future internet

MDPI

Article

FedAWR: Aggregation Optimization in Federated Learning with Adaptive Weights and Learning Rates

Tong Yao [1], Jianqi Li [2,3] and Jianhua Liu [1,*]

[1] School of Transportation and Electrical Engineering, Hunan University of Technology, Zhuzhou 412007, China; tongyao@stu.hut.edu.cn
[2] School of Artificial Intelligence, Hunan Institute of Science and Technology, Yueyang 414006, China; 12025043@hnist.edu.cn
[3] School of Computer and Electrical Engineering, Hunan University of Arts and Science, Changde 415000, China
* Correspondence: jhliu@hut.edu.cn

Abstract

Federated Learning (FL) enables collaborative model training without sharing raw data, offering a promising solution for privacy-sensitive applications. However, in real-world deployments, significant disparities in client computational capabilities lead to imbalanced model updates, resulting in slow convergence and degraded model generalization. To address this challenge, this paper proposes a novel federated aggregation optimization method, FedAWR, which features adaptive adjustment of learning rates and weights. Specifically, during the global aggregation phase, our method dynamically adjusts each client's aggregation weight based on its computational capability and configures an appropriate learning rate to balance training progress. Experiments on multi-classification tasks using the Steel Rail Defect and CIFAR-10 datasets demonstrate that the proposed method exhibits significant advantages over mainstream federated algorithms in both convergence efficiency and model generalization performance, thereby validating its effectiveness and superiority.

Keywords: federated learning; model aggregation; device heterogeneity; adaptive weights

1. Introduction

Federated Learning (FL) represents an innovative distributed machine learning paradigm that facilitates the collaborative training of a global model without the need for clients to share their local raw data. This unique feature endows FL with substantial application potential in privacy-sensitive domains, including finance and healthcare [1–4]. In the contemporary landscape where data has emerged as a core production factor, FL provides a feasible technological pathway to overcome "data silos" and enable cross-organizational collaborative modeling. It holds profound strategic significance for fostering the circulation of data elements, safeguarding data security and privacy, and promoting the broader adoption of artificial intelligence technologies.

Nevertheless, the practical deployment of FL is confronted with the formidable challenge of "System Heterogeneity." A particularly acute aspect of this challenge is the disparity in client computational performance. The hardware configurations, energy supplies, and operational loads of various end devices—such as smartphones, IoT sensors, and edge servers—vary considerably, leading to significant differences in their computational capabilities. This disparity directly triggers a cascade of issues. For example, clients with

Check for updates

Academic Editor: Paolo Bellavista

Received: 12 January 2026
Revised: 10 February 2026
Accepted: 13 February 2026
Published: 18 February 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) license.

inferior computational power necessitate extended local training durations, causing them to lag in each global update round and creating imbalances in the step sizes of model updates. The resulting frequent waits and delays extend the overall training cycle, substantially impeding the convergence rate of the global model. Moreover, clients that are persistently disadvantaged may have their updates repeatedly overlooked, preventing their local data distribution characteristics from being adequately learned. This ultimately leads to the global model becoming "biased" towards high-performance clients, thereby compromising its fairness and generalization capability [5–9]. Consequently, the design of optimization algorithms capable of effectively mitigating computational performance disparities is central to improving the practicality and fairness of FL, representing a critical scientific problem that demands immediate attention.

In response to this, we propose a federated learning model aggregation optimization method with adaptive weights and learning rates (FedAWR). This method adjusts client weight allocation during the global aggregation phase, increasing the weight of clients with weaker computational performance to enhance the contribution of updates from low-performance devices. Simultaneously, it adjusts the learning rate based on client training speeds to balance the training progress and optimization direction across different devices, thereby accelerating model convergence efficiency.

The main contributions of this paper are summarized as follows:

We propose FedAWR, a computation-aware federated aggregation framework that jointly couples adaptive aggregation weighting and adaptive client-specific learning rate scheduling to address system heterogeneity. Unlike existing methods that apply these mechanisms independently, FedAWR explicitly aligns update direction and update magnitude based on clients' effective computation within a fixed time window.

We provide intuitive insights into how the proposed joint design mitigates the bias introduced by heterogeneous local update frequencies, improving both convergence speed and stability under computational heterogeneity.

Extensive experiments on both a real-world industrial dataset and the CIFAR-10 benchmark demonstrate that FedAWR consistently outperforms representative baselines under heterogeneous computational settings.

2. Related Work

Model aggregation strategies designed to address computational performance disparities can be primarily categorized into three domains. The first category involves communication-based mechanisms, such as asynchronous learning [10–13]. This approach accommodates resource-constrained devices, thereby enhancing the overall efficiency and scalability of federated learning. However, the principle of immediate aggregation can lead to version inconsistencies between updates from slower nodes and the global model, which paradoxically degrades training efficiency and performance [14–16]. The second category encompasses client selection-based aggregation strategies [17–20], which accelerate training by selecting high-performance devices. While potentially increasing training speed, this approach excludes slower clients from participation, risking the omission of data and compromising the model's training efficacy [21–23]. The final category involves regularization-based model manipulation, which reduces local model divergence through various regularization operations or normalized constraints on parameter update directions, thereby improving the generalization capability of the global model. Specifically, FedProx [24] introduces a regularization term to FedAvg [25] to mitigate divergence, yet it remains insufficient in suppressing the model shift induced by computational heterogeneity. FedNova [26] normalizes local client updates before performing weighted aggregation, reducing model bias caused by unbalanced local iterations and partially alleviating the slow

convergence associated with computational disparities. Nevertheless, the aforementioned methods fail to address the problem of computational performance disparity at its root. Nevertheless, the aforementioned methods fail to address the problem of computational performance disparity at its root. Whether by passively adapting to the environment or imposing one-sided constraints, these approaches overlook the synergistic optimization of the two core parameters: the learning rate and aggregation weights.

3. Problem Formulation

The core mechanism of federated learning is "data stays put, models move." Specifically, a central server distributes a global model to clients for local training and processing. Clients upload only model updates rather than raw data, and the server aggregates these updates to optimize the international model. This process iterates until the model converges. Taking the FedAvg algorithm as an example, the optimization objective for each client is to minimize the average loss across all samples, expressed as (1):

$$F(x) = \sum_{i=1}^{N} \frac{n_i f_i(x)}{n} \tag{1}$$

$$f_i(x) = \frac{1}{n_i} \sum_{k \in D_i} l(x_i, y_i; w_i) \tag{2}$$

$F(x)$ is defined as the weighted average loss across all clients under the current parameters, thereby serving as the objective function. $f_i(x)$ is represented by the average loss of the local data from the i-th client participating in federated learning. w_i is the local model parameter of the i-th client, while x_i and y_i represent the input data and its corresponding label, respectively. $l(\cdot)$ is the loss function employed by each client.

Despite the provision of a theoretical framework for disrupting data silos through federated learning, the inherent disparity in computational performance has emerged as a pivotal bottleneck, hindering both performance and efficiency during the transition from theoretical frameworks to practical applications, as illustrated in Figure 1.

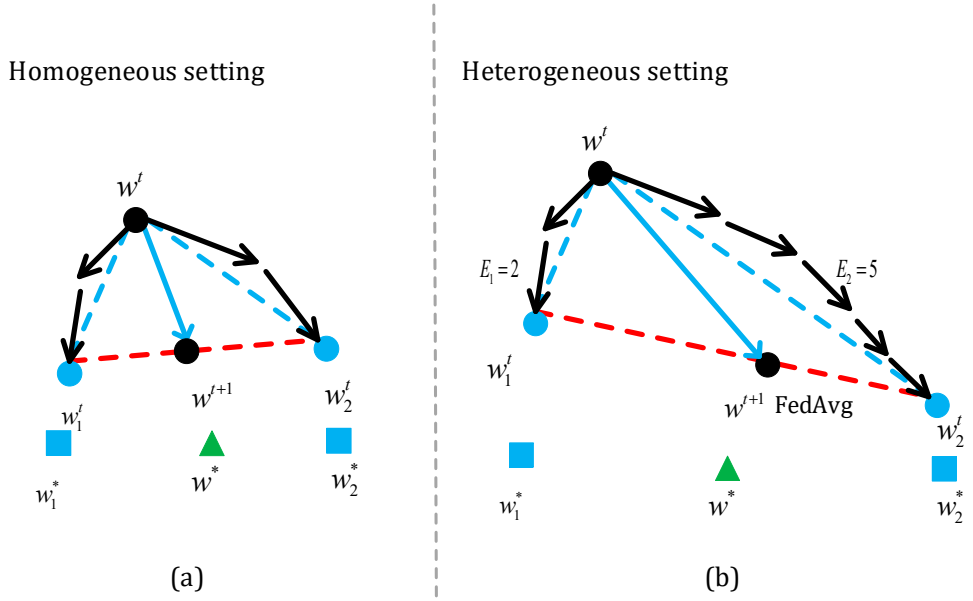

Figure 1. Parameter updates under different system settings: (**a**) Homogeneous setting, where clients perform an equal number of local updates; (**b**) Heterogeneous setting, where clients execute different numbers of local update steps. The black arrows indicate local update trajectories, the blue lines represent client updates, and the red dashed line illustrates the aggregation direction under FedAvg.

As illustrated in Figure 1a, during the t-th iteration of model updates, w_1^t and w_2^t denote the model parameters that are updated by clients 1 and 2, respectively, utilizing their local data. In contrast, w^{t+1} signifies the global model parameters that have been aggregated from both clients. It is noteworthy that w_1^* and w_2^* correspond to the optimal values for the local and global models, respectively. Assuming identical computational capabilities across devices, consistent update steps are maintained for each local model. Consequently, it is possible to progressively update the aggregated global model parameters in the direction that optimizes the model. w_2^* and w^* represent the optimal values for the respective device and global models. Assuming identical computational capabilities across devices, consistent local model update steps ensure the aggregated global model parameters converge stepwise toward the global optimum along the optimal direction.

However, in practical application environments, computational resources and update speeds often vary across clients. This computational performance imbalance leads to inconsistent model update steps and synchronization frequencies, resulting in shifts in the global model and a corresponding reduction in convergence efficiency. Figure 1b provides a visual representation of the impact of computational heterogeneity on global model updates. In the context of federated learning, the client on the left is constrained to completing a maximum of two training rounds within the allocated time frame. Conversely, the high-performance client on the right is capable of achieving five rounds. In scenarios where a server aggregates models from disparate clients by implementing the FedAvg parameter averaging strategy, the resulting global model often fails to converge to the global optimum. As high-performance clients execute additional local training rounds, their update directions exert a more substantial influence on the global model. The consequence of this phenomenon is the deviation of the aggregated result, w^{t+1}, from the global optimum, w^*, and its subsequent drift toward the local minimum, w_2^*, of the high-performance client. The weight allocation mechanism, which is based on data volume, is weakened by computational disparities among clients. The utilization of high-computing-power clients results in the amplification of their data contribution through additional local iterations, thereby causing the global model to exhibit a skew toward them. Furthermore, applying a uniform learning rate to all clients results in over-updating for high-computing-power clients and under-updating for low-computing-power clients, thereby further exacerbating the model's convergence instability.

4. Methods

To address issues of model bias and convergence instability caused by computational performance disparities among terminal devices, a solution is proposed. This solution is referred to as Federated Learning Model Aggregation Optimization with Adaptive Weight and Learning Rate Adjustment (FedAWR). The fundamental principle of FedAWR is to augment the contribution weight of pivotal clients during the aggregation phase and to adjust the learning rate adaptively based on computational capabilities. This dynamic optimization of the model update direction has been shown to enhance global convergence efficiency. By taking into account each client's computational capability and training contribution, this method dynamically allocates global aggregation weights and local learning rates. This approach is designed to balance the influence of high- and low-computational-power devices during model updates. The efficacy of this approach is demonstrated by its improvement of the efficiency, stability, and generalization performance of global model training, as illustrated in Figure 2.

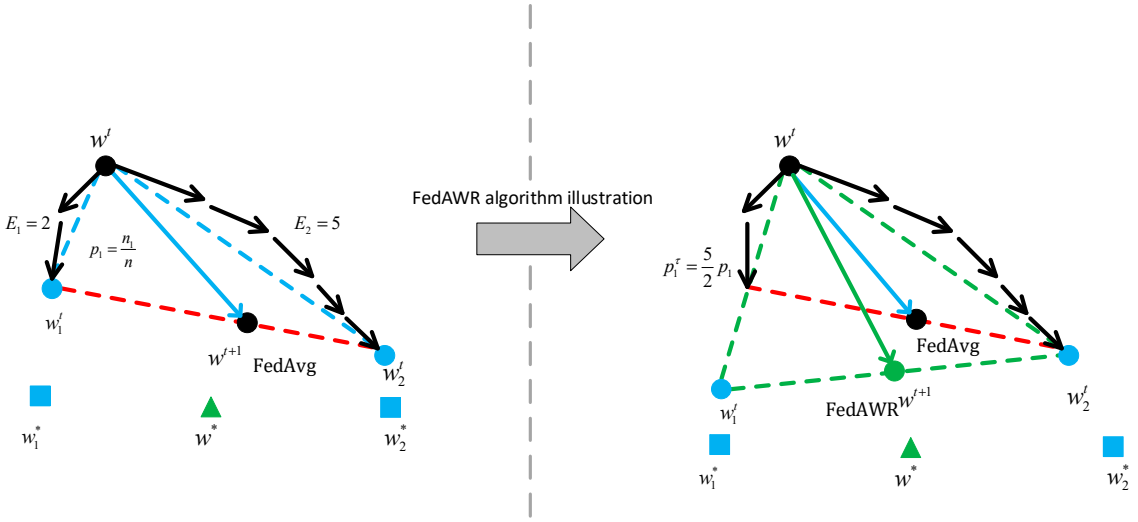

Figure 2. Illustration of parameter update trajectories under FedAvg and FedAWR. Solid arrows denote local model updates, dashed arrows represent adjusted update paths under heterogeneous local computation, and the red dashed line indicates the global aggregation direction. FedAWR adaptively adjusts aggregation weights and learning rates to better align the global update under system heterogeneity.

FedNova addresses objective inconsistency caused by heterogeneous numbers of local update steps by explicitly normalizing client updates so that the aggregated update corresponds to a well-defined global objective. In contrast, FedAWR does not reformulate or normalize the optimization objective itself. Instead, it dynamically adjusts aggregation weights and client-specific learning rates based on effective local computation within a fixed time window.

From an optimization perspective, FedNova enforces objective consistency through update normalization, whereas FedAWR focuses on aligning the optimization dynamics by jointly compensating update direction and magnitude under system heterogeneity. As a result, FedAWR remains compatible with the original federated optimization objective while introducing computation-aware adaptation.

Under these assumptions, FedAWR preserves the convergence properties of FedAvg by performing weighted aggregation of client updates, while introducing computation-aware normalization to mitigate bias caused by heterogeneous local update frequencies. Compared to FedAvg, which implicitly assumes homogeneous local computation, FedAWR explicitly accounts for variability in effective local updates. Compared to FedNova, which normalizes updates based on the number of local steps to eliminate objective inconsistency, FedAWR additionally adapts client-specific learning rates to align update magnitudes under system heterogeneity.

To mitigate model bias caused by computational performance disparities, a weight adjustment strategy based on computational capabilities is introduced during the aggregation phase. It has been demonstrated that, by increasing the aggregation weight of low-computing-power clients, the updated global model parameters w^{t+1} can be shown to converge toward the global optimum w^*. This results in the avoidance of the local optimum w_2^*, which is biased toward high-computing-power clients. The weight adjustment formula for client i is given by Equation (3).

$$p_i^\tau = \gamma_i p_i \tag{3}$$

The weight amplification factor for the global model update aggregation of the i-th client is represented by $\gamma_i = E/E_i$. The symbol E denotes the target number of local training

rounds for federated training clients, E_i denotes the actual number of update rounds for the i-th client, and $p_i = n_i/n$ denotes the aggregation coefficient for data proportion.

As demonstrated in Figure 2, by increasing the aggregation weight of the low-computing-power clients on the left—whose adjusted weight is 5/2 times the original data-based allocation p_1—we prevent their marginalization during long-term training. This adjustment rectifies the update direction of the global model, thereby enhancing its generalization performance.

In the context of federated learning, the heterogeneity of clients is characterised by two distinct manifestations. Firstly, this heterogeneity is evident in the scale of local datasets. Secondly, it is apparent in the computational speeds of clients, resulting in substantial variations in the number of local updates. Specifically, if each client is required to perform E training rounds using mini-batch size B for local stochastic gradient descent (SGD), the number of local SGD iterations, τ_i, is determined by its data volume, n_i, as shown in Equation (4). Furthermore, disparities in computational speed exacerbate this heterogeneity: within a fixed communication round time window, computationally powerful clients can execute far more local updates than their slower counterparts. In this study, the client's computational capability is quantified by τ, the number of local training steps completed within a fixed time window, which directly reflects real-time computational efficiency.

$$\tau_i = \frac{E_i n_i}{B} \tag{4}$$

To address issues arising from suboptimal learning rate configurations on clients with varying computational capabilities and the resulting decline in convergence efficiency, the learning rate is dynamically adjusted based on each client's computational power. Clients with strong computational capabilities employ a smaller learning rate to prevent overfitting, while clients with weaker capabilities use a larger learning rate to accelerate convergence. The learning rate adjustment formula for client i is shown in Equation (5).

$$\eta_i = \frac{\sum_{i=1}^{N} \tau_i}{N\tau_i}\eta \tag{5}$$

Within the formula, N denotes the number of clients participating in federated learning, and η is the initial uniform learning rate. The server assesses the client's computational capability by monitoring the number of local gradient descent steps, τ, that the client completes. The server then adjusts the learning rate accordingly to enhance overall convergence efficiency.

The learning rate scheduling in Equation (5) is motivated by the observation that clients with fewer local update steps contribute smaller update magnitudes under a fixed learning rate, which may bias the global optimization process under computational heterogeneity. By inversely adjusting the learning rate with respect to the effective number of local updates, Equation (5) aims to align the expected update magnitude across clients within each communication round. Moreover, the learning rate is bounded to ensure stable optimization dynamics, which is consistent with standard convergence conditions in stochastic gradient descent and federated optimization. While a formal convergence proof is beyond the scope of this work, the proposed formulation is grounded in established optimization intuition and validated empirically in our experiments.

In practice, τ_i may be subject to minor inaccuracies due to network delays or system noise. Since τ_i is used in a relative and normalized manner to adjust both aggregation weights and learning rates, and the adaptive learning rate is bounded within a predefined range, such moderate variations do not lead to unstable updates or convergence issues in FedAWR.

While FedAWR is not explicitly designed as a defense mechanism against adversarial behaviors such as label-flipping attacks, its design incorporates variance-control principles that are commonly adopted in robust federated learning. Prior studies have shown that limiting the influence of abnormal or noisy client updates through aggregation control or update bounding is effective in mitigating performance degradation under label-flipping and poisoning attacks [27,28]. By bounding adaptive learning rates and normalizing aggregation weights, FedAWR similarly restricts the impact of under-trained or noisy client updates, thereby improving training stability under heterogeneous system conditions.

To avoid potential training instability caused by aggressive updates, the adaptive learning rate is explicitly bounded within a predefined range. This design ensures that even for clients with limited computational capability, the increased learning rate does not lead to excessively large update steps or divergence during training.

We discuss the convergence behavior of FedAWR under standard assumptions commonly adopted in federated learning. Specifically, we assume that each client's local objective function is smooth and possibly non-convex, and that stochastic gradients have bounded variance. We further assume bounded learning rates and partial client participation per communication round, consistent with the settings of FedAvg and its variants.

Adaptive aggregation weighting and adaptive learning rate adjustment play complementary roles. The aggregation weight determines the relative contribution of each client update to the global model, thereby influencing the update direction, while the learning rate controls the magnitude of each local update. For low-capability clients that perform fewer local update steps, increasing the aggregation weight alone may not sufficiently compensate for the reduced update magnitude, and increasing the learning rate alone may still result in biased aggregation dominated by high-capability clients. By jointly adjusting both factors, FedAWR aligns both the direction and magnitude of client updates, mitigating bias caused by heterogeneous effective local computation.

This aggregation strategy effectively mitigates model update direction bias caused by computational performance disparities among clients by adaptively compensating for weaker clients. Concurrently, the learning rate of each client is dynamically configured based on its computational capability, thereby accelerating the convergence of the global model.

5. Experiments and Discussion

To validate the effectiveness of the proposed federated learning framework in industrial settings, we conducted validation analyses using both a rail defect dataset and the publicly available CIFAR-10 dataset.

5.1. Application and Experimental Analysis in Steel Rail Defect Recognition

5.1.1. Experimental Dataset

The data for the study were collected from vibration signals in various areas of a railway maintenance depot in China. Each area in the data represents a separate client. The data collection process involves devices that measure movement and are attached to rail inspection vehicles, aiming to capture the actual movement features of rails during operation. The raw data that measure movement show particular features, including the distribution of energy that differs across the signal, interference from noise, and changes over time. These features require a procedure that consistently prepares the data for analysis. The procedure includes several main steps. The study first applies random reordering to the data to provide randomness in the samples. The study then employs a method that adjusts peak values to ensure consistent data scales across all samples. The procedure selects the initial 2500 time steps to match the requirements that the model uses

for input. The study divides the data into training and test sets using a seven-to-three ratio and provides labels that correspond to these sets. This approach addresses the problems in the raw data, improves the data quality, and establishes a foundation that supports the model training that follows.

5.1.2. Steel Dataset Experimental Setup

The experimental design of this study aims to fairly evaluate the impact of computational performance differences on federated learning. To achieve this objective, it was necessary to establish identical experimental conditions for all comparison methods. To this end, the LeNet neural network architecture was adopted uniformly across all clients, and the data were partitioned following a non-IID setting with heterogeneous label distributions. This standardised configuration serves to ensure that performance variations are attributable solely to differences in computational capabilities, as opposed to model or data factors.

To validate the impact of computational performance variations on federated learning outcomes, experiments were conducted comparing FedAvg, FedProx, FedNova, and the proposed FedAWR algorithm under a synchronous communication mode. FedProx has been developed to mitigate system bias by introducing a proximal regularisation term. FedNova has been designed to correct global update directions through gradient normalisation. FedAWR has been developed to simultaneously address computational performance disparities and parameter configuration heterogeneity by dynamically adapting aggregation weights and learning rates.

All experiments were conducted on a single machine equipped with an NVIDIA GeForce RTX 4060 GPU. To simulate real-world client computational performance variations, VMware Workstation virtual machines were used to allocate different computational resources to each client, thereby replicating computational performance differences on uniform hardware. Experiments were conducted under identical system constraints. The implementation of all algorithms was undertaken utilising the PyTorch 2.5.1 framework, with the configuration of the core hyperparameters being as follows: the federated system comprised 10 clients, undergoing 600 communication rounds with 10 local training rounds per round. The local optimiser utilised a stochastic gradient descent method with a learning rate of 0.001 and a batch size of 24. The subsequent results analysis will provide a detailed account of the specific distribution of client computational performance variations.

5.1.3. Analysis of Experimental Results

To quantify the computational performance differences among clients, it is necessary to send the relevant data to the server. This data comprises the client's training step count, τ_i, and training round count, E_i, which are sent to the server upon completion of local training within the specified timeframe. The mean update step count, τ, was calculated for each training round across all clients. The mean training step counts for each client are illustrated in Figure 3.

A substantial degree of heterogeneity is observed in the mean number of update steps across diverse client populations. High-performance devices, namely Clients 7, 8, and 9, exhibited average update steps of 42, 49, and 46, respectively. Conversely, low-performance devices, such as Clients 4 and 6, demonstrated average update steps of merely 2 and 3, respectively. This substantial disparity in step counts vividly illustrates the pervasive issue of uneven computational capabilities among clients in federated learning environments, providing an experimental foundation to validate the effectiveness of our proposed method in heterogeneous settings.

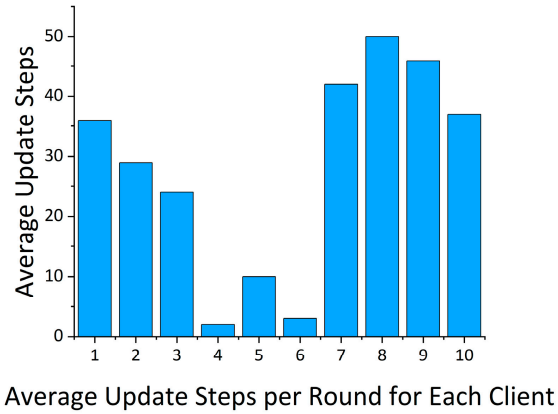

Figure 3. Average Training Steps per Client.

In the context of multi-classification tasks, a multifaceted evaluation approach is employed, encompassing both accuracy metrics and loss values. The specific expression for accuracy is given by Equation (6) below.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

Within the formula, TP, TN, FP, and FN denote True Positive, True Negative, False Positive, and False Negative, respectively.

This section employs a rigorous evaluation framework to assess the model's performance. The evaluation metrics encompass accuracy, loss value, and the number of communication rounds required to achieve the target accuracy. The accuracy curves for various federated learning algorithms are depicted in Figure 4.

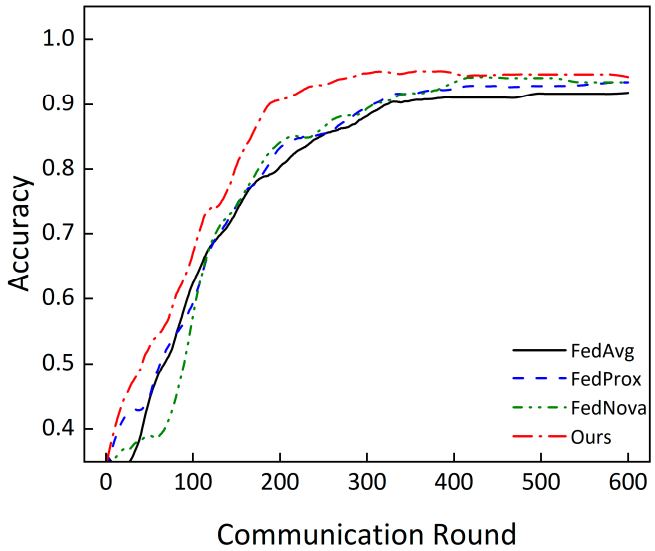

Figure 4. Test Accuracy Convergence Curves of Different Federated Learning Methods on the Rail Dataset.

As the number of communication rounds increases, all algorithms exhibit an upward trend in accuracy. However, substantial disparities emerge in both convergence speed and ultimate performance. FedAWR demonstrates a marked improvement in accuracy during the early stages, attaining an accuracy of 0.90 by approximately 188 rounds. It subsequently maintains this level of accuracy while stabilizing above 0.93. In contrast, FedAvg exhibits the slowest convergence, gradually approaching 0.90 after 300 rounds

and ultimately stabilizing at that level. FedProx and FedNova demonstrate intermediate performance, attaining an approximate value of 0.90 after approximately 300 iterations and subsequently stabilizing.

To evaluate the generalization performance of the algorithms on multi-classification tasks, we employ the cross-entropy as the loss function. The specific expression of the loss function is given in Equation (7) as follows:

$$L_{CS} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{i,j}\log \hat{y}_{i,j} \tag{7}$$

The test loss convergence curves for different federated learning methods on the rail dataset are shown in Figure 5.

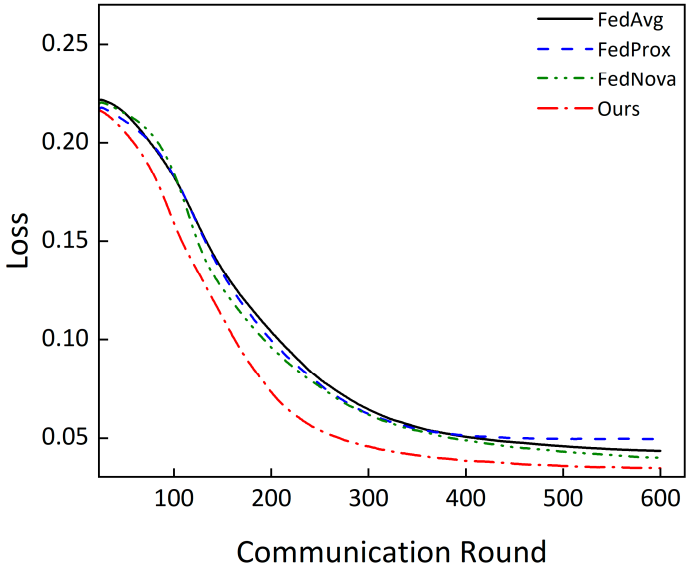

Figure 5. Test Loss Convergence Curves of Different Federated Learning Methods on the Rail Dataset.

It has been observed that the four algorithms demonstrate disparate patterns in the rate of loss reduction across communication rounds. Specifically, FedAvg demonstrates a reduced rate of convergence, with a more gradual decrease in loss values, requiring a greater number of communication rounds to approach convergence. In contrast, FedProx and FedNova exhibit a marginally enhanced rate of loss reduction, although it remains below that of FedAWR. The proposed algorithm has been shown to achieve a significant decrease in loss within a reduced number of communication rounds, thereby demonstrating a noticeably faster rate of loss reduction. The experimental results for final accuracy, loss value, and the number of rounds required to achieve target accuracy are shown in Table 1.

Table 1. Final Accuracy, Loss, and Rounds to Target Accuracy for Each Method on the Rail Dataset.

Method	Target Accuracy $\geq$ 0.90	Accuracy	Loss
FedAvg	323	0.916	0.043
FedProx	308	0.928	0.049
FedNova	307	0.934	0.040
FedAWR	**188**	**0.951**	**0.035**

A comparison of the FedAWR method with baseline methods reveals its superior convergence performance. Specifically, FedAWR achieves a convergence speed that is 1.72 times faster than that of FedAvg, while also demonstrating superior performance

compared to FedProx and FedNova. Moreover, the final model achieves an accuracy of 95.1%, surpassing the other three approaches.

The superiority of FedAWR is primarily attributable to its adaptive mechanism. FedAWR employs a dynamic adjustment of learning rates for individual clients and aggregation weights for the server to direct the global model towards more optimal updates, thereby enhancing the efficiency of the training process. This advantage is particularly evident on the rail damage dataset. Due to significant computational performance disparities among clients, FedAWR achieves faster convergence and lower loss values by linking aggregation weights and learning rates to device computational capabilities.

5.2. Application and Experimental Analysis on CIFAR-10

5.2.1. CIFAR-10 Experimental Setup

To evaluate the generalization capability and universality of the proposed method, experiments were conducted on the publicly available CIFAR-10 image classification dataset. In the comparative experiments, four algorithms were selected for performance assessment: FedAvg, FedProx, FedNova, and the proposed FedAWR. All methods employed the same standard Convolutional Neural Network (CNN) architecture, utilized identical data partitioning strategies, and operated under similar system constraints to ensure a fair comparison. The specific experimental hyperparameter settings are as follows: The experiment was conducted with a total of ten clients, encompassing 300 communication rounds and 10 local training epochs. The Stochastic Gradient Descent (SGD) algorithm was utilized as the local optimizer, with a learning rate set to 0.001 and a batch size of 24.

5.2.2. Experimental Results and Analysis

The performance metrics for the generalization experiments in this section are consistent with those in the previous section. Figure 6 illustrates the convergence accuracy test results for the various federated learning methods on the CIFAR-10 dataset.

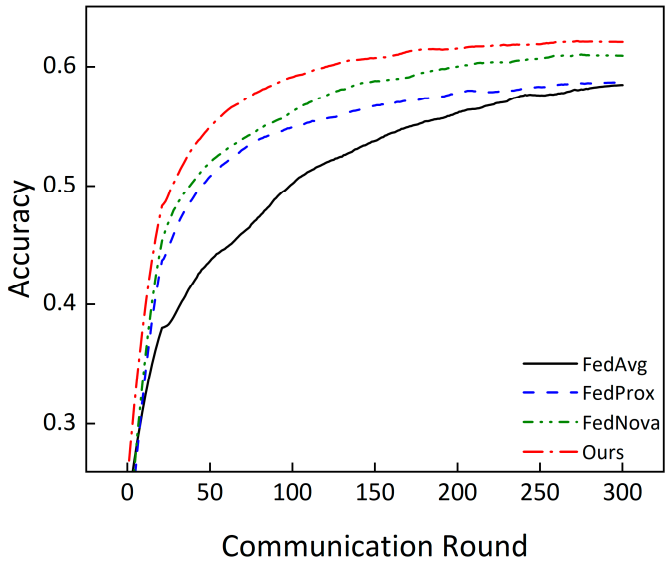

Figure 6. Test Accuracy Convergence Curves of Different Federated Learning Methods on the CIFAR-10 Dataset.

The experimental results demonstrate that the proposed FedAWR algorithm significantly outperforms all baseline methods in terms of generalisation capability. Specifically, while the accuracy of all algorithms improves with increasing training iterations, FedAWR exhibits faster convergence and achieves a higher final accuracy. In the initial 100 training iterations, FedAWR's accuracy rapidly increased, surpassing 0.58 and showing a substantial

performance improvement over FedAvg and FedProx. By the end of the training process, FedAWR's final performance surpassed that of FedNova, attaining the global optimum.

The test loss convergence curves for the different federated learning methods on the CIFAR-10 dataset are presented in Figure 7.

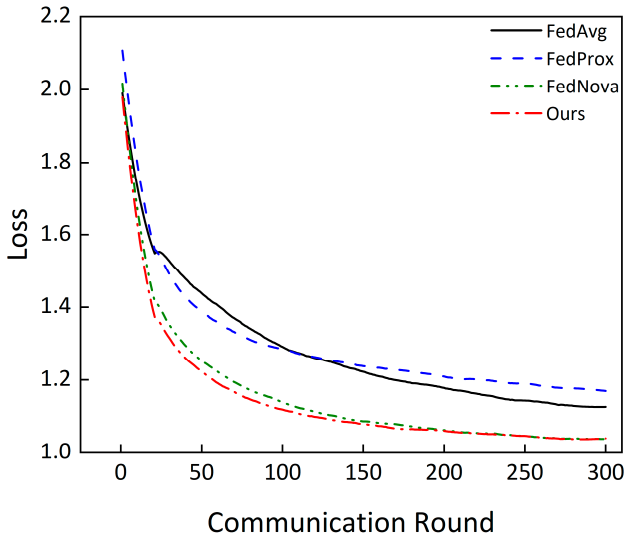

Figure 7. Test Loss Convergence Curves of Different Federated Learning Methods on the CIFAR-10 Dataset.

Regarding the loss reduction rate, FedAWR exhibited a significant advantage during the initial 150 communication rounds, with its loss values decreasing markedly faster than those of FedAvg and FedProx. In the subsequent convergence phase, FedAWR's final loss values reached levels comparable to those of FedNova. The results of the generalisation experiment are presented in Table 2

Table 2. Final Accuracy, Loss, and Rounds to Target Accuracy for Each Method on the CIFAR-10 Dataset.

Method	Target Accuracy $\geq$ 0.56	Accuracy	Loss
FedAvg	271	0.564	1.134
FedProx	234	0.566	1.166
FedNova	127	0.609	1.038
FedAWR	**80**	**0.621**	**1.037**

A thorough analysis of the accuracy and loss values reveals that the FedAWR method surpasses competing algorithms in terms of both convergence speed and loss minimisation. Notably, FedAWR achieves the target accuracy after only 80 communication rounds, demonstrating a faster rate of improvement compared to other methods. Ultimately, FedAWR attains a higher final accuracy than the other three approaches. Furthermore, generalisation experiments confirm that this method achieves convergence more efficiently. In addition, the proposed approach demonstrates superiority in loss reduction, optimising the loss within a reduced timeframe, which further validates its enhanced stability during training.

Under the synchronous federated learning setting considered in this work, all methods exchange model parameters between the server and clients at each communication round. Therefore, the communication cost per round is largely identical across different methods.

In addition to model parameters, FedAWR requires clients to transmit lightweight computation-related information, such as the number of local update steps τ_i. However, this auxiliary information consists of only a small number of scalar values and introduces

negligible communication overhead compared to the transmission of full model parameters. As a result, the total communication volume per round of FedAWR remains comparable to that of FedAvg, FedProx, and FedNova.

Consequently, the overall communication cost is mainly determined by the number of communication rounds required to reach a target accuracy. As demonstrated in our experiments, FedAWR converges with fewer communication rounds under heterogeneous computational settings, leading to reduced overall communication cost.

5.2.3. Ablation Study

To isolate the contribution of each component in FedAWR, we conduct ablation studies on CIFAR-10 under heterogeneous computational settings. Specifically, we consider two variants: (i) FedAWR-W, which employs adaptive aggregation weights only while using a fixed learning rate, and (ii) FedAWR-LR, which adopts adaptive learning rates only while using uniform aggregation weights. The full FedAWR incorporates both components.

As shown in Figure 8, FedAvg exhibits the slowest convergence and the lowest final accuracy, highlighting its inability to handle computational heterogeneity effectively. Both FedAWR-W and FedAWR-LR significantly improve convergence behavior compared to FedAvg, indicating that adaptive aggregation weighting and adaptive learning rate adjustment are individually beneficial.

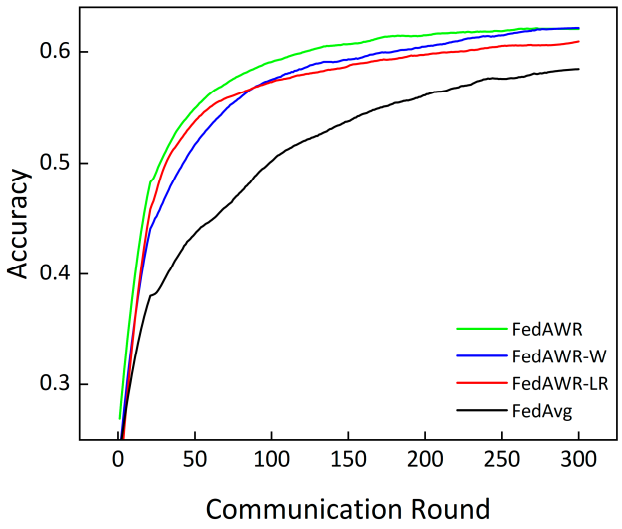

Figure 8. Test accuracy convergence curves of ablation variants on CIFAR-10 under heterogeneous computational settings.

Specifically, FedAWR-W improves convergence stability by correcting the aggregation direction through computation-aware weighting, while FedAWR-LR accelerates early-stage convergence by compensating for reduced local update magnitudes of low-capability clients. However, each variant alone shows limitations in either convergence speed or final accuracy.

In contrast, the full FedAWR consistently achieves faster convergence and higher final accuracy, demonstrating that adaptive aggregation weighting and learning rate scheduling are complementary rather than redundant. Their joint design effectively aligns both update direction and update magnitude under computational heterogeneity.

To further quantify the contributions of each component, Table 3 summarizes the convergence speed and final accuracy of the ablation variants.

Table 3. Quantitative ablation results on CIFAR-10 under heterogeneous computational settings.

Method	Target Accuracy $\geq$ 0.56	Final Accuracy
FedAvg	271	0.564
FedAWR-W	122	0.619
FedAWR-LR	209	0.609
FedAWR	**82**	**0.620**

The quantitative results further confirm that while each component individually improves performance, their combination in FedAWR achieves the fastest convergence and the highest final accuracy.

These results also motivate the following sensitivity analysis, which further examines the robustness of FedAWR to different hyperparameter settings.

5.2.4. Sensitivity Analysis of Learning Rate

To evaluate the robustness of FedAWR to hyperparameter settings, we conduct a sensitivity analysis on the client learning rate using the CIFAR-10 dataset under heterogeneous computational conditions. Since the learning rate directly affects both convergence speed and training stability, we focus on analyzing its impact while keeping other parameters fixed.

As shown in Figure 9, FedAWR demonstrates consistent convergence behavior across a range of learning rates. Learning rates of 0.001 and 0.005 achieve the fastest convergence and the highest final accuracy, while smaller learning rates such as 0.0005 and 0.0002 lead to slower convergence and lower final performance.

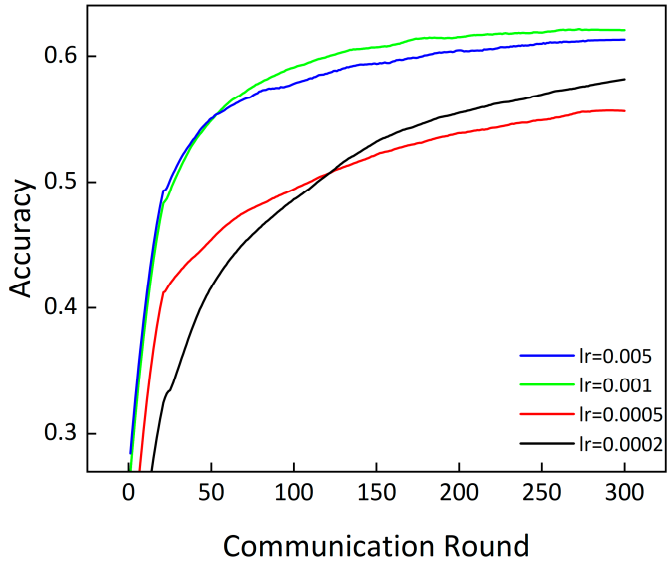

Figure 9. Sensitivity analysis of FedAWR with respect to different learning rates on CIFAR-10.

Importantly, no training instability or divergence is observed even with relatively larger learning rates, indicating that FedAWR maintains stable optimization dynamics under heterogeneous computational settings. This suggests that the adaptive aggregation and learning rate mechanisms effectively mitigate potential instability caused by aggressive local updates.

Overall, the results indicate that FedAWR is not overly sensitive to the choice of learning rate within a reasonable range, demonstrating robust performance across different hyperparameter settings.

6. Conclusions and Future Work

To address the challenges of slow convergence and reduced generalization arising from client computational heterogeneity in federated learning, we propose a novel federated aggregation optimization method, FedAWR. The core of our approach lies in the coordinated optimization of aggregation weights and local learning rates. Firstly, an adaptive weighting mechanism is introduced to enhance the contribution of low-performance clients, thereby preventing their marginalization during long-term training. Secondly, learning rates are dynamically assigned to clients based on their computational capabilities, which accelerates learning for weaker participants while ensuring global convergence. Experimental results demonstrate that, compared to mainstream algorithms such as FedAvg, FedProx, and FedNova, FedAWR achieves significant improvements in convergence speed, communication efficiency, and final accuracy, thereby yielding higher training efficiency and precision. Furthermore, validation on the CIFAR-10 dataset substantiates the efficacy of FedAWR, showing accelerated accuracy enhancement and reduced loss convergence. This validates the method's effectiveness in optimizing federated learning within heterogeneous computing environments.

This work focuses on addressing system heterogeneity in federated learning while maintaining high training efficiency and competitive accuracy. In future work, we plan to incorporate additional practical factors, such as network latency and more diverse non-IID data distributions, to further enhance the applicability of FedAWR in real-world scenarios. Moreover, while FedAWR requires clients to report lightweight computation-related statistics to enable computation-aware aggregation, such information may introduce potential side-channel risks under certain threat models. We will therefore explore privacy-enhancing techniques, such as secure aggregation and controlled noise injection, to further mitigate these risks.

