. And they complement each other when integrated, achieving optimal learning outcomes.

Limitations of Client Selection: Client selection alone, without appropriate weight allocation, may lead to an over-reliance on certain clients' data. Particularly in scenarios where client data is highly imbalanced, selecting clients with larger data volumes can result in a model bias towards these clients' characteristics, overlooking other important but smaller data volume clients. Purely basing client selection on criteria such as data volume or update frequency may neglect the differences between client models and the global model, which is detrimental to capturing the diversity of data across the entire network.

Limitations of Weight Allocation: If there is only weight allocation without an effective client selection mechanism, then weights might be allocated to clients that contribute minimally to model updates. For instance, clients with minimal divergence from the global model might receive higher weights, but these minor updates may not be sufficient to significantly enhance the model. Relying solely on weight allocation

to handle all clients can lead to inefficient computations, especially when the number of clients is very large, resulting in significant computational and communication overhead that is largely unnecessary.

Client selection can reduce the number of clients that need to be processed each round, thereby reducing computational and communication burdens. On the other hand, weight allocation ensures that the maximum learning benefit is extracted from these selected clients, particularly by weighting updates from those clients who differ significantly from the global model, thereby accelerating model convergence. This combined strategy effectively learns from the data across the entire network, avoiding model over-fitting to specific client data and thus improving the model's performance on unseen data.

5.5 Communication Complexity

FedCW requires each selected client to send its local model parameters to the server as in standard FL. The server computes the Euclidean distance d_k for all participating clients, with complexity $O(Nd)$ for N clients and d-dimensional parameters, followed by sorting in $O(N \log N)$. These costs are negligible compared to the communication of model parameters. Furthermore, since FedCW dynamically reduces the number of participating clients via α_t, the overall communication overhead is significantly reduced compared to FedAvg, as also validated in our experiments.

5.6 Security and Privacy Considerations

While FedCW demonstrates clear advantages in convergence and communication efficiency, its deployment in real-world vehicular networks also raises important concerns related to security, privacy, and system-level robustness. In particular, the Euclidean distance ranking strategy, although effective for selecting diverse clients, may inadvertently expose distributional characteristics of local model updates, thereby creating potential side-channel privacy risks. To mitigate this, future implementations of FedCW could be combined with secure aggregation protocols or differential privacy mechanisms that protect individual updates while still allowing the server to compute distance-based rankings. Another issue is the vulnerability of distance-based selection and adaptive weighting to malicious or Byzantine clients, who may inject manipulated updates to distort the aggregation process. Addressing this challenge requires the integration of anomaly detection techniques or Byzantine-resilient aggregation rules that can identify and suppress adversarial contributions without significantly increasing communication overhead. Beyond adversarial risks, large-scale vehicular deployments often encounter system bottlenecks such as channel congestion, synchronization delays, and frequent client churn, which can compromise the effectiveness of synchronous distance-based selection. To address these challenges, FedCW can be extended with asynchronous update strategies or hierarchical aggregation frameworks, where edge servers first coordinate subsets of vehicles before forwarding updates to the global server, thereby reducing latency and improving scalability.

5.7 Hyperparameter Discussion

The hyperparameters λ and β play central roles in balancing convergence and stability. The parameter λ controls the decay rate of client participation: larger λ leads to fewer clients in later rounds, reducing communication but potentially harming diversity, while smaller λ retains more clients, improving robustness but increasing overhead. The parameter β regulates the impact of Euclidean distance in weight allocation: positive β emphasizes clients with divergent models, accelerating adaptation to non-IID data but risking instability when extreme; negative β favors alignment with the global model, improving stability but slowing adaptation. Hence, λ and β jointly determine the trade-off between convergence speed, stability, and communication cost, and must be tuned according to heterogeneity levels and system constraints.

5.8 Theorem Analysis

Notation and update rule.

Let $\mathcal{K}_t$ be the set of selected clients at round t. Denote server iterate by w^t and client k's local iterate after its local training by w_k^t. Let $a_k^t \geq 0$ be the server aggregation weights with $\sum_{k\in\mathcal{K}_t} a_k^t = 1$ (in FedCW, $a_k^t \propto n_k \exp(\beta d_k)$). Define the aggregated direction

$$g_t \;=\; \sum_{k\in\mathcal{K}_t} a_k^t \,\nabla f_k(w_k^t). \tag{10}$$

The server performs the update

$$w^{t+1} \;=\; w^t \;-\; \eta_t\, g_t, \tag{11}$$

where $\eta_t > 0$ is the stepsize.

Assumption 1 *(Smoothness): Each f_k is L-smooth, i.e., for all w_1, w_2,*

$$\|\nabla f_k(w_1) - \nabla f_k(w_2)\| \;\leq\; L\|w_1 - w_2\|. \tag{12}$$

Consequently, the global objective $F(w) = \frac{1}{N}\sum_{k=1}^{N} f_k(w)$ is also L-smooth.

Assumption 2 *(Bounded gradients and variance): There exists $G > 0$ such that for all k, w,*

$$\|\nabla f_k(w)\| \;\leq\; G. \tag{13}$$

Let $\zeta_t := g_t - \mathbb{E}[g_t \,|\, w^t]$ be the zero-mean noise conditioned on w^t. There exists $\sigma^2 \geq 0$ such that

$$\mathbb{E}\big[\|\zeta_t\|^2 \,\big|\, w^t\big] \;\leq\; \sigma^2. \tag{14}$$

Assumption 3 *(Selection/weighting and local-drift bias): Define the conditional-mean decomposition at the server point w^t:*

$$\mathbb{E}[g_t \,|\, w^t] \;=\; \nabla F(w^t) \;+\; \varepsilon_t, \tag{15}$$

where ε_t collects (i) selection/weighting mismatch ($\mathcal{K}_t \subset [N]$, non-uniform a_k^t) and (ii) local drift because $\nabla f_k(w_k^t)$ is evaluated at $w_k^t \neq w^t$. Assume the average bias energy is bounded:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\varepsilon_t\|^2 \;\leq\; \overline{\varepsilon^2}. \tag{16}$$

Step 1: Descent lemma (exact algebra, no convexity).

By L-smoothness of F and the update (11),

$$\begin{aligned} F(w^{t+1}) &\leq F(w^t) + \langle \nabla F(w^t),\, w^{t+1} - w^t\rangle + \frac{L}{2}\|w^{t+1} - w^t\|^2 \\ &= F(w^t) - \eta_t \,\langle \nabla F(w^t), g_t\rangle + \frac{L\eta_t^2}{2}\,\|g_t\|^2. \end{aligned} \tag{17}$$

Step 2: Decompose g_t and expand the inner product.

From (15), write

$$g_t \;=\; \nabla F(w^t) \;+\; \varepsilon_t \;+\; \zeta_t, \qquad \mathbb{E}[\zeta_t \,|\, w^t] = 0. \tag{18}$$

Then

$$\begin{aligned}\langle \nabla F(w^t), g_t\rangle &= \langle \nabla F(w^t), \nabla F(w^t)\rangle + \langle \nabla F(w^t), \varepsilon_t\rangle + \langle \nabla F(w^t), \zeta_t\rangle \\ &= \|\nabla F(w^t)\|^2 + \langle \nabla F(w^t), \varepsilon_t\rangle + \langle \nabla F(w^t), \zeta_t\rangle.\end{aligned} \tag{19}$$

Taking conditional expectation given w^t, the noise term vanishes:

$$\mathbb{E}\big[\langle \nabla F(w^t), \zeta_t\rangle \,\big|\, w^t\big] = 0. \tag{20}$$

Using Young's inequality $\langle a, b\rangle \geq -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2$ with $a = \nabla F(w^t)$ and $b = \varepsilon_t$ gives

$$-\langle \nabla F(w^t), \varepsilon_t\rangle \;\leq\; \frac{1}{2}\|\nabla F(w^t)\|^2 + \frac{1}{2}\|\varepsilon_t\|^2. \tag{21}$$

Multiplying (19) by $-\eta_t$ and taking expectation yields

$$\mathbb{E}\big[-\eta_t\langle \nabla F(w^t), g_t\rangle\big] \;\leq\; -\frac{\eta_t}{2}\,\mathbb{E}\|\nabla F(w^t)\|^2 \;+\; \frac{\eta_t}{2}\,\mathbb{E}\|\varepsilon_t\|^2. \tag{22}$$

Step 3: Bound the quadratic term $\|g_t\|^2$.

From (18) and $(x+y+z)^2 \leq 3(x^2+y^2+z^2)$,

$$\|g_t\|^2 \;\leq\; 3\|\nabla F(w^t)\|^2 \;+\; 3\|\varepsilon_t\|^2 \;+\; 3\|\zeta_t\|^2. \tag{23}$$

Taking expectation and using (14),

$$\mathbb{E}\|g_t\|^2 \;\leq\; 3\,\mathbb{E}\|\nabla F(w^t)\|^2 \;+\; 3\,\mathbb{E}\|\varepsilon_t\|^2 \;+\; 3\,\sigma^2. \tag{24}$$

Step 4: One-step expected descent inequality.

Taking expectation of (17) and substituting (22) and (24),

$$\begin{aligned}\mathbb{E}F(w^{t+1}) \leq\; &\mathbb{E}F(w^t) - \frac{\eta_t}{2}\,\mathbb{E}\|\nabla F(w^t)\|^2 + \frac{\eta_t}{2}\,\mathbb{E}\|\varepsilon_t\|^2 \\ &+ \frac{L\eta_t^2}{2}\Big(3\,\mathbb{E}\|\nabla F(w^t)\|^2 + 3\,\mathbb{E}\|\varepsilon_t\|^2 + 3\,\sigma^2\Big).\end{aligned} \tag{25}$$

Group coefficients of each term:

$$\mathbb{E}F(w^{t+1}) \;\leq\; \mathbb{E}F(w^t) - \Big(\tfrac{\eta_t}{2} - \tfrac{3L\eta_t^2}{2}\Big)\mathbb{E}\|\nabla F(w^t)\|^2 + \Big(\tfrac{\eta_t}{2} + \tfrac{3L\eta_t^2}{2}\Big)\mathbb{E}\|\varepsilon_t\|^2 + \tfrac{3L\eta_t^2}{2}\sigma^2. \tag{26}$$

Step 5: Stepsize choice and telescoping.

