Discovering Communities With Clustered Federated Learning

Mickaël Bettinelli, Alexandre Benoit, Kévin Grandjean

Discovering Communities With Clustered Federated Learning

Mickaël Bettinelli
Univ. Savoie Mont Blanc
LISTIC
Annecy, France
mickael.bettinelli@univ-smb.fr

Alexandre Benoit
Univ. Savoie Mont Blanc
LISTIC
Annecy, France
alexandre.benoit@univ-smb.fr

Kévin Grandjean
Univ. Savoie Mont Blanc
LISTIC
Annecy, France
kevin.grandjean@etu.univ-smb.fr

***Abstract*—We address a community detection problem in a realistic federated learning setup where clients own non-iid data. We propose a Clustered Federated Learning-based method (CFL) that can dynamically discover client communities according to their model distances along the federated rounds. This method is based on Louvain clustering, a relevant model similarity measure and a client aggregated model attribution strategy. The proposed framework enables the unsupervised detection of communities with no prior knowledge while maximizing client task performances. We propose an extensive study based on the Cifar10 dataset to assess the sensitivity of the approach to critical factors including data non-iidness level, model initialization, client participation rates and client cluster attribution strategy. Importantly, both model task accuracy and clustering relevance are evaluated thus extending state-of-the-art standard evaluation. Compared to state-of-the-art on an image classification problem, we show the interest of our continuous clustering and attribution strategy along federated rounds that maintain client migration capability while preserving the aggregated model relevance. This facilitates learning convergence while reducing result variability along trials. This work can be flawlessly integrated in standard FL approaches and opens new directions for both task performance and community detection relevance in a federated learning context. Results show the relevance of the clustering on an image classification task to discover communities of related classes.**

***Index Terms*—Federated Learning, Community Detection, Clustered Federated Learning.**

I. Introduction

Federated Learning (FL) has recently emerged as a research direction for enhanced privacy and secure decentralized machine learning approach [18]. However, heterogeneity and diversity of data across multiple clients remains a challenging but realistic issue that limits the relevance of a unique federated model [11], [16]. Recently model personalization [23] and more specifically Clustered Federated Learning (CFL) [5] has been proposed to address such problem by grouping clients according to a similarity criteria therefore producing intermediate models between local models and the global model. Promising results have been shown, improving task performance for each client, taking advantage of both the aggregation of similar local models and a global knowledge. However, to the best of our knowledge, such approaches are task performance oriented and do not explore the community detection potential of CFL. In addition, several questions remain on the model similarity measure relevance, the clustering strategy and the result variability with respect to common basic but critical factors such as the client's participation rates, model initialization and so on. In this work, we explore the community detection potential of CFL while studying both model task and clustering convergence behaviours with respect to some important factors encountered in real life application. Finding these communities during the training phase opens to new methods relevant for model optimization in a variety of realistic scenarios including non-iid data and also learning on data streams by optimizing community models that gather clients with similar data distributions. In this work, we explore this direction, continuously detecting client communities along training, and integrating it into a clustered federated learning approach. Our generic approach is evaluated on a controlled dataset allