Genetic Programming in Federated Aggregation: A comparison of FedGP with State of the Art Methods

Elia Pacioni[1,2][0000−0002−1557−4870], Célien Muller[1][0009−0002−5377−0415], Francisco Fernández de Vega[2][0000−0002−1086−1483], and Davide Calvaresi[1][0000−0001−9816−7439]

[1] HES-SO Valais-Wallis, Switzerland
elia.pacioni@hevs.ch, celien.muller@students.hevs.ch, davide.calvaresi@hevs.ch
https://www.hevs.ch
[2] Universidad de Extremadura, Spain
fcofdez@unex.es
https://www.unex.es

Abstract. The efficacy of a Federated Learning system is tightly coupled to its model-aggregation strategy. Such a dependency becomes critical when client data are non-IID. To address this challenge, we undertake a comprehensive empirical assessment of FedGP, a genetic programming based aggregation framework (previously proposed for heterogeneous settings) vs state of the art FL aggregators for non-IID data.
Method. FedGP is benchmarked against three canonical baselines (i.e., FedAVG, FedPROX, and FedNOVA), using two clinically relevant image collections, *PathMNIST* and *PneumoniaMNIST*. Experiments are executed under both IID and deliberately skewed non-IID partitions. Performance is scrutinized through aggregate metrics (accuracy and F1-score), learning curve trajectories, distributional box plots, and formal significance testing.
Results. In IID regimes, FedGP yields a modest yet consistent edge over the comparators. Under non-IID conditions, however, FedGP delivers marked and statistically significant gains in accuracy, F1-score, and training stability, whereas the reference methods exhibit pronounced variance and occasional collapse.
Conclusions. The evidence substantiates the robustness and superior generalization of FedGP in heterogeneous federated environments. Although its genetic programming search entails additional computation, the overhead remains tractable for centrally orchestrated deployments. Future research will focus on reducing this cost through transfer-learning warm starts and parallel evaluation, extending FedGP to asynchronous or fully decentralized topologies, and incorporating agent-based orchestration for personalized model specialization.

Keywords: FedGP · Federated Learning · Genetic Programming · Multi-Agents System · Models Aggregation

1 Introduction

In recent years, Federated Learning (FL) has emerged as a central paradigm in distributed machine learning, enabling models to be trained on decentralized data without the need to transfer sensitive information to a central server [26, 14]. Introduced by Google in 2016, FL has found widespread application in critical domains such as healthcare [38] and finance [21], where data protection is essential. A well-known example is Gboard, where predictive models are trained directly on Android devices to improve user experience while preserving privacy [11].

At the core of the FL process lies the aggregation of local models. In this phase, devices send their locally trained models (or the corresponding weights) to a central server, which combines them using a predefined aggregation strategy [26]. This approach maintains data locality, reduces communication overhead, and safeguards confidentiality. The most widely adopted method is FedAVG, based on the weighted average of updates according to local data size. However, FedAVG has known limitations under non-IID (non-independent and non-identically distributed) conditions, where data variability across clients may hinder the generalization capability of the global model [31, 37, 20, 42].

Variants of FedAVG—such as unweighted averaging—have been explored to enhance privacy, but often fail to address heterogeneity effectively. Moreover, FL scalability is hindered by device diversity and uneven quality of local updates [20, 31]. Another significant issue is communication inefficiency: standard FL transmits all client updates indiscriminately to the server, resulting in high bandwidth usage, particularly in large-scale or resource-constrained environments, such as Internet of Things (IoT) networks [16]. Techniques such as parameter compression [15], federated dropout [6], and structured updates [51, 15, 43] offer partial solutions but lack adaptability to FL's dynamic nature.

To tackle these challenges, in our previous work [33, 34] we introduced FedGP, an aggregation method based on Genetic Programming (GP). GP, known for its effectiveness in solving adaptive and complex problems, is used here to dynamically generate mathematical functions that aggregate client models, thereby overcoming the limitations of static formulas, such as averaging. FedGP thus approaches the aggregation challenge as a symbolic regression problem, improving generalization and reducing bias.

Supporting this flexibility is a multi-agent system (MAS) architecture, where each client is modeled as an autonomous, reactive agent. This structure enables decentralized orchestration, facilitates experimentation, and enhances the scalability and personalization of FL systems, aligned with recent architectural evolutions [3].

In this extended study, we systematically compare FedGP with three widely adopted state-of-the-art aggregation methods: FedAVG, FedPROX [20, 19], and FedNOVA [42]. These algorithms were selected for their relevance, complementarity, and practical adoption. Experiments are conducted on two real-world medical datasets, PathMNIST and PneumoniaMNIST [46], under both IID and

non-IID conditions, within a centralized and synchronous FL architecture where clients are implemented as independent software agents.

Results demonstrate that FedGP maintains high accuracy and stability even in heterogeneous non-IID scenarios, where other strategies exhibit significant performance degradation. Despite its higher computational cost, FedGP's superior generalization capability makes it particularly well-suited for high-stakes domains such as clinical diagnostics. The study also includes statistical significance testing, accuracy curves, and an analysis of execution times.

The remainder of this paper is organized as follows: Section 2 presents the theoretical background and the aggregation methods considered; Section 3 details the experimental methodology; Section 4 presents and analyzes the results; and Section 5 offers conclusions and directions for future work.

2 State Of The Art

Since its inception, FL has become a cornerstone of decentralized ML. Among the advantages, FL enables distributed apparatus (e.g., mobile and IoT devices) to exploit the vast quantities of data they generate locally without relinquishing custody of the raw information. By eschewing central data collection, FL has transformed distributed intelligence across an array of domains – including mobile services such as Gboard's next-word prediction [11] and Apple's Siri speech models [10], privacy-sensitive healthcare collaborations [41], and privacy-aware financial analytics [14], while maintaining stringent data-protection guarantees.

At its core, FL proceeds through a cyclical protocol in which client devices optimize local models on private data and transmit only model updates (e.g., weights or gradients) to a coordinating entity for aggregation. This coordination can follow a centralized topology, where a single server fuses the updates, or a decentralized topology, where participating devices collectively fulfill the aggregation role, thereby mitigating single-point-of-failure and adversarial risks [3]. Complementing the architectural choice is how data are partitioned. Horizontal partitioning distributes examples possessing the same feature space across clients, whereas vertical partitioning disperses complementary feature spaces across clients that share the same samples, demanding sophisticated alignment mechanisms [47, 45]. The synchronization policy governs communication efficiency and timeliness.

Synchronous FL aggregates only after all clients have completed a training round, incurring stragglers' delay in heterogeneous environments; asynchronous FL integrates updates on arrival, accelerating progress but complicating convergence analysis [45]. Meanwhile, a rich toolbox of privacy-preserving technologies—differential privacy, secure multiparty computation, and homomorphic encryption – safeguards both data and intermediate gradients throughout the protocol [29, 24]. To further constrain communication overhead, gradient compression and quantization strategies are routinely employed.

Model aggregation remains the algorithmic linchpin of FL. The ubiquitous FedAVG algorithm computes a data-size-weighted mean of client updates [36].

However, its efficacy deteriorates when client data are non-IID, yielding a generic global model that may underperform on individual distributions. Contemporary research thus explores personalization layers, local finetuning, meta-learning, and other adaptive schemes designed to reconcile cross-client heterogeneity [9]. Ultimately, the pursuit of rapid adaptability in non-IID and resource-constrained conditions continues to drive novel methodological innovations. Meta-learning frameworks expedite adaptation from scant data [9], while bio-inspired optimization algorithms have begun to offer promising avenues for balancing transmission efficiency with predictive performance [39]. Collectively, these advances delineate a fervent research landscape aimed at rendering FL both robust and agile to face real-world complexity.

2.1 IID vs non-IID - Importance of data distributions

In the context of FL, IID (Independent and Identically Distributed, Figure 1) refers to a data distribution in which each client has a local dataset representative of the same global statistical distribution. Conversely, a non-IID (non-Independent and Identically Distributed, Figure 2) scenario refers to a situation in which data distributions vary significantly from one client to another, resulting in heterogeneous or unbalanced distributions [20, 52].

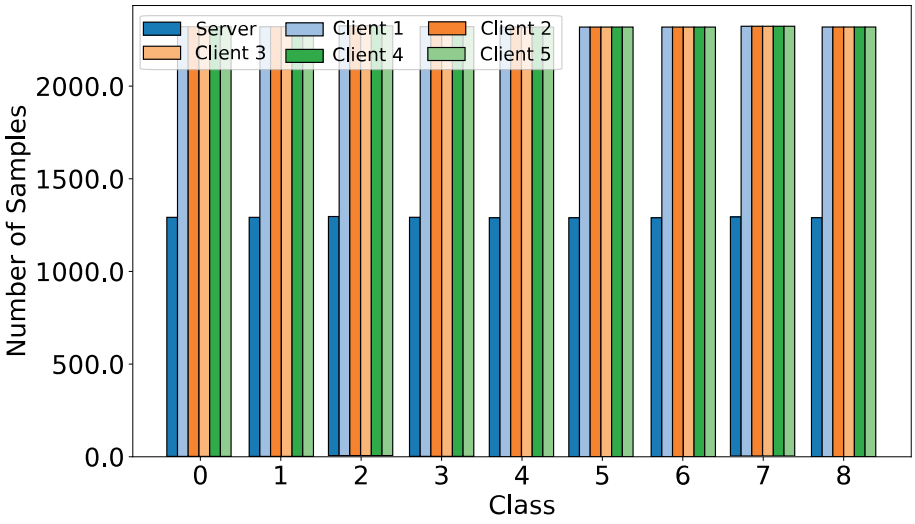

Fig. 1. Example of IID data distribution in FL context.

This situation is the norm in many real-world FL use cases. It stems from the fact that each client—for example, a smartphone, sensor, or local institution—collects data independently from its user or environment, reflecting specific preferences, behaviors, or conditions [49]. As a result, local datasets tend

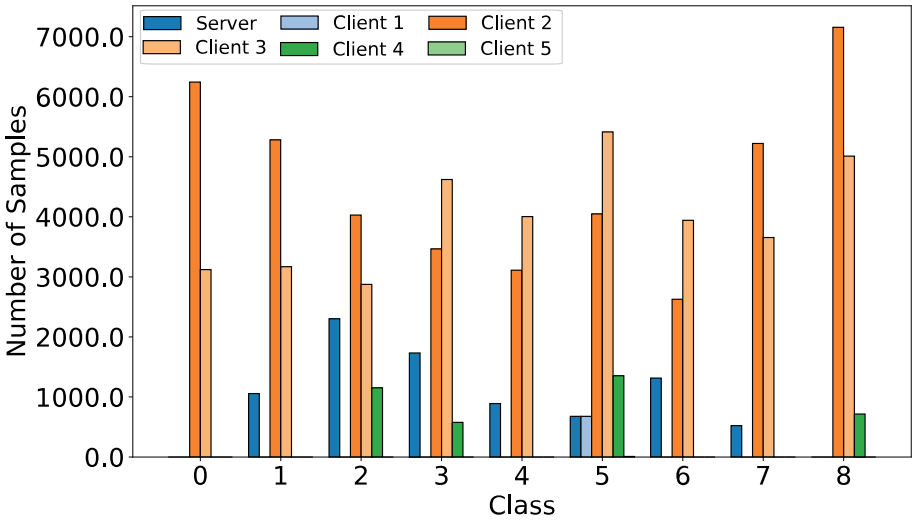

Fig. 2. Example of non-IID data distribution in FL context.

to have different distributions, meaning that some devices may have very different amounts of data (imbalance) and content that does not reflect the overall distribution.

Due to the decentralized nature of FL, the central server typically does not have direct knowledge of the unified global data distribution, as raw data remains private on devices. Despite this, the literature has observed that the presence of strong non-IID in local data can degrade the performance of the global model. In other words, if client data has marked statistical differences, the resulting federated model tends to have lower accuracy than in the IID case due to biases introduced by the dominance of specific clients or the model's difficulty in reconciling overly diverse distributions.

Studies conducted on FL identify different forms of data heterogeneity among clients [52]:

- Label skew: different class distributions across clients (e.g., some devices collect mainly data from specific classes, others from different classes).
- Quantity skew: variability in the number of samples owned by each client, with local datasets of very different sizes. - Feature skew: differences in the distribution of data features (e.g., images with different resolutions, styles, or conditions on different clients).
- Temporal skew: data from some clients comes from different periods, generating shifts over time.

These distributional differences introduce significant difficulties in the FL process. In particular, updates to local models may diverge from the common optimal direction, resulting in a degradation of the global model's performance

and a slowdown in convergence. Empirical studies confirm that the presence of strong non-IID data can lead to drastic drops in accuracy.

In recent years, FL research has proposed numerous methods to mitigate the adverse effects of non-IID data. One mitigation strategy is to reduce the divergence between local distributions, for example, by sharing a small subset of global data among all clients; however, this approach compromises user privacy and security. Other approaches intervene in the federated optimization algorithm; for example, the Federated Proximal (FedPROX) method is presented in Section 2.2. Despite these advances, the issue of non-IID data remains open and central to FL.

2.2 Aggregation Methods

FedAVG Introduced by McMahan et al. [26] and widely recognized as the de facto standard in FL, FedAVG consists of a simple average (weighted or unweighted) of the parameters updated by the clients. At each round, the server distributes the global model to the devices; each client performs local training epochs and returns the weights. The server calculates the averages, weighted or unweighted, and propagates the new model to the clients.

FedAVG is the essential reference for three main reasons: (i) *Simplicity and scalability* — implementation requires only the weighted average of the models, without regularization terms or additional hyperparameters; (ii) *Widespread adoption* — most subsequent algorithms have been developed as extensions or modifications of FedAVG, making it the most common baseline in scientific publications [26]; (iii) *Design guidelines* — the project specifications require the inclusion of FedAVG among the comparison aggregators.

Although effective in scenarios with relatively homogeneous data distributions, FedAVG has known limitations in the presence of non-IID data and uneven client participation, conditions that can compromise the convergence and quality of the global model.

FedPROX Proposed by Li et al. [19], *FedPROX* extends FedAVG with a minimal but crucial modification: the addition of a proximal term, represented in the equation 1 to the objective function optimized by each client. This penalty "anchors" local updates to the current global model w_t, reducing the phenomenon of client drift that arises when data are non-IID or local epochs vary across devices. As μ increases, the constraint becomes more stringent; at the limit $\mu = 0$, the algorithm effectively reduces to FedAVG.

$$\frac{\mu}{2} \|w - w_t\|^2 \tag{1}$$

Thanks to this simple modification, FedPROX is particularly well suited to realistic scenarios in which participating devices have different statistical distributions or non-uniform computational resources. The proximal term guarantees: (i) robustness to statistical and system heterogeneity because it limits the divergence between local and global models; (ii) stability of convergence even with

intermittent participation or reduced local epochs; (iii) flexibility in the amount of work that each client can perform, making the algorithm suitable for heterogeneous device networks; (iv) theoretical guarantees of convergence under assumptions of smoothness and limited variance, extended to both convex and non-convex cases.

From an operational point of view, the server selects a subset of clients, transmits the model w_t , and aggregates—using weighted averaging—the updated parameters $w_i^{(t+1)}$ received from the participants. The implementation, therefore, remains lightweight and easily integrated into existing frameworks: it only requires the new hyperparameter μ, which can be adjusted according to the degree of heterogeneity observed. Due to its simplicity and theoretical soundness, FedPROX is now one of the reference methods for FL in real environments and is frequently used as an advanced baseline for evaluating more sophisticated aggregation algorithms.

Federated Normalized Averaging (FedNOVA) FedNOVA, introduced by Wang et al. [42], addresses the problem of objective inconsistency that arises when clients perform different numbers of local optimization steps. In FedAVG, a device that performs many more iterations of SGD contributes a disproportionate update, causing an "imbalance" in the global gradient. FedNOVA addresses this critical issue by normalizing each local update Δ_i concerning the actual number of steps m_i performed by client i. More precisely, at the end of round t, the server calculates

$$w_{t+1} \;=\; w_t \;-\; \eta \sum_{i \in \mathcal{S}_t} \frac{n_i}{N} \, \frac{\Delta_i}{\zeta_i}, \qquad \zeta_i \;=\; \sum_{j=0}^{m_i - 1} (1 - \eta\lambda)^j \tag{2}$$

where n_i is the cardinality of the local dataset, $N = \sum_i n_i$ is the total number of samples, and η is the learning rate. The factor ζ_i acts as a normalization coefficient, ensuring that each client's influence is proportional to its data and not to the computational work performed.

FedNOVA, therefore, maintains: (i) *Computational fairness* — updates are weighted to reflect the informational quality of the data rather than the computing power of the device; (ii) *Robustness to system heterogeneity* — convergence remains stable even with very different local epochs or batch sizes; (iii) *Generality* — normalization applies to a broad class of optimization methods (SGD, Adam, etc.) without introducing additional hyperparameters at the client level; (iv) *Theoretical guarantees* — the algorithm converges in both IID and non-IID data settings, providing upper bounds on the error rate.

Thanks to these properties, FedNOVA provides an advanced baseline for evaluating federated algorithms in scenarios characterized by strong computational and statistical asymmetry.

2.3 GP and its role in FL

Koza et al. [17] introduced GP as a technique capable of optimizing nonlinear functions and dynamically generating original solutions. Since then, GP has branched out into numerous variants, each with a distinctive contribution to the basic methodology. Among the most relevant are: (i) *Linear GP* [7], which arranges programs as linear sequences of instructions, improving their readability and efficiency; (ii) *PushGP* [40], based on the Push language to manage complex data structures and evolve flexible programs; (iii) *Cartesian GP* [27], which represents programs as directed acyclic graphs, offering a highly modular structure; (iv) *Geometric Semantic GP* [28], which uses geometric semantic operators to accelerate research and obtain more robust solutions; (v) *Grammatical Evolution*) [32], which uses formal grammars to guide evolution and allows code to be produced in specified languages; (vi) *Genetic Improvement* [35], aimed at improving existing software without altering its functionality, increasing its performance and adaptability.

GP has found successful applications in heterogeneous domains: in the field of creativity, for example, the design of a Portuguese commemorative coin [23], from the generation of behaviors and controllers in video games [44] to the analysis of medical images [18], and even the prediction of financial option prices [12].

Among the best-known uses is *symbolic regression* [2], which aims to derive mathematical models that can accurately describe a set of data. In FL, it is necessary to define a mathematical function—the aggregator—that combines updates from multiple clients; this coincides with the objective of symbolic regression. GP is, therefore, a natural candidate for developing aggregation strategies suitable for heterogeneous data and non-homogeneous client resources.

Thanks to its ability to adapt to complex problems, GP can enhance aggregation methods in FL [34]. However, traditional methodologies, such as FedAVG, have significant limitations in the presence of non-iid distributions [14, 21, 22], highlighting the need for more sophisticated approaches.

Federated GP Aggregation (FedGP) FedGP [34] is an aggregation method for FL developed to overcome the limitations of traditional methods, particularly in non-IID scenarios. Pacioni et al. theorized and presented FedGP [33, 34] in 2025. The approach is based on the use of GP to dynamically evolve aggregation functions, offering a flexible, adaptive, and customized alternative. Instead of a fixed formula (such as the mean), FedGP evolves complex mathematical expressions represented as expression trees, constructed from a series of PyTorch primitives (e.g.,*mean, median, abs, mul, protected_ div, etc.*) applied to the client weight tensors.

In FedGP, candidate solutions (individuals in the population) are evaluated by the fitness function based on the accuracy values obtained on the test dataset. During each aggregation phase, FedGP applies the best mathematical function (the best individual) to the weights received from clients to produce a new global model. Figure 3 shows the typical workflow applied by FedGP and the main difference compared to FedAVG.

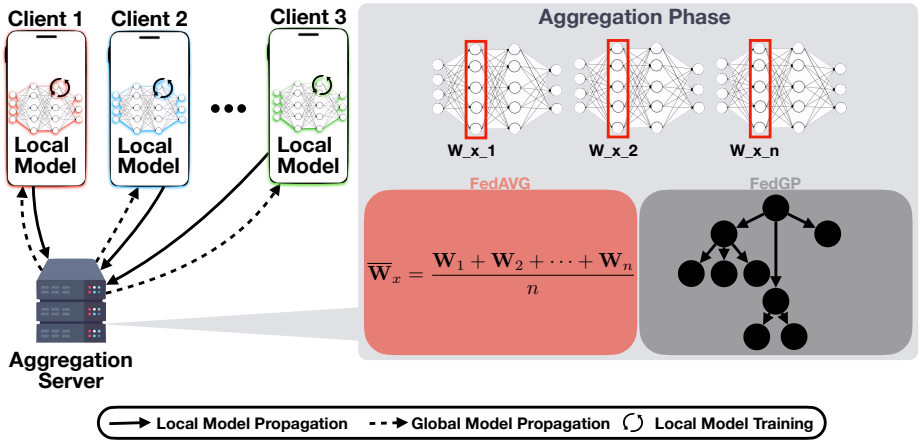

Fig. 3. Typical FL workflow with a comparison between FedAVG and FedGP.

FedGP has been tested on the PathMNIST dataset. Comparison with unweighted FedAVG showed that FedGP has higher accuracy, greater robustness to non-IID distributions, and, in some cases, outperformed the best client, thus improving generalization capabilities.

2.4 Agentifications of FL: optimization and personalization

MAS represent a well-established paradigm in distributed systems engineering and artificial intelligence. In a typical MAS architecture, a system consists of a set of autonomous agents, i.e., computational entities capable of perceiving their environment, making decisions locally, and interacting—either cooperatively, competitively, or both—with other agents to achieve individual or collective goals.

This modeling has proven particularly effective in dynamic environments such as intelligent sensor networks [30], decentralized control systems [5], adaptive communication protocols [13], and, more recently, distributed learning contexts [25]. The adoption of MAS architectures enables operational flexibility, local adaptation, scalability, and robustness —crucial characteristics for modern, complex systems.

From an engineering perspective, agents are often designed to support three fundamental capabilities:

- autonomy (they do not require central supervision to act);
- reactivity and proactivity (they respond to the environment but also pursue their own goals);
- coordination (they interact to maximize collective performance or their utility).

In practice, this translates into systems where collective behavior emerges from locally managed and decentralized interactions rather than centralized planning. MAS has also proven effective in the presence of heterogeneous interests and asymmetric resources, two characteristics frequently found in modern edge computing and IoT scenarios [1].

In recent years, the MAS paradigm has also found application in the field of FL, where the problem of large-scale distributed ML presents structural and functional similarities with MAS systems. In particular: (i) Each client in an FL system can be seen as an agent, which has computational resources, private local data, and the ability to learn models locally. (ii) The central server can be modeled as a coordinating meta-agent whose task is to aggregate information from client agents, manage synchronization, and regulate the progression of collective learning.

This agent-centric interpretation introduces significant benefits over the rigid client/server architecture traditionally used in FL:

- Client decision autonomy: Client agents can decide when and whether to participate in learning rounds based on local criteria such as energy availability, data quality, network congestion, or local error variation. This approach is consistent with selective communication strategies, such as event-triggered communication [48], and allows for optimization of network traffic and latency.
- Scalability and decentralization: In large-scale scenarios (thousands or millions of clients), direct continuous interaction between servers and clients becomes impractical. A MAS structure enables the emergence of coordinated and scalable behaviors, potentially utilizing regional servers or hierarchical agent structures [4]
- Adaptability to dynamic conditions: Clients in FL may be subject to significant variations, including disconnections, changes in data distribution, and changes in computational resources. Modeling them as reactive and proactive agents enables them to adapt without relying on static rules imposed by the server.
- Personalization and cooperative negotiation: In non-IID contexts, it has been recognized that a single global model may not be optimal for all clients. Agents can pursue personalized strategies, negotiate the loss function with the server, or selectively participate in training to maximize their local utility [50].
- Integration with intelligent techniques: Agent-centric modeling enables the seamless integration of reinforcement learning, evolutionary optimization, and symbolic reasoning techniques, which would be more challenging to incorporate into a traditional client/server view. For example, several recent works [8] employ intelligent policies in clients to improve action selection (e.g., communication, updating, outlier filtering).

As a result, the use of MAS in FL is not just an implementation choice but a natural architectural evolution for autonomous, scalable, and customizable distributed systems.

3 Methodology

To overcome the limitations of traditional aggregation methods, such as FedAVG, in FL contexts with non-IID data, we have introduced FedGP, a method based on GP designed to improve model aggregation. FedGP dynamically adapts aggregation functions to handle data variability across clients. Initially, its performance was compared with that of FedAVG; in this work, we extend the analysis to include FedPROX and FedNOVA, known to be more robust and accurate. The entire experiment was conducted using the same configurations and parameters as in the previous study. The evaluation focuses on two main aspects: model accuracy and the time required for aggregation.

The methodology adopted is divided into five main phases, consistent with the development of the project: selection of the reference dataset (3.1); choice of aggregation algorithms (3.2); configuration of test environments and execution of experiments (3.3); finally, collection and analysis of the results obtained (3.4).

For the technology stack, we reused the one presented in [34]. We used the DEAP and PyTorch libraries with Python 3.11. We organized the servers and clients as autonomous agents.

3.1 Dataset Selection

We employ two datasets from the MedMNIST v2 collection [46], specifically designed for lightweight, standardized, and accessible biomedical image classification tasks. The first dataset, PathMNIST, comprises 107,180 histopathological images of colorectal tissue, each with a resolution of 28 $\times$ 28 pixels and RGB channels. The images are extracted from digitized pathology slides of hematoxylin-and-eosin-stained samples and annotated into nine distinct classes, representing relevant tissue types and histological structures: (i) adipose tissue, (ii) background, (iii) debris, (iv) lymphocytes, (v) mucus, (vi) smooth muscle, (vii) normal colon mucosa, (viii) cancer-associated stroma, and (ix) colorectal adenocarcinoma epithelium. Unlike studies that rely on pre-extracted features, our experiments operate directly on raw image data, preserving visual patterns that are crucial in clinical diagnosis. The high number of samples and class variety make PathMNIST particularly suitable for evaluating FL systems in challenging multi-class non-IID scenarios.

To complement this, we include PneumoniaMNIST, which enables testing under a binary classification setting. This dataset contains 5,856 grayscale chest X-ray images, each resized to 28$\times$28 pixels and aims to distinguish between normal lungs and cases of pneumonia, primarily in pediatric patients. PneumoniaMNIST provides a clinically relevant binary diagnostic task with inherent class imbalance and inter-patient variability — characteristics commonly encountered in real-world applications.

The combination of PathMNIST and PneumoniaMNIST enables us to evaluate the proposed approach in both multi-class and binary classification settings, spanning a broad spectrum of complexity. This dual setup reflects practical deployment scenarios, where federated systems must generalize across different

diagnostic tasks while adapting to varied data distributions and task granularities.

To simulate non-IID conditions, we implemented a combination of label skew and quantity skew across the clients. Specifically, each client receives samples from a limited subset of classes (label skew), and the number of samples assigned to each client varies (quantity skew). This setup reflects realistic heterogeneity observed in federated medical scenarios, where different hospitals or devices may specialize in different types of cases and collect varying amounts of data.

An overview of the two selected datasets and their characteristics within the broader MedMNIST collection is shown in Figure 4.

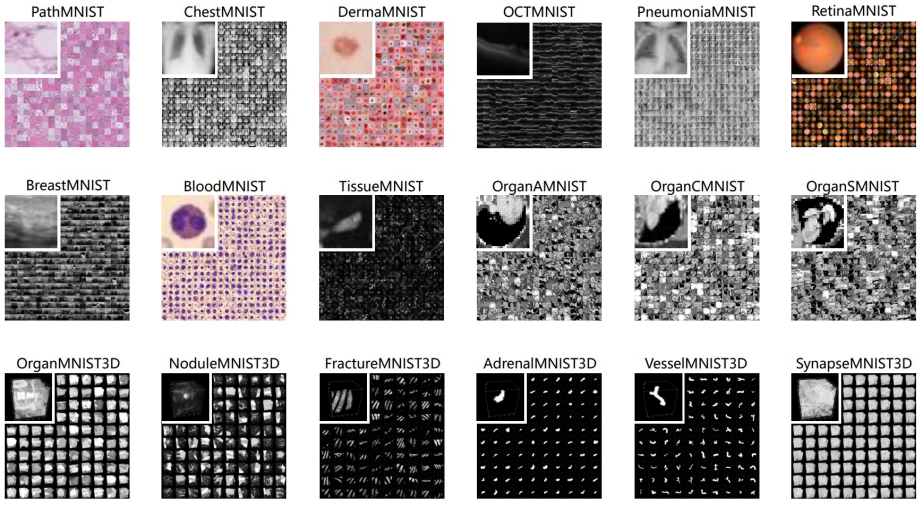

Fig. 4. Overview of the datasets that constitute the MNIST collection (source: https://medmnist.com)).

3.2 Aggregation method Selection

There are dozens of variants of federated aggregation methods, each optimized for different scenarios and assumptions. Comparing them all is beyond the scope of this study; we therefore decided to focus on a representative subset consisting of FedAVG, FedPROX, and FedNOVA, in addition to the proposed FedGP method. The choice was guided by three main criteria: (i) prevalence in the literature and practical adoption, (ii) complementarity in the way they address data and device heterogeneity, and (iii) availability of theoretical results and stable implementations that ensure a fair comparison.

FedAVG as a consolidated baseline. FedAVG is the reference algorithm [26] on which most subsequent extensions are based. Its operational simplicity, weighted

average of parameters after a fixed number of local epochs, and widespread adoption in publications make it the essential baseline. Comparing FedGP with FedAVG allows us to evaluate the gain introduced by our approach compared to the de facto standard.

FedPROX for managing heterogeneity. FedPROX introduces a proximal term

$$\mu \|w - w_t\|^2$$

into the local objectives, anchoring updates to the global model and reducing so-called *client drift*. This minimal modification to FedAVG has been shown to stabilize convergence in the presence of non-IID distributions and partial device participation (scenarios that are very common in healthcare, where data remains fragmented across hospitals) [20, 19]. Since FedGP also aims to address heterogeneity through advanced aggregation functions, comparing the two methods highlights their differences and similarities.

FedNOVA to normalize uneven updates. FedNOVA addresses the issue of objective inconsistency resulting from the varying numbers of SGD steps executed locally by clients. The method realigns updates through normalization, ensuring an unbiased global gradient and improving convergence when devices have highly variable computing power or network availability [42]. Including it in the comparison enriches the analysis because FedNOVA represents an orthogonal strategy—based on statistical reweighting rather than constraints or genetic evolution—to counteract the exact source of heterogeneity that FedGP aims to address.

In summary, the FedAVG–FedPROX–FedNOVA combination covers, as a whole, the main approaches currently adopted to make federated aggregation robust: naive averaging, proximal regularization, and update normalization. Comparing FedGP with these methods, therefore, provides a clear picture of the benefits (or possible limitations) introduced by our strategy, both in terms of accuracy and computational efficiency, while maintaining comparability with the most cited studies in the literature.

3.3 Tests performed

Since this study is an extension of previous articles [33, 34], all tests were conducted using the same global hyperparameters: *(i)* 5 federated clients; *(ii)* 2 epochs of global model training; *(iii)* 15 epochs of local training for each client. The convolutional neural network was optimized using SGD with a learning rate of 0.0005 and momentum of 0.9; the objective function is the unweighted CrossEntropyLoss.

The evolutionary parameters adopted are 10 generations, elitism set to 1, mutation rate of 40%, and crossover rate of 60%. To balance the complexity of the generated aggregation functions and their generalization ability, the depth

of the trees was limited to a minimum of 1 and a maximum of 5 levels. The GP primitives as shown in equation (3).

$$F = \begin{Bmatrix} \text{torch.sum, torch.sub, torch.mul,} \\ \text{torch\_protected\_div, torch\_mean,} \\ \text{torch\_median, torch.abs, torch\_protected\_sqrt} \end{Bmatrix} \tag{3}$$

The entire workflow was performed in separate virtual environments, simulating the presence of distinct physical devices and thus isolating variables not relevant to the study. The tests were performed on a distributed cluster of Ubuntu 22.04 nodes managed by SLURM. Most jobs ran on nodes equipped with Nvidia A2 GPUs (15 GB of VRAM) and Intel Xeon E5-2690 v3 CPUs (12 cores, 24 threads), along with 125 GB of RAM. It should be noted that, in some time windows, other users ran processes on the same servers without SLURM mediation, which may have affected execution times due to the sharing of computational resources.

To ensure the statistical robustness of the results, each configuration was repeated 5 times under identical conditions. All experiments adopt horizontal data partitioning and a centralized, synchronous federated architecture; nevertheless, the proposed methodology lends itself to being extended, with minimal modifications, to asynchronous and decentralized scenarios.

3.4 Comparison and Analysis

For each execution, the accuracy values for every epoch of both the server and all clients are recorded, as well as data from each aggregation phase. Using the collected information, the mean and standard deviation for the server, clients, and aggregate are computed. Afterward, the data is organized for analysis and processed to create a graph that facilitates the comparison of aggregation methods for the selected dataset and data distribution. In addition to the visual analysis, statistical significance tests are performed, and summary box plots are presented. All statistical analyses were conducted using pairwise t-tests (or Welch's t-test when variances were unequal), after verifying the assumptions of normality (Shapiro-Wilk test) and homogeneity of variances (Levene's test). Significance was assessed at the 0.05 level of statistical significance. Finally, the execution times of each algorithm are analyzed and compared to assess their efficiency and performance under varying conditions.

4 Results

The results of the experiments are structured by dataset to enhance the comparison of aggregation methods for both IID and non-IID distributions.

For each type of experiment, a table presents summary metrics, while a graph illustrates the accuracy for each round of aggregation. Additionally, box plots, along with statistical significance tests, are provided. Following this, the execution time for each aggregation method is analyzed.

Concluding the section, a comprehensive summary of the results is offered to provide the reader with a clear overview.

4.1 Experiments on the PathMNIST dataset

This subsection analyzes the results of the PathMNIST dataset on IID and non-IID data.

IID data. Table 1 shows the results of the experiments with IID data distribution. In absolute terms, FedGP achieves better results than the other aggregation methods, with a lower standard deviation. However, in the case of IID distribution, aggregation is simpler, so all methods have similar accuracy and F1 values.

Table 1. Aggregated performance metrics for FedAVG, FedPROX, FedNOVA, FedGP. Dataset: PathMNIST, Distribution: IID.

Method	Acc	St.Dev	F1	St.Dev
FedPROX	77.128	2.051	0.726	0.017
FedNOVA	77.220	2.456	0.718	0.020
FedGP	**79.716**	**1.102**	**0.743**	**0.018**
FedAVG	78.833	1.572	0.738	0.017

Figure 5 shows the accuracy values for each aggregation round. As the number of aggregation rounds increases, the accuracy values improve for all methods. From the first round onwards, FedGP is the best, while FedPROX and FedNOVA offer similar results. The same results can be observed in the box plot in Figure 6.

We proceed to perform paired statistical significance tests to assess whether there are significant differences between the models. After verifying the conditions of applicability, we perform paired t-tests. In this case, we obtain all $p - values > 0.05$, so we can conclude that there are no statistically significant differences between the aggregation methods tested in the case of the PathMNIST dataset with an IID distribution.