Choose a constant stepsize satisfying

$$\eta_t \equiv \eta \;\leq\; \frac{1}{6L} \quad\Longrightarrow\quad \frac{\eta}{2} - \frac{3L\eta^2}{2} \;\geq\; \frac{\eta}{4}, \qquad \frac{\eta}{2} + \frac{3L\eta^2}{2} \;\leq\; \frac{3\eta}{4}. \tag{27}$$

Then (26) becomes

$$\mathbb{E}F(w^{t+1}) \;\leq\; \mathbb{E}F(w^t) - \frac{\eta}{4}\,\mathbb{E}\|\nabla F(w^t)\|^2 + \frac{3\eta}{4}\,\mathbb{E}\|\varepsilon_t\|^2 + \frac{3L\eta^2}{2}\,\sigma^2. \tag{28}$$

Summing (28) over $t = 0, \ldots, T-1$ and telescoping gives

$$\frac{\eta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \;\le\; \mathbb{E}F(w^0) - \mathbb{E}F(w^T) + \frac{3\eta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\varepsilon_t\|^2 + \frac{3L\eta^2}{2}\,T\,\sigma^2. \tag{29}$$

Using $F(w^T) \ge F^\star$ and dividing by T yields the ergodic stationarity bound

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \;\le\; \frac{4\,(F(w^0) - F^\star)}{\eta T} \;+\; 3\,\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\varepsilon_t\|^2 \;+\; 6L\,\eta\,\sigma^2. \tag{30}$$

Theorem 1 *(Stationarity with selection/weighting bias): Under (12)–(16) and $\eta \le 1/(6L)$,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \;\le\; \frac{4\,(F(w^0) - F^\star)}{\eta T} \;+\; 3\,\overline{\varepsilon^2} \;+\; 6L\,\eta\,\sigma^2. \tag{31}$$

Remark 1 *(Interpretation and roles of λ, β): In FedCW, the bias ε_t is controlled by the coverage of selected clients and the skewness of weights a_k^t. A larger decay rate λ (fewer clients in later rounds) or a larger $|\beta|$ (heavier emphasis on d_k) typically increases $\|\varepsilon_t\|$, enlarging the $3\overline{\varepsilon^2}$ term and slowing convergence (or enlarging the neighborhood). Conversely, ensuring a minimal selection size and tempering $|\beta|$ reduce $\overline{\varepsilon^2}$.*

Corollary 1 *(Convex case): If, additionally, F is convex and η_t is diminishing with $\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$, and $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\varepsilon_t\|^2 \to 0$, then every limit point of $\{w^t\}$ is stationary; for convex F, stationary points are globally optimal.*

Additional nonconvex result: iteration complexity and neighborhood size

Goal.

We turn (30) into an explicit *iteration complexity* statement under nonconvexity by optimizing the stepsize. Let $A := 4\,(F(w^0) - F^\star)$ and $B := 3\,\overline{\varepsilon^2}$. Then for constant $\eta \le 1/(6L)$,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \;\le\; \frac{A}{\eta T} \;+\; B \;+\; 6L\,\eta\,\sigma^2. \tag{32}$$

Optimizing the constant stepsize.

For fixed T, the right-hand side of (32) as a function of η is

$$\Phi(\eta) \;=\; \frac{A}{\eta T} \;+\; 6L\,\eta\,\sigma^2 \;+\; B, \qquad \eta \in \Big(0, \frac{1}{6L}\Big]. \tag{33}$$

Ignoring the box constraint momentarily, the unconstrained minimizer solves $-\frac{A}{\eta^2 T} + 6L\sigma^2 = 0$, i.e.,

$$\eta^\star \;=\; \sqrt{\frac{A}{6L\sigma^2 T}}. \tag{34}$$

If $\eta^\star \le \frac{1}{6L}$, plugging (34) back into (32) yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \;\le\; 2\sqrt{\frac{6L\,A\,\sigma^2}{T}} \;+\; B \;=\; \underbrace{\frac{C}{\sqrt{T}}}_{\text{stochastic term}} \;+\; \underbrace{B}_{\text{bias floor}}, \tag{35}$$

where $C := 2\sqrt{6L\,A\,\sigma^2}$. Thus the ergodic stationarity measure decays as $O(1/\sqrt{T})$ until it hits the *bias floor* $B = 3\overline{\varepsilon^2}$.

Reaching an ϵ-stationary point (ergodic).

Given a target $\epsilon > 0$, to ensure

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \le \epsilon^2, \tag{36}$$

we need $\epsilon^2 > B$ and

$$\frac{C}{\sqrt{T}} \le \epsilon^2 - B \quad \Longleftrightarrow \quad T \ge \frac{C^2}{(\epsilon^2 - B)^2} = \frac{24\,L\,(F(w^0) - F^\star)\,\sigma^2}{(\epsilon^2 - B)^2}. \tag{37}$$

Hence, under nonconvexity and stochasticity, FedCW attains an ϵ-stationarity level in $T = O(1/(\epsilon^2 - B)^2)$ rounds, provided the bias floor $B = 3\overline{\varepsilon^2}$ is below ϵ^2. Reducing B (e.g., by ensuring a minimum selection size and tempering $|\beta|$) directly improves the attainable accuracy and the iteration complexity.

When $\boldsymbol{\eta^\star > 1/(6L)}$.

If (34) violates the box constraint, set $\eta = 1/(6L)$ in (32) to obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \le \frac{24L(F(w^0) - F^\star)}{T} + B + \frac{\sigma^2}{L}, \tag{38}$$

which shows $O(1/T)$ decay of the optimization term plus a fixed variance floor σ^2/L and the bias floor B.

Diminishing stepsizes.

Alternatively, choosing $\eta_t = \eta_0/\sqrt{t+1}$ (with η_0 small enough) yields the classical nonconvex rate

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \le O\Big(\frac{1}{\sqrt{T}}\Big) + B, \tag{39}$$

again saturating at the bias floor B.

Remark 2 *(Practical knobs): The bias floor $B = 3\overline{\varepsilon^2}$ shrinks when (i) the per-round selection size does not become too small (controlled by λ), and (ii) the weight skewness is moderated (controlled by $|\beta|$). These knobs trade off communication/computation with convergence speed and final accuracy. Moreover, λ directly controls the number of participating clients per round: a larger λ lowers communication overhead but increases the risk of slower convergence due to reduced diversity, while a smaller λ improves convergence at the cost of higher communication. Similarly, $|\beta|$ determines the emphasis on divergent updates: larger values may accelerate convergence under moderate non-IID but risk instability under extreme heterogeneity, which could increase the total number of communication rounds required. Therefore, the choice of λ and β balances communication cost against convergence efficiency in non-IID environments.*

6 Experiment

In this section, we will introduce the experiment setup and the results.

6.1 Experiment Set

The experiments were conducted on a machine equipped with an AMD Ryzen 5600X processor, 32 GB DDR4 RAM, a 1 TB SSD, and an NVIDIA GeForce RTX 4070ti super GPU. The experimental environment

was set up using Python 3.8, with machine learning frameworks TensorFlow 2.5 and PyTorch 1.9 employed for implementing and training neural networks. We also used CUDA 11.8 with cuDNN support, and fixed random seeds (set to 42) for NumPy, PyTorch, and TensorFlow to ensure reproducibility. These tools and hardware configurations provided an efficient environment to conduct the experiments and analyze the results.

We utilized the LEAF framework [33], which is a benchmarking framework for learning in federated settings. LEAF provides tools and datasets for various applications, including federated learning, multi-task learning, meta-learning. This framework allowed us to simulate realistic federated environments and evaluate the performance of FedCW and baseline algorithms under different data distributions and client behaviors.

We evaluated FedCW on four image classification tasks using the MNIST, CIFAR-10, ImageNet-100, and CelebA datasets. To simulate different degrees of Non-IID across clients, we used a Dirichlet distribution to partition the data. Specifically, the training data for each dataset was divided among clients by sampling from a Dirichlet distribution with varying concentration parameters (α). For reproducibility, we set $\alpha = 0.1, 0.5, 1.0, 10$ in experiments to represent strong non-IID, moderate non-IID, near-IID, and IID-like distributions, respectively. Lower values of α correspond to higher degrees of Non-IID, meaning that clients receive data concentrated on fewer classes. Conversely, higher values of α lead to a more IID-like distribution, where each client receives a more balanced share of the data across all classes. This method allows us to systematically control the degree of data heterogeneity among clients and assess the performance of FedCW under different Non-IID settings.

In our experiments, we focused on full client participation for each communication round. We compared FedCW with several popular baselines, including FedAvg, FedProx, SCAFFOLD, FedOpt, FedNova, and FedMA, across all datasets. Each algorithm was implemented using the same neural network architectures to ensure fair and consistent comparisons. The hyperparameters for each method, such as learning rate, batch size, and number of communication rounds, were tuned empirically based on prior experience. For FedCW specifically, we report the exact values used for λ and β in the experiment tables to ensure reproducibility.

6.2 Experiment Results

6.2.1 Accuracy and Communication Efficiency Comparison

Table 1 shows the performance of FedCW across all four datasets (MNIST, CIFAR-10, ImageNet-100, and CelebA) in terms of communication overhead, computation time, FLOPs, and accuracy. On the MNIST dataset, FedCW achieves a communication overhead of 198.6 MB, a 35% reduction compared to FedAvg's 305.3 MB. Similar reductions are seen on CIFAR-10 (401.8 MB for FedCW vs. 603.2 MB for FedAvg, a 33% decrease). This substantial reduction is due to FedCW's selective client participation strategy based on Euclidean distance, which prioritizes clients contributing the most valuable updates, thus minimizing unnecessary data exchange.

In terms of computation time, FedCW completes training faster; on MNIST, it takes 4173.25 s, a 17% reduction from FedAvg's 5025.6 s. This speed-up comes from focusing only on clients whose updates are most informative, enhancing convergence and reducing the need for prolonged training. Furthermore, FedCW demonstrates computational efficiency with a 9.4% reduction in FLOPs on MNIST compared to FedAvg, owing to its ability to learn quickly from divergent updates and avoid redundant computations.