Author Contributions: Conceptualization, T.Y., J.L. (Jianqi Li) and J.L. (Jianhua Liu); data collection and analysis, T.Y.; writing—original draft preparation, T.Y.; writing—review and editing, T.Y., J.L. (Jianqi Li) and J.L. (Jianhua Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (62273142, 52272347), the National Natural Science Foundation of China of Hunan Province (2024JJ7132), and the Hunan Provincial Key Laboratory for Control Technology of Distributed Electric Propulsion Aircraft.

Data Availability Statement: The datasets used in this study consist of both publicly available and private data. Due to enterprise privacy restrictions, the rail defect data are not publicly available but can be obtained from the corresponding author upon reasonable request; the CIFAR-10 dataset is publicly available at https://www.kaggle.com/datasets/pankrzysiu/cifar10-python (accessed on 14 January 2026).

Conflicts of Interest: The authors declare that this study was conducted without any commercial or financial relationships that could be considered potential conflicts of interest.

References

1. Zhang, H.; Hong, J.; Dong, F.; Drew, S.; Xue, L.; Zhou, J. A privacy-preserving hybrid federated learning framework for financial crime detection. *arXiv* **2023**, arXiv:2302.03654. [CrossRef]
2. Li, P.; Zhang, H.; Wu, Y.; Qian, L.; Yu, R.; Niyato, D.; Shen, X. Filling the missing: Exploring generative AI for enhanced federated learning over heterogeneous mobile edge devices. *IEEE Trans. Mob. Comput.* **2024**, *23*, 10001–10015. [CrossRef]
3. Lewis, C.; Varadharajan, V.; Noman, N. Attacks against federated learning defense systems and their mitigation. *J. Mach. Learn. Res.* **2023**, *24*, 1–50.
4. Zhao, R.; Falvey, J.; Shi, X.; Chinchilli, V.M.; Chen, C. A New Targeted-Federated Learning Framework for Estimating Heterogeneity of Treatment Effects: A Robust Framework with Applications in Aging Cohorts. *arXiv* **2025**, arXiv:2510.19243. [CrossRef]

5. Pei, J.; Liu, W.; Li, J.; Wang, L.; Liu, C. A review of federated learning methods in heterogeneous scenarios. *IEEE Trans. Consum. Electron.* **2024**, *70*, 5983–5999. [CrossRef]
6. Huang, W.; Ye, M.; Shi, Z.; Wan, G.; Li, H.; Du, B.; Yang, Q. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 9387–9406. [CrossRef]
7. Gong, Y.; Li, Y.; Freris, N.M. FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity. In *Proceedings of the 38th IEEE International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 2575–2587.
8. Ma, Q.; Xu, Y.; Xu, H.; Jiang, Z.; Huang, L.; Huang, H. FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3654–3672. [CrossRef]
9. Tziotis, I.; Shen, Z.; Pedarsani, R.; Hassani, H.; Mokhtari, A. Straggler-resilient personalized federated learning. *arXiv* **2022**, arXiv:2206.02078. [CrossRef]
10. Zhou, X.; Liang, W.; Kawai, A.; Fueda, K.; She, J.; Wang, K.I.-K. Adaptive segmentation enhanced asynchronous federated learning for sustainable intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6658–6666. [CrossRef]
11. Zhao, X.; Bai, X.; Sun, G.; Yan, Z. Asynchronous Federated Learning with Local Differential Privacy for Privacy-Enhanced Recommender Systems. *IEEE Internet Things J.* **2025**, *12*, 7915–7929. [CrossRef]
12. Xu, G.; Li, X.; Li, H.; Fan, Q.; Wang, X.; Leung, V.C.M. Energy-efficient dynamic asynchronous federated learning in mobile edge computing networks. In *Proceedings of the 2023 IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 160–165.
13. Bian, J.; Xu, J. Accelerating asynchronous federated learning convergence via opportunistic mobile relaying. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10668–10680. [CrossRef]
14. Xu, C.; Qu, Y.; Xiang, Y.; Gao, L. Asynchronous federated learning on heterogeneous devices: A survey. *Comput. Sci. Rev.* **2023**, *50*, 100595. [CrossRef]
15. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
16. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Demirors, E.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
17. Zheng, F.; Sun, Y.; Ni, B. FedAEB: Deep reinforcement learning based joint client selection and resource allocation strategy for heterogeneous federated learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 8835–8846. [CrossRef]
18. Chen, Q.; You, Z.; Wen, D.; Zhang, Z. Enhanced hybrid hierarchical federated edge learning over heterogeneous networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 14601–14614. [CrossRef]
19. Mayhoub, S.M.; Shami, T. A review of client selection methods in federated learning. *Arch. Comput. Methods Eng.* **2024**, *31*, 1129–1152. [CrossRef]
20. Vardhan, H.; Yu, X.; Rosing, T.; Mazumdar, A. Client Selection in Federated Learning with Data Heterogeneity and Network Latencies. *arXiv* **2025**, arXiv:2504.01921. [CrossRef]
21. Lo, S.K.; Lu, Q.; Wang, C.; Paik, H.-Y.; Zhu, L. A systematic literature review on federated machine learning: From a software engineering perspective. *ACM Comput. Surv.* **2021**, *54*, 95. [CrossRef]
22. Zhang, W.; Chen, Y.; Jiang, Y.; Liu, J. Delay-constrained client selection for heterogeneous federated learning in intelligent transportation systems. *IEEE Trans. Netw. Sci. Eng.* **2023**, *11*, 1042–1054. [CrossRef]
23. Fu, L.; Zhang, H.; Gao, G.; Zhang, M.; Liu, X. Client selection in federated learning: Principles, challenges, and opportunities. *IEEE Internet Things J.* **2023**, *10*, 21811–21819. [CrossRef]
24. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
25. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017), Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
26. Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; Poor, H.V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7611–7623.
27. Rezaei, H.; Taheri, R.; Nowroozi, E. Defense Strategies in Federated Learning Against Adversarial Attacks. In *Adversarial Example Detection and Mitigation Using Machine Learning*; Springer Nature: Cham, Switzerland, 2026; pp. 237–248.
28. Eslamnejad, M.; Taheri, R.; Shojafar, M.; Bader-El-Den, M. Federated learning-based robust android malware detection: Label-flipping attacks and defenses. *Neural Comput. Appl.* **2025**, *37*, 27057–27082. [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.