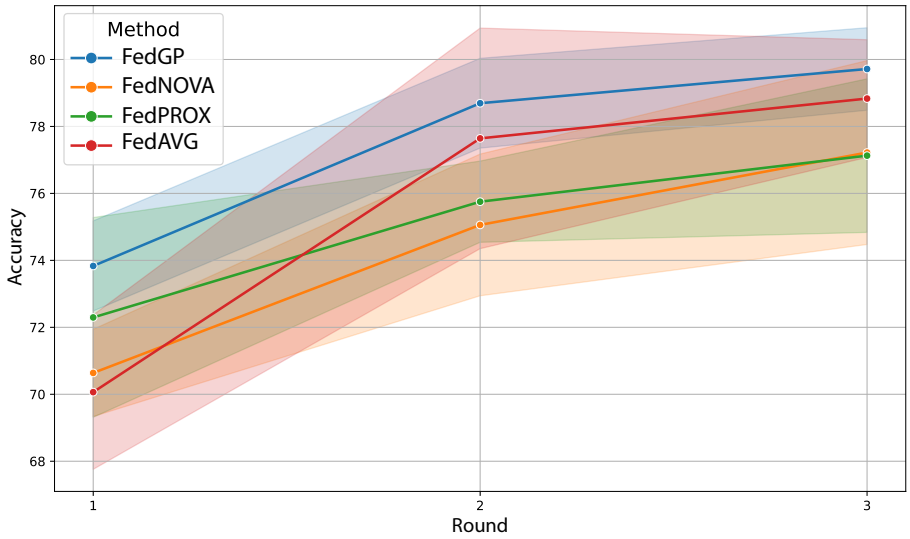

Fig. 5. Aggregation accuracy on the PathMNIST dataset with IID distribution.

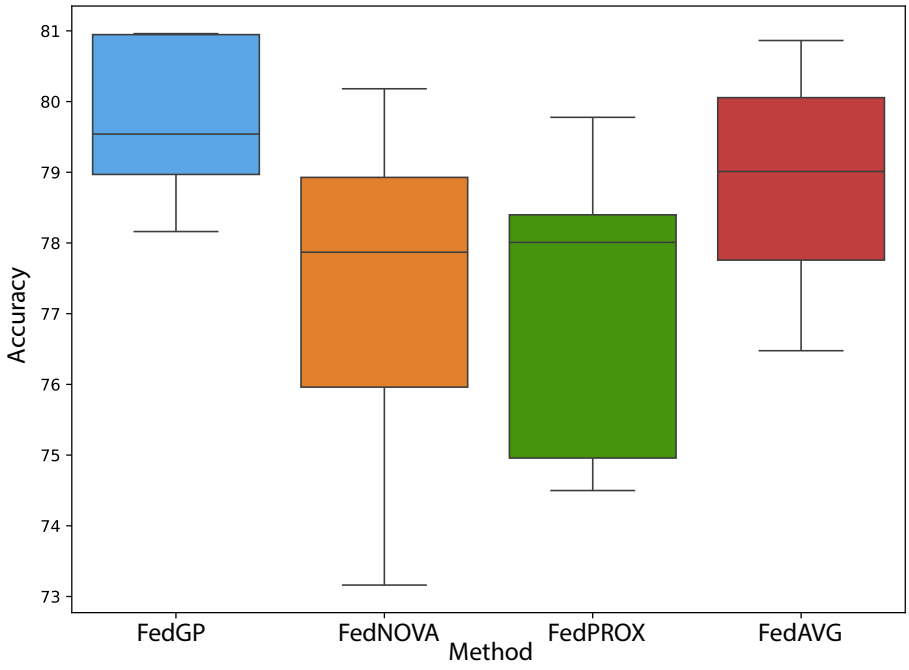

Fig. 6. Boxplot of experiments on the PathMNIST dataset with IID distribution.

non-IID data. For non-IID data, Table 2 shows a deterioration in performance overall for FedPROX, FedNOVA, and FedAVG. In particular, FedAVG shows the most significant deterioration, falling from an accuracy of 78.88 to 62.17 and increasing the standard deviation by about 10 times. Even when analyzing the F1 metric, the performance degradation is similar.

Table 2. Aggregated performance metrics for FedAVG, FedPROX, FedNOVA, FedGP. Dataset: PathMNIST, Distribution: non-IID.

Method	Acc	St.Dev	F1	St.Dev
FedPROX	65.97	10.86	0.62	0.09
FedNOVA	67.84	5.36	0.63	0.05
FedGP	**78.39**	**3.62**	**0.72**	**0.05**
FedAVG	62.17	12.14	0.56	0.13

However, FedGP maintains results similar to the IID version, with a 1% degradation in accuracy and a slight deterioration in standard deviation.

Figure 7 shows how FedGP achieves superior results compared to other aggregation methods, despite a decline in the second round of results. Figure 8), shows events and outliers for both FedPROX and FedAVG. Overall, FedNOVA, FedPROX, and FedAVG achieve similar performance.

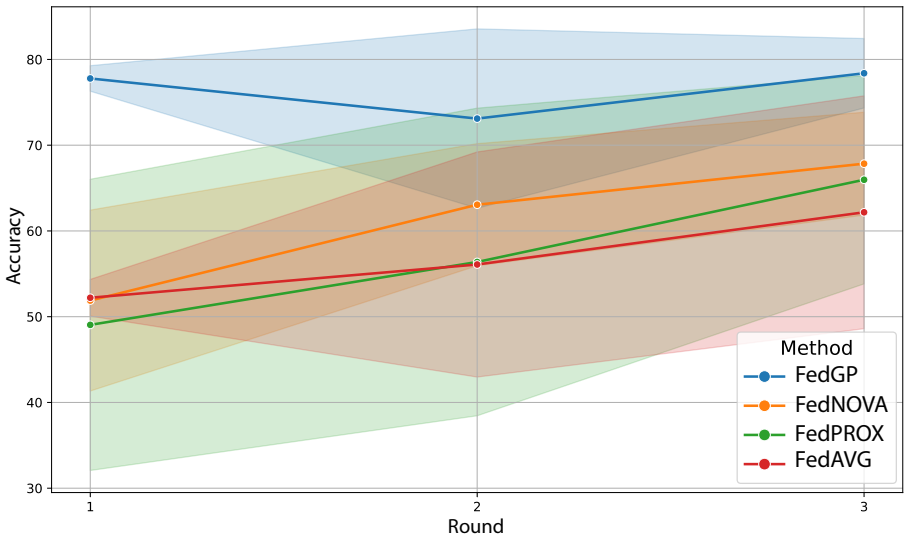

Fig. 7. Aggregation accuracy on the PathMNIST dataset with non-IID distribution.

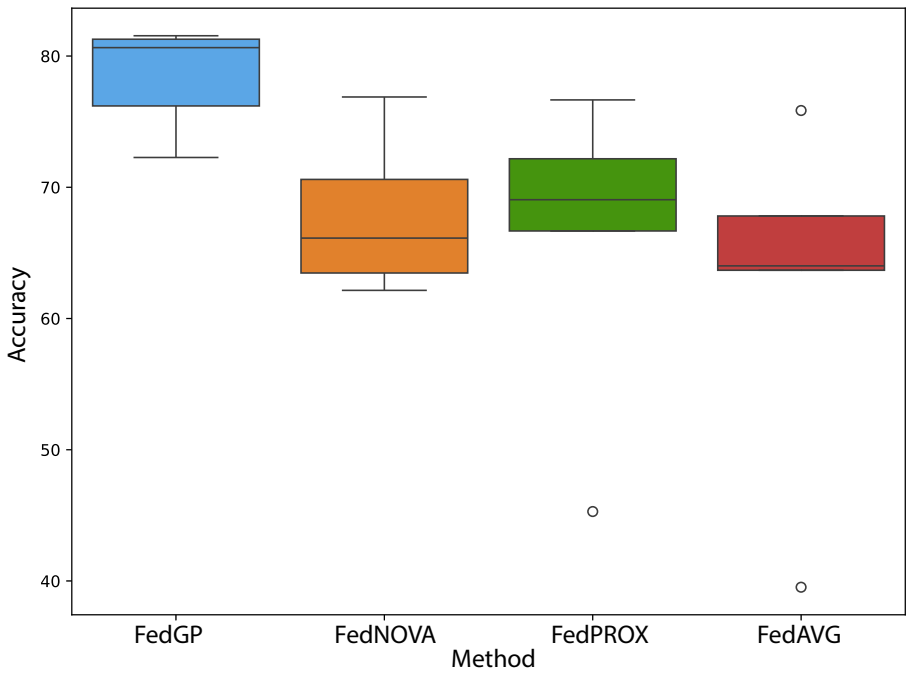

Fig. 8. Boxplot of experiments on the PathMNIST dataset with non-IID distribution

In this case, we also performed statistical significance tests (*t-test*). The results are summarized in Table 3. We observe that the difference between FedGP and FedNOVA is significant, with a p-value of 0.011; the same is true between FedGP and FedAVG, with a p-value of 0.034. However, no statistically significant differences were found between FedGP and FedPROX. Therefore, we can conclude that, in this case, FedGP is superior to FedNOVA and FedAVG.