Table 1: Comparison of federated learning algorithms across different datasets

MNIST	**FedCW**	**FedAvg**	**FedProx**	**SCAFFOLD**	**FedOpt**	**FedNova**	**FedMA**
Testing accuracy (%)	92.27	90.43	91.15	91.52	91.07	91.84	91.93
Communication Overhead (MB)	198.6	305.3	278.7	291.4	274.9	268.2	259.7
Time (s)	4173.25	5025.6	4598.2	4710.7	4643.3	4539.9	4490.1
FLOPs	480,523.0	530,342	504,763	510,124	507,842	500,198	494,972
CIFAR-10	FedCW	FedAvg	FedProx	SCAFFOLD	FedOpt	FedNova	FedMA
Testing accuracy (%)	88.50	84.97	86.13	86.48	86.08	87.75	88.01
Communication Overhead (MB)	401.8	603.2	558.5	582.9	549.3	538.7	521.4
Time (s)	8896.5	9523.1	9187.4	9325.8	9248.9	9167.6	8995.3
FLOPs	960,124.3	1,030,453	979,856	985,302	969,482	959,721	950,631
ImageNet-100	FedCW	FedAvg	FedProx	SCAFFOLD	FedOpt	FedNova	FedMA
Testing Accuracy (%)	75.20	69.87	71.48	72.05	71.76	73.48	74.03
Communication Overhead (MB)	903.4	1205.7	1123.6	1152.4	1101.3	1084.5	1038.9
Time (s)	15,012.8	16,034.2	15,196.7	15,387.5	15,310.4	15,128.9	15,047.3
FLOPs	1,500,254.7	1,601,023	1,550,248	1,561,037	1,545,231	1,520,128	1,510,987
CelebA	FedCW	FedAvg	FedProx	SCAFFOLD	FedOpt	FedNova	FedMA
Testing accuracy (%)	85.70	81.03	82.47	83.01	82.78	84.52	84.96
Communication Overhead (MB)	702.5	904.8	839.2	868.7	823.1	809.6	779.3
Time (s)	12,013.7	13,025.4	12,237.8	12,302.1	12,262.5	12,119.8	12,047.2
FLOPs	1,100,482.2	1,201,235	1,150,843	1,160,452	1,135,246	1,120,329	1,110,847

The higher accuracy achieved by FedCW—92.27% on MNIST, which is 1.84% higher than FedAvg—is attributed to its dynamic client selection process. By selecting clients that provide the most significant contributions to the model, FedCW maintains robustness, especially in non-IID data scenarios. In contrast, other algorithms like FedAvg involve random client selection, which may include clients with less impactful updates, resulting in slower convergence and lower accuracy. Thus, FedCW's strategic client selection and efficient aggregation contribute to its superior performance.

6.2.2 Ablation Study on FedCW Components

The plot in Fig. 1 illustrates the results of an ablation study comparing different versions of the FedCW algorithm over 100 communication rounds. It includes the complete FedCW algorithm and versions where client selection or weighted averaging is removed, as well as a baseline random client selection. From the plot, the **complete FedCW algorithm** achieves the highest accuracy, starting at around 67% and steadily rising to the capped maximum of 95%. In comparison, the version of FedCW without weighted averaging starts lower, around 65%, and improves more slowly, ultimately plateauing below 90%. Similarly, the version without client selection performs better than random selection but still underperforms compared to the complete FedCW algorithm, showing the importance of both components. Random client selection achieves the lowest accuracy, indicating that without any strategic selection mechanism, the model struggles to achieve high accuracy, with slower improvements over the rounds.

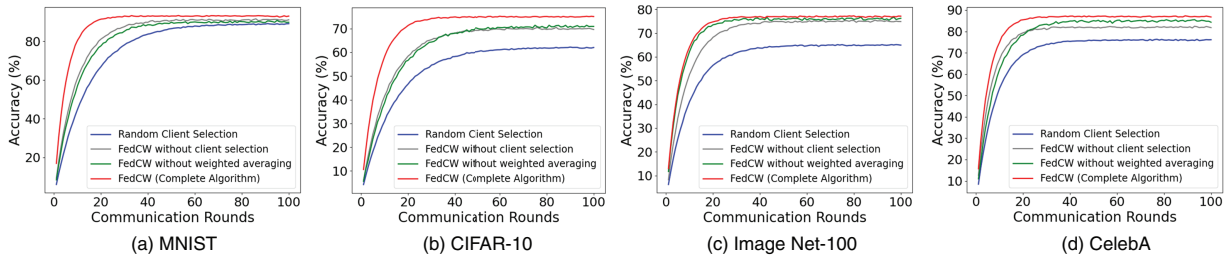

Figure 1: Impact of different components on FedCW performance. Different lines shows the performance of FedCW when key components such as client selection and weighted averaging are removed

The superior performance of FedCW can be attributed to two key components: **client selection** and **weighted averaging**. The **client selection strategy**, which prioritizes clients with the most informative updates, ensures that each round of communication focuses on maximizing the global model's progress. By selecting clients with more divergent local data, the algorithm captures a wider range of variations in the dataset, leading to faster convergence. The **weighted averaging mechanism** dynamically adjusts the importance of client contributions based on their relevance, ensuring that significant updates have a larger impact on the global model. This helps maintain balance and avoids overfitting to any particular subset of clients, which is crucial in non-IID federated learning environments where client data distributions are heterogeneous. In contrast, removing these components diminishes the model's ability to effectively utilize client data, resulting in slower convergence and lower final accuracy, as shown by the other lines in the plot. Overall, FedCW's combination of these strategies allows it to leverage diverse and non-IID data efficiently, leading to faster convergence and higher overall performance compared to ablated versions of the algorithm.