Table 3. Statistical significance test (t-test) in the case of the PathMNIST dataset and non-IID distribution

Method 1	Method 2	p-value
FedGP	FedNOVA	**0.011**
FedGP	FedPROX	0.062
FedGP	FedAVG	**0.034**
FedNOVA	FedPROX	0.076
FedNOVA	FedAVG	0.418
FedPROX	FedAVG	0.654

4.2 Experiments on the PneumoniaMNIST dataset

IID data. Table 4 reports the aggregated results of the FL aggregation methods evaluated on the PneumoniaMNIST dataset under an IID distribution setting. Among the methods, FedGP achieves the highest accuracy (86.80%) and F1-score (0.85), substantially outperforming the other approaches.

Table 4. Aggregated performance metrics for FedAVG, FedPROX, FedNOVA, FedGP. Dataset: PneumoniaMNIST, Distribution: IID.

Method	Acc	St.Dev	F1	St.Dev
FedPROX	83.08	1.12	0.800	0.01
FedNOVA	81.26	1.46	0.77	0.02
FedGP	**86.79**	**1.81**	**0.85**	**0.03**
FedAVG	82.15	2.41	0.78	0.04

Figure 9 shows the accuracy curves across aggregation rounds. FedGP demonstrates superior convergence behavior and maintains higher accuracy throughout the training process. FedAVG and FedPROX display comparable performance, albeit with more fluctuations, while FedNOVA lags slightly behind.

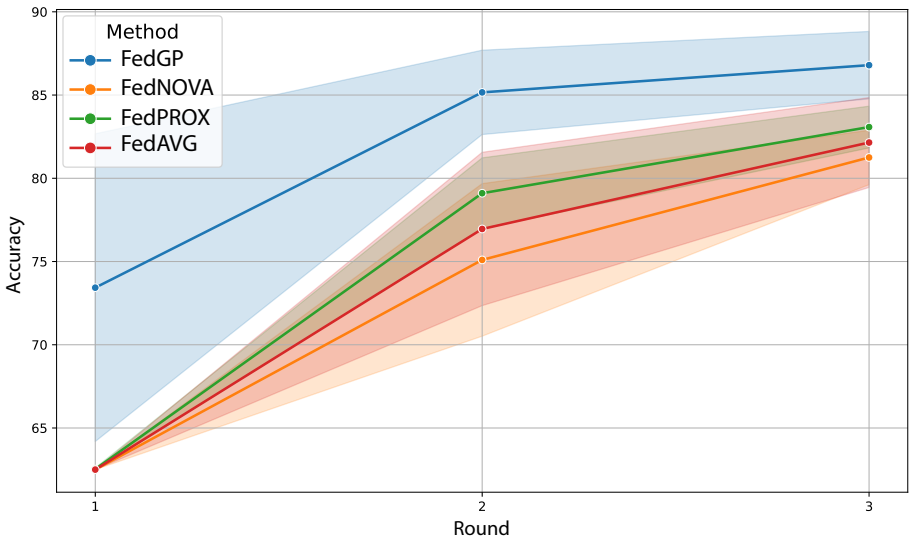

Fig. 9. Aggregation accuracy on the PneumoniaMNIST dataset with IID distribution.

The boxplot in Figure 10 further confirms the performance trends observed in the aggregated results. FedGP exhibits both a higher median and a narrower interquartile range compared to the other methods.

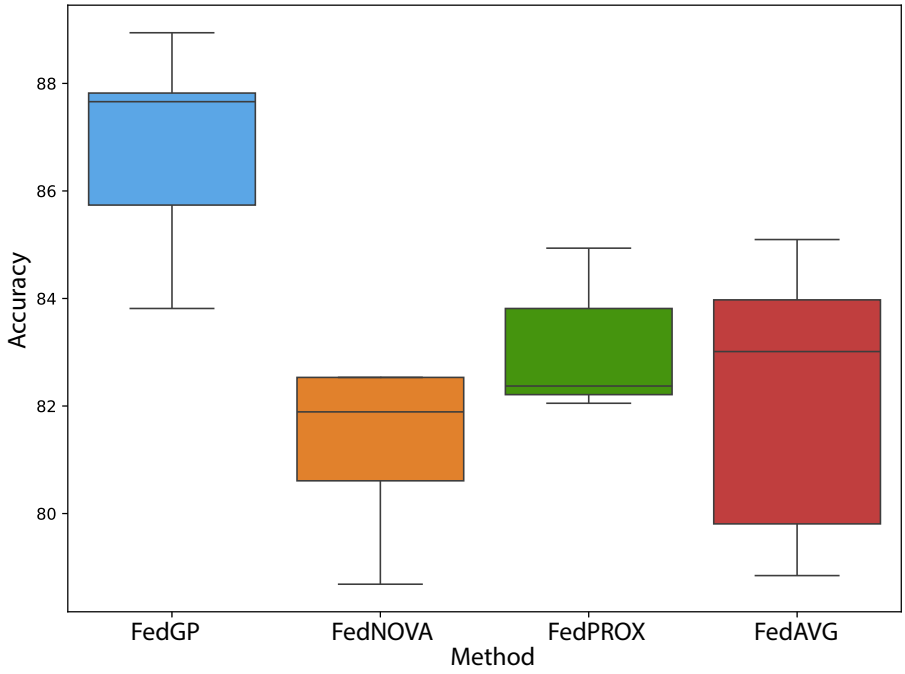

Fig. 10. Experiments accuracy on the PneumoniaMNIST dataset with IID distribution

To assess the statistical significance of the observed performance differences, we conducted pairwise *t-tests* on the accuracy values. The results are shown in Table 5. FedGP significantly outperforms all other methods ($p < 0.05$), confirming the robustness of the findings. No statistically significant differences were found between FedAVG, FedPROX, and FedNOVA.

Table 5. Pairwise statistical significance tests (t-test) on accuracy values for PneumoniaMNIST IID.

Method 1	Method 2	p-value
FedGP	FedNOVA	**0.001**
FedGP	FedPROX	**0.008**
FedGP	FedAVG	**0.015**
FedNOVA	FedPROX	0.083
FedNOVA	FedAVG	0.543
FedPROX	FedAVG	0.505

non-IID data. Table 6 summarizes the results on the PneumoniaMNIST dataset with non-IID distribution. In this more challenging scenario, the performance gap between FedGP and the other methods becomes even more pronounced. FedGP achieves an accuracy of 88.94% and an F1-score of 0.88, with very low standard deviations (1.28 and 0.01, respectively), indicating both high performance and strong stability across runs.

Table 6. Aggregated performance metrics for FedAVG, FedPROX, FedNOVA, FedGP. Dataset: PneumoniaMNIST, Distribution: non-IID.

Method	Acc	St.Dev	F1	St.Dev
FedPROX	74.71	8.86	0.64	0.18
FedNOVA	70.45	7.83	0.56	0.15
FedGP	**88.94**	**1.28**	**0.88**	**0.01**
FedAVG	75.67	7.86	0.66	0.15

In contrast, the other methods—FedAVG, FedPROX, and FedNOVA—suffer substantial drops in both accuracy and F1-score. Notably, FedNOVA exhibits the poorest performance, with an average F1-score of 0.560 and high variability. FedPROX and FedAVG show marginally better results, but remain far from the performance level of FedGP.

Figure 11 shows the accuracy progression over aggregation rounds. The performance degradation caused by non-IID data is clearly visible for all methods except FedGP, which maintains a consistently superior curve. Figure 12 confirms this trend: FedGP's box plot displays a narrow distribution and a significantly higher median, while all other methods show wider spread and lower central values.

The statistical analysis in Table 7 further supports these observations. FedGP significantly outperforms all other methods ($p < 0.05$). No significant differences were observed among FedAVG, FedPROX, and FedNOVA, suggesting similar performance under non-IID conditions.

Table 7. Pairwise statistical significance tests (t-test) on accuracy values for PneumoniaMNIST non-IID.

Method 1	Method 2	p-value
FedGP	FedNOVA	**0.0016**
FedGP	FedPROX	**0.0130**
FedGP	FedAVG	**0.0103**
FedNOVA	FedPROX	0.4915
FedNOVA	FedAVG	0.3738
FedPROX	FedAVG	0.8750

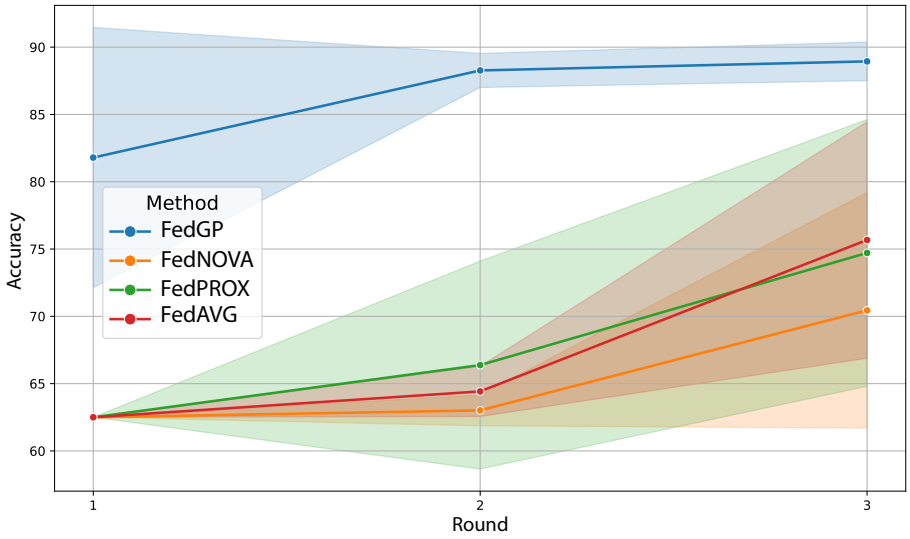

Fig. 11. Aggregation accuracy on the PneumoniaMNIST dataset with non-IID distribution.

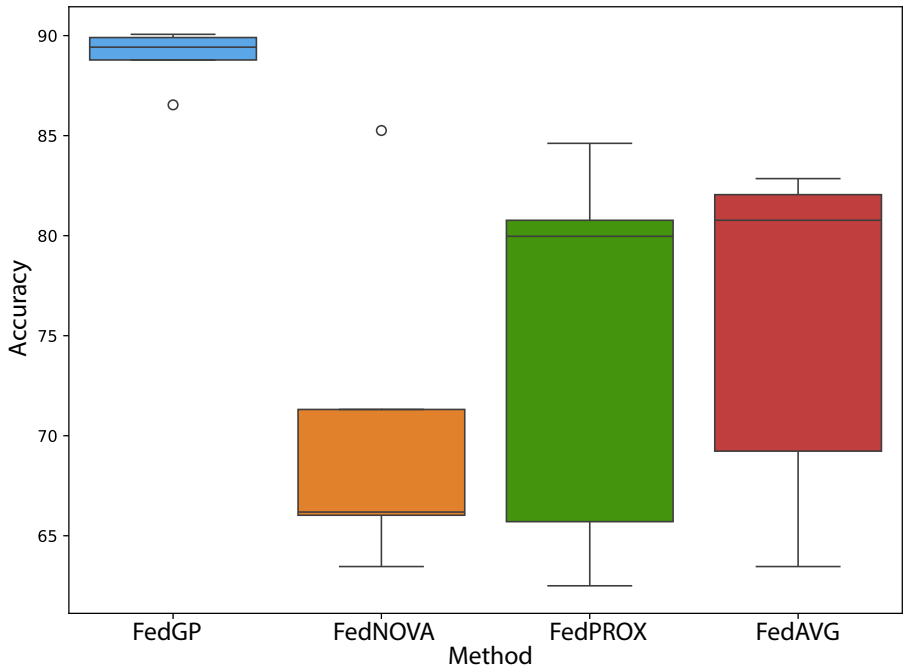

Fig. 12. Boxplot of experiments on the PneumoniaMNIST dataset with non-IID distribution

4.3 Study of execution time

Table 8 presents a summary of the average execution times, along with their relative standard deviations, for each aggregation method evaluated on two datasets: PathMNIST and PneumoniaMNIST. The timings are reported in seconds.

The FedPROX, FedNOVA, and FedAVG methods demonstrate comparable execution times across both datasets, with average durations ranging from several hundred seconds on PathMNIST (approximately 430–475 seconds) to around twenty seconds on PneumoniaMNIST (approximately 21–25 seconds). In contrast, FedGP is notable for its considerably higher computational cost, requiring 3335.63 seconds on PathMNIST and 445.28 seconds on PneumoniaMNIST. However, the standard deviations remain relatively low compared to the absolute times involved.

This distinct difference can be attributed to the inherent characteristics of the FedGP algorithm, which employs GP techniques to enhance the aggregation phase. Unlike other methods that typically perform simple weighted or normalized averaging, FedGP utilizes iterative processes, including selection, mutation, and crossover among models, necessitating repeated performance evaluations on subsets of data. These operations contribute to the algorithm's computational complexity, particularly when dealing with complex models or a large number of clients.

This computational burden poses a significant limitation to the method's broader adoption. However, it is essential to acknowledge that model aggregation represents a centralized phase of the FL process, typically executed on the server and is conducted infrequently compared to the local training phases.

Thus, if FedGP can deliver a substantial advantage in terms of accuracy and predictive performance, the associated overhead becomes justifiable and manageable in many contexts. This trade-off is particularly acceptable in high-value applications, such as in the medical field, where enhanced diagnostic accuracy can warrant longer computation times. Nonetheless, reducing the computational cost of FedGP remains a priority for future improvements.

Table 8. Average execution times with standard deviation. The results are shown in seconds.

Method	Dataset	Time (s)	St.Dev (s)
FedPROX	PathMNIST	474.92	11.95
FedNOVA	PathMNIST	432.28	12.45
FedGP	PathMNIST	3335.63	41.92
FedAVG	PathMNIST	440.86	10.90
FedPROX	PneumoniaMNIST	21.40	1.33
FedNOVA	PneumoniaMNIST	24.63	1.49
FedGP	PneumoniaMNIST	445.28	53.25
FedAVG	PneumoniaMNIST	23.70	1.35

5 Conclusions

This study conducted a rigorous empirical assessment of FedGP, a GP-based aggregation strategy for GL. FedGP was benchmarked against three canonical baselines (FedAVG, FedPROX, and FedNOVA) on the clinically relevant image corpora PathMNIST and PneumoniaMNIST, using both IID and deliberately skewed non-IID partitions. The evaluation assessed overall accuracy, F1-score, learning curve stability, distributional spread via box plots, and formal significance tests.

Findings.
IID setting. All four algorithms performed comparably, with FedGP attaining a modest but consistent edge in accuracy and F1-score; the gain reached statistical significance on PneumoniaMNIST but not on PathMNIST. *Non-IID setting.* The advantages of FedGP became prominent. Whereas the reference methods experienced sharp performance degradation and increased variance, FedGP preserved high accuracy and tight dispersion, yielding statistically significant improvements across all metrics and both datasets. No significant differences were detected among FedAVG, FedPROX, and FedNOVA.

Computational cost.
The evolutionary search that underpins FedGP introduces a non-trivial overhead. In centrally orchestrated FL, this overhead is tolerable because aggregation occurs episodically on the server and is not subject to real-time constraints. Nonetheless, reducing runtime remains a key objective. Promising research directions include warm-starting successive evolutionary phases with transfer learning, pruning inferior candidate programs early, and exploiting greater parallelism.

Implications and outlook.
FedGP emerges as a robust and generalizable aggregator for FL, particularly under the heterogeneous data distributions typical of real-world environments. Its fusion of evolutionary optimization with agent-based orchestration offers a fertile direction for building adaptive, explainable, and high-performance FL systems. Future work will concentrate on (i) accelerating FedGP's search process, (ii) extending the framework to asynchronous and fully decentralized topologies, and (iii) leveraging agents' autonomy to tailor models and to compress communication overheads.

Acknowledgments. This work was partially supported by the Validate-H (PInter 14-2025), the HES-SO RCSO ISNet Hybrid Ai foR RelIable perSonalized cOachiNg (HARRISON) grant (WP2), the Spanish Ministry of Economy and Competitiveness (PID2020-115570GB-C21, PID2023-147409NB-C22), funded by MCIN/AEI/10.13039/501100011033, and the Junta de Extremadura (GR24142).

References

1. Anuraj, B., Calvaresi, D., Aerts, J.M., Calbimonte, J.P.: Dynamic swarm orchestration and semantics in iot edge devices: A systematic literature review. IEEE Access **12**, 116917–116938 (2024). https://doi.org/10.1109/ACCESS.2024.3446876
2. Augusto, D.A., Barbosa, H.J.C.: Symbolic regression via genetic programming. In: Proceedings of the Sixth Brazilian Symposium on Neural Networks. vol. 1, pp. 173–178. Rio de Janeiro, Brazil (2000). https://doi.org/10.1109/SBRN.2000.889734
3. Beltrán, E.T.M., Pérez, M.Q., Sánchez, P.M.S., Bernal, S.L., Bovet, G., Pérez, M.G., Pérez, G.M., Celdrán, A.H.: Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Communications Surveys & Tutorials **25**(4), 2983–3013 (2023)
4. Benila, S., Devi, K.: Federated synergy: Hierarchical multi-agent learning for sustainable edge computing in iiot. IEEE Access **13**, 68311–68322 (2025). https://doi.org/10.1109/ACCESS.2025.3560781
5. Biswal, S., Elamvazhuthi, K., Berman, S.: Decentralized control of multiagent systems using local density feedback. IEEE Transactions on Automatic Control **67**(8), 3920–3932 (2022)
6. Bouacida, N., Hou, J., Zang, H., Liu, X.: Adaptive federated dropout: Improving communication efficiency and generalization for federated learning. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 1–6 (2021). https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484526
7. Brameier, M.F., Banzhaf, W.: Basic concepts of linear genetic programming. Springer (2007)
8. Cao, S., Zhang, H., Wen, T., Zhao, H., Zheng, Q., Zhang, W., Zheng, D.: Fedqmix: Communication-efficient federated learning via multi-agent reinforcement learning. High-Confidence Computing **4**(2), 100179 (2024)
9. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning: A meta-learning approach. arXiv preprint arXiv:2002.07948 (2020)
10. Granqvist, F., Seigel, M., Van Dalen, R., Cahill, A., Shum, S., Paulik, M.: Improving on-device speaker verification using federated learning with privacy. arXiv preprint arXiv:2008.02651 (2020)
11. Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2019)
12. Hsu, C.M.: A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming. Expert Systems with Applications **38**(11), 14026–14036 (2011). https://doi.org/https://doi.org/10.1016/j.eswa.2011.04.210, https://www.sciencedirect.com/science/article/pii/S0957417411007378
13. Ivoghlian, A., Salcic, Z., Wang, K.I.K.: Adaptive wireless network management with multi-agent reinforcement learning. Sensors **22**(3) (2022). https://doi.org/10.3390/s22031019
14. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and trends® in machine learning **14**(1–2), 1–210 (2021)
15. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency (2017)