6.2.3 Impact of Non-IID Data Distribution on Algorithm Performance

The experimental results illustrated in Fig. 2 demonstrate the robustness and adaptability of the FedCW algorithm across multiple datasets. The goal of this experiment was to assess the performance of FedCW under different levels of Non-IID data distributions. As observed, FedCW consistently outperforms the baseline algorithms such as FedAvg, FedProx, SCAFFOLD, FedOpt, FedNova, and FedMA across all degrees of Non-IID. For example, in the MNIST dataset, FedCW reaches an accuracy of 97% under IID conditions, while FedAvg only achieves 91.5%. A similar trend can be observed across other datasets like CIFAR-10, where FedCW outperforms FedAvg significantly, achieving 75% accuracy compared to 60% for FedAvg under IID conditions. These results also highlight the intrinsic relationship between non-IID distributions, convergence, and communication overhead. As the degree of non-IID increases (lower α), all algorithms require more communication rounds to reach the same accuracy, reflecting slower convergence. However, FedCW mitigates this cost by selecting only the most informative clients and adaptively weighting their updates, which reduces redundant transmissions and accelerates progress. In practice, this means that FedCW can achieve a target accuracy with fewer communication rounds than baselines under strong non-IID, thereby lowering the overall communication overhead while maintaining robust convergence behavior.

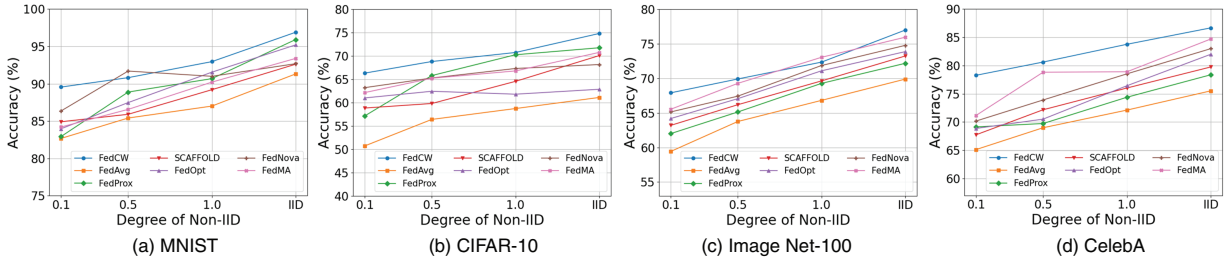

Figure 2: Accuracy of federated learning algorithms under varying degrees of Non-IID Data. The degrees of Non-IID are controlled by four different values of the dirichlet distribution's concentration parameter (α). In this experiment, α values of 0.1, 0.5, 1.0, and 10 were used to simulate increasing levels of data heterogeneity among clients. A lower α (e.g., 0.1) represents a highly Non-IID distribution, while a higher α (e.g., 10) leads to a more IID-like distribution

6.2.4 Convergence Speed Comparison across Algorithms

In this experiment, we compare the convergence speed of several federated learning algorithms, including FedCW, FedAvg, FedProx, SCAFFOLD, FedOpt, FedNova, and FedMA, by tracking their training loss over 100 communication rounds on the MNIST dataset. As illustrated in Fig. 3, FedCW demonstrates the fastest convergence, rapidly reducing its training loss to around 0.5 within the first 10 rounds and maintaining a stable performance throughout the remaining rounds on the MNIST dataset. This behavior is attributable to its client selection strategy and dynamic weighted averaging, which ensures that the most informative clients are selected, and their updates are efficiently aggregated. FedProx and FedMA follow in terms of convergence speed, with FedProx benefiting from the addition of a proximal term that stabilizes local updates in non-IID environments, leading to faster convergence compared to FedAvg. FedMA, with its layer-wise neuron matching, also performs well by preserving more structural information during aggregation, thereby reducing the loss more effectively than the other methods. SCAFFOLD exhibits a slightly slower convergence than FedCW and FedProx, although it manages to address client drift by utilizing control variates. FedOpt and FedNova show moderate improvements over FedAvg but are slower than FedCW due to their simpler aggregation mechanisms. FedAvg, as the baseline, has the slowest convergence, as it relies on random client selection and equal-weight averaging, which are less effective in non-IID settings. In several other data sets, the experimental results show similar characteristics. The results highlight the importance of sophisticated client selection and weighted aggregation strategies, as seen in FedCW, in achieving faster convergence, especially in heterogeneous federated learning environments.