16. Kontar, R., Shi, N., Yue, X., Chung, S., Byon, E., Chowdhury, M., Jin, J., Kontar, W., Masoud, N., Nouiehed, M., Okwudire, C.E., Raskutti, G., Saigal, R., Singh, K., Ye, Z.S.: The internet of federated things (ioft). IEEE Access **9**, 156071–156113 (2021). https://doi.org/10.1109/ACCESS.2021.3127448
17. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press (1992), http://mitpress.mit.edu/books/genetic-programming
18. Langdon, W.B., Modat, M., Petke, J., Harman, M.: Improving 3d medical image registration cuda software with genetic programming. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. p. 951–958. GECCO '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2576768.2598244, https://doi.org/10.1145/2576768.2598244
19. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Dhillon, I., Papailiopoulos, D., Sze, V. (eds.) Proceedings of Machine Learning and Systems. vol. 2, pp. 429–450 (2020)
20. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. arXiv preprint (2020), https://arxiv.org/abs/1907.02189
21. Liu, Y., Ai, Z., Sun, S., Zhang, S., Liu, Z., Yu, H.: Fedcoin: A peer-to-peer payment system for federated learning. In: Federated learning: privacy and incentive. pp. 125–138. Springer (2020)
22. Long, G., Tan, Y., Jiang, J., Zhang, C.: Federated Learning for Open Banking, pp. 240–254. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-63076-8_17
23. Machado, P., Martins, T., Correia, J.a., Santo, L.E., Lourenço, N., Cunha, J.a., Rebelo, S., Martins, P., Bicker, J.a.: Designing coins with evolutionary computation. SIGEVOlution **17**(2) (Sep 2024). https://doi.org/10.1145/3695933.3695934, https://doi.org/10.1145/3695933.3695934
24. Madi, A., Stan, O., Mayoue, A., Grivet-Sébert, A., Gouy-Pailler, C., Sirdey, R.: A secure federated learning framework using homomorphic encryption and verifiable computing. In: 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS). pp. 1–8. IEEE (2021)
25. Mamond, A.W., Kundroo, M., Yoo, S.e., Kim, S., Kim, T.: Fldqn: Cooperative multi-agent federated reinforcement learning for solving travel time minimization problems in dynamic environments using sumo simulation. Sensors **25**(3) (2025). https://doi.org/10.3390/s25030911
26. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
27. Miller, J.F., Thomson, P.: Cartesian genetic programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) Genetic Programming. pp. 121–132. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
28. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) Parallel Problem Solving from Nature - PPSN XII. pp. 21–31. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
29. Mugunthan, V., Polychroniadou, A., Byrd, D., Balch, T.H.: Smpai: Secure multi-party computation for federated learning. In: Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services. vol. 21. MIT Press Cambridge, MA, USA (2019)

30. Muntean, M.V.: Multi-agent system for intelligent urban traffic management using wireless sensor networks data. Sensors **22**(1) (2022). https://doi.org/10.3390/s22010208
31. Nie, W., Yu, L., Jia, Z.: Research on aggregation strategy of federated learning parameters under non-independent and identically distributed conditions. In: 2022 4th International Conference on Applied Machine Learning (ICAML). pp. 41–48. IEEE (2022). https://doi.org/10.1109/ICAML57167.2022.00016
32. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Transactions on Evolutionary Computation **5**(4), 349–358 (2001). https://doi.org/10.1109/4235.942529
33. Pacioni, E., Fernández De Vega, F., Calvaresi, C.: Towards a meaningful communication and model aggregation in federated learning via genetic programming. In: In Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025). vol. 3, pp. 1427–1431 (2025). https://doi.org/10.5220/0013380400003890
34. Pacioni, E., Fernández De Vega, F., Calvaresi, D.: Fedgp: Genetic programming for evolutionary aggregation in federated learning with non-iid data. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 419–434. Springer (2025)
35. Petke, J., Haraldsson, S.O., Harman, M., Langdon, W.B., White, D.R., Woodward, J.R.: Genetic improvement of software: A comprehensive survey. IEEE Transactions on Evolutionary Computation **22**(3), 415–432 (2018). https://doi.org/10.1109/TEVC.2017.2693219
36. Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., Piccialli, F.: Model aggregation techniques in federated learning: A comprehensive survey. Future Generation Computer Systems **150**, 272–293 (2024). https://doi.org/10.1016/j.future.2023.09.008
37. Reguieg, H., El Hanjri, M., El Kamili, M., Kobbane, A.: A comparative evaluation of fedavg and per-fedavg algorithms for dirichlet distributed heterogeneous data. In: 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM). pp. 1–6. IEEE (2023). https://doi.org/10.1109/WINCOM59760.2023.10322899
38. Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K., et al.: The future of digital health with federated learning. NPJ digital medicine **3**(1), 119 (2020)
39. Marin Machado de Souza, R., Holm, A., Biczyk, M., de Castro, L.N.: A systematic literature review on the use of federated learning and bioinspired computing. Electronics **13**(16), 3157 (2024)
40. Spector, L., Goodman, E., Wu, A., Langdon, W., Voigt, H., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M., Burke, E., Publishers, M.: Autoconstructive evolution: Push, pushgp, and pushpop. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001) (05 2001)
41. du Terrail, J.O., Léopold, A., Joly, C., Beguier, C., Andreux, M., Maussion, C., Schmauch, B., Tramel, E.W., Bendjebbar, E., Zaslavskiy, M., Wainrib, G., Milder, M., Gervasoni, J., Guérin, J., Durand, T., Livartowski, A., Moutet, K., Gautier, C., Djafar, I., Moisson, A.L., Marini, C., Galtier, M., Bataillon, G., Heudel, P.E.: Collaborative federated learning behind hospitals' firewalls for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer. medRxiv (2021). https://doi.org/10.1101/2021.10.27.21264834, https://www.medrxiv.org/content/early/2021/10/28/2021.10.27.21264834

42. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, vol. 33, pp. 7611–7623. Curran Associates Inc., Red Hook, NY, USA (2020)
43. Wang, L., Li, J., Chen, W., Wu, Q., Ding, M.: Communication-efficient model aggregation with layer divergence feedback in federated learning. IEEE Communications Letters **28**(10), 2293–2297 (2024). https://doi.org/10.1109/LCOMM.2024.3454632
44. Wilson, D.G., Cussat-Blanc, S., Luga, H., Miller, J.F.: Evolving simple programs for playing atari games. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 229–236. GECCO '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3205455.3205578, https://doi.org/10.1145/3205455.3205578
45. Xu, C., Qu, Y., Xiang, Y., Gao, L.: Asynchronous federated learning on heterogeneous devices: A survey. Computer Science Review **50**, 100595 (2023)
46. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)
47. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) **10**(2), 1–19 (2019)
48. Zehtabi, S., Hosseinalipour, S., Brinton, C.G.: Decentralized event-triggered federated learning with heterogeneous communication thresholds. In: 2022 IEEE 61st Conference on Decision and Control (CDC). pp. 4680–4687. IEEE (2022)
49. Zeng, Y., Mu, Y., Yuan, J., Teng, S., Zhang, J., Wan, J., Ren, Y., Zhang, Y.: Adaptive federated learning with non-iid data. The Computer Journal **66**(11), 2758–2772 (09 2022). https://doi.org/10.1093/comjnl/bxac118
50. Zhang, S.Q., Lin, J., Zhang, Q.: A multi-agent reinforcement learning approach for efficient client selection in federated learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 9091–9099 (2022)
51. Zhang, W., Zhou, T., Lu, Q., Yuan, Y., Tolba, A., Said, W.: Fedsl: A communication-efficient federated learning with split layer aggregation. IEEE Internet of Things Journal **11**(9), 15587–15601 (2024). https://doi.org/10.1109/JIOT.2024.3350241
52. Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-iid data: A survey. Neurocomputing **465**, 371–390 (2021). https://doi.org/https://doi.org/10.1016/j.neucom.2021.07.098