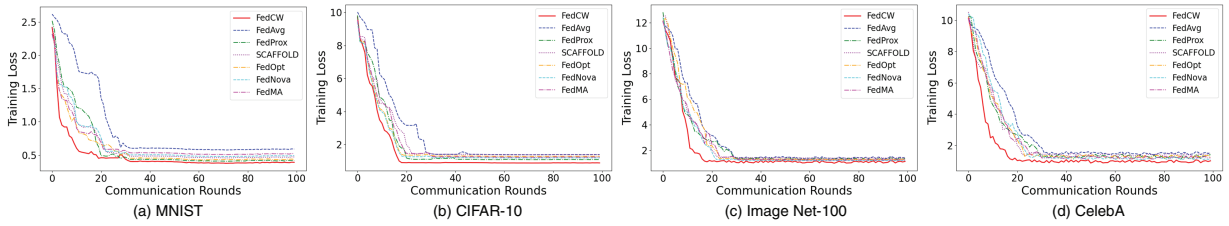

Figure 3: Convergence speed of algorithms (Training Loss vs. Communication Rounds)

6.2.5 Effect of Client Participation on Accuracy

As previously discussed, client selection and weight allocation are fundamentally the same problem. Therefore, the FedCW algorithm, which addresses both issues simultaneously, demonstrates a significant advantage over other algorithms. Fig. 4 clearly reveal this. The results from the experiment testing the

impact of varying client numbers on accuracy reveal that FedCW consistently achieves superior performance compared to other algorithms across different datasets, including MNIST, CelebA, CIFAR-10, and ImageNet-100. As the number of clients increases, FedCW is better able to aggregate diverse and representative data, allowing the model to generalize more effectively and maintain high accuracy. In contrast, algorithms like FedAvg, which use simple averaging across all clients, struggle to maintain high accuracy as the number of clients increases, particularly in non-IID settings. FedProx, SCAFFOLD, FedOpt, FedNova, and FedMA also benefit from their respective aggregation and optimization strategies, but none match FedCW's ability to handle a large number of clients while maintaining robust performance. FedCW's ability to adaptively adjust weight allocation during aggregation ensures that the most valuable updates are emphasized, further enhancing the model's convergence speed and overall accuracy. This makes FedCW particularly advantageous in federated learning environments with a varying number of participating clients.

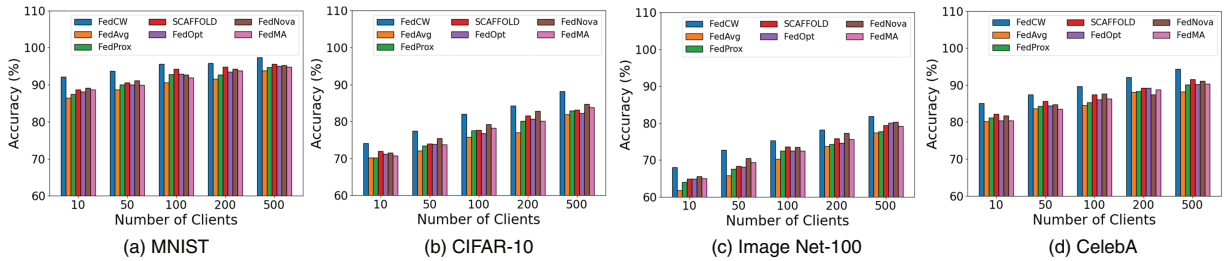

Figure 4: Accuracy vs. number of clients across multiple datasets. Each bar represents the performance of different algorithms, including FedCW, FedAvg, FedProx, SCAFFOLD, FedOpt, FedNova, and FedMA

7 Conclusion

In this paper, we proposed FedCW. It is an algorithm designed to simultaneously address the challenges of client selection and weight allocation in heterogeneous Federated Learning (FL) environments. Our approach leverages digital twins to assist in real-time computation offloading, while selecting clients based on their Euclidean distance from the global model and dynamically adjusting aggregation weights to balance data volume and model divergence. Through extensive experiments, we demonstrated that FedCW significantly improves model accuracy and reduces convergence time compared to existing methods such as FedAvg, FedProx, and SCAFFOLD, particularly in non-IID settings. In the future, further work can focus on enhancing the adaptability of FedCW to even more dynamic edge environments and exploring more advanced techniques for optimizing resource allocation in large-scale FL systems.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Haotian Wu and Jiaming Pei; methodology, Haotian Wu; software, Haotian Wu; validation, Haotian Wu; formal analysis, Haotian Wu; investigation, Haotian Wu; resources, Jinhai Li; data curation, Jinhai Li; writing—original draft preparation, Haotian Wu and Jiaming Pei; writing—review and editing, Haotian Wu and Jiaming Pei; visualization, Haotian Wu; supervision, Jiaming Pei and Jinhai Li; project administration, Jinhai Li; funding acquisition, Jinhai Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. Westminster, UK: PMLR; 2017. p. 1273–82.
2. Tang F, Mao B, Kato N, Gui G. Comprehensive survey on machine learning in vehicular network: technology, applications and challenges. IEEE Commun Surv Tutor. 2021;23(3):2027–57.
3. Posner J, Tseng L, Aloqaily M, Jararweh Y. Federated learning in vehicular networks: opportunities and solutions. IEEE Netw. 2021;35(2):152–9. doi:10.1109/mnet.011.2000430.
4. Pei J, Li W, Mumtaz S. From routine to reflection: pruning neural networks in communication-efficient federated learning. IEEE Trans Artifi Intell. 2024:1–10. doi:10.1109/tai.2024.3462300.
5. Elbir AM, Soner B, Çöleri S, Gündüz D, Bennis M. Federated learning in vehicular networks. In: 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom); 2022 Sep 5–8; Athens, Greece. p. 72–7.
6. Lai C, Lu R, Zheng D, Shen X. Security and privacy challenges in 5G-enabled vehicular networks. IEEE Netw. 2020;34(2):37–45. doi:10.1109/mnet.001.1900220.
7. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Proc Mag. 2020;37(3):50–60. doi:10.1109/msp.2020.2975749.
8. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Found Trends® Mach Learn. 2021;14(1–2):1–210.
9. Pei J. F3: fair federated learning framework with adaptive regularization. Knowl Based Syst. 2025;316(1–2):113392. doi:10.1016/j.knosys.2025.113392.
10. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol. 2019;10(2):1–19. doi:10.1145/3298981.
11. Wang J, Liu Q, Liang H, Joshi G, Poor HV. Tackling the objective inconsistency problem in heterogeneous federated optimization. Adv Neural Inf Proc Syst. 2020;33:7611–23.
12. Pei J, Omar M, Dabel MMA, Mumtaz S, Liu W. Federated few-shot learning with intelligent transportation cross-regional adaptation. IEEE Trans Intell Transp Syst. 2025:1–10. doi:10.1109/tits.2025.3563928.
13. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. IEEE Internet Things J. 2016;3(5):637–46. doi:10.1109/jiot.2016.2579198.
14. Pei J, Li J, Song Z, Dabel MMA, Alenazi MJF, Zhang S, et al. Neuro-VAE-symbolic dynamic traffic management. IEEE Trans Intell Transp Syst. 2025:1–10. doi:10.1109/tits.2025.3571210.
15. Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019—2019 IEEE International Conference on Communications (ICC); 2019 May 20–24; Shanghai, China. p. 1–7.
16. Cho YJ, Wang J, Joshi G. Client selection in federated learning: convergence analysis and power-of-choice selection strategies. arXiv:2010.01243. 2020.
17. Yoshida N, Nishio T, Morikura M, Yamamoto K, Yonetani R. Hybrid-FL for wireless networks: cooperative learning mechanism using non-IID data. In: ICC 2020—2020 IEEE International Conference On Communications (ICC); 2020 Jun 7–11; Online. p. 1–7 .
18. Wu W, He L, Lin W, Maple C. FedProf: selective federated learning with representation profiling. arXiv:2102.01733. 2021.
19. Shi F, Hu C, Lin W, Fan L, Huang T, Wu W. VFedCS: optimizing client selection for volatile federated learning. IEEE Internet Things J. 2022;9(24):24995–5010. doi:10.1109/jiot.2022.3195073.
20. Zhang R, Mao J, Wang H, Li B, Cheng X, Yang L. A survey on federated learning in intelligent transportation systems. IEEE Trans Intell Vehicles. 2025;10(5):3043–59. doi:10.1109/tiv.2024.3446319.
21. Maroua D. A state-of-the-art on federated learning for vehicular communications. Veh Commun. 2024;45(3):100709. doi:10.1016/j.vehcom.2023.100709.
22. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. Proc of Mach Learn and Syst. 2020;2:429–50.

23. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. Scaffold: stochastic controlled averaging for federated learning. In: International Conference on Machine Learning. Westminster, UK: PMLR; 2020. p. 5132–43.
24. Reddi S, Charles Z, Zaheer M, Garrett Z, Rush K, Konečnỳ J, et al. Adaptive federated optimization. arXiv:2003.00295. 2020.
25. Wang H, Yurochkin M, Sun Y, Papailiopoulos D, Khazaeni Y. Federated learning with matched averaging. arXiv:2002.06440. 2020.
26. Huang K, Xian R, Xian M, Wang H, Ni L. A comprehensive intrusion detection method for the internet of vehicles based on federated learning architecture. Comput Secur. 2024;147:104067.
27. Althunayyan M, Javed A, Rana O. A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning. Veh Commun. 2024;49(6):100837. doi:10.1016/j.vehcom.2024.100837.
28. Hasan MK, Jahan N, Nazri MZA, Islam S, Khan MA, Alzahrani AI, et al. Federated learning for computational offloading and resource management of vehicular edge computing in 6G-V2X network. IEEE Trans Consum Electron. 2024;70(1):3827–47. doi:10.1109/tce.2024.3357530.
29. Yan J, Chen T, Sun Y, Nan Z, Zhou S, Niu Z. Dynamic scheduling for vehicle-to-vehicle communications enhanced federated learning. IEEE Trans Wirel Commun. 2025. doi:10.1109/twc.2025.3573048.
30. Zhang C, Zhang W, Wu Q, Fan P, Fan Q, Wang J, et al. Distributed deep reinforcement learning based gradient quantization for federated learning enabled vehicle edge computing. IEEE Internet Things J. 2025;12(5):4899–913. doi:10.1109/jiot.2024.3447036.
31. Alqubaysi T, Asmari AFA, Alanazi F, Almutairi A, Armghan A. Federated learning-based predictive traffic management using a contained privacy-preserving scheme for autonomous vehicles. Sensors. 2025;25(4):1116. doi:10.3390/s25041116.
32. Ali W, Din IU, Almogren A, Rodrigues JJ. Federated learning-based privacy-aware location prediction model for internet of vehicular things. IEEE Trans Veh Technol. 2024;74(2):1968–78. doi:10.1109/tvt.2024.3368439.
33. Caldas S, Duddu SMK, Wu P, Li T, Konečnỳ J, McMahan HB, et al. Leaf: a benchmark for federated settings. arXiv:1812.01097. 2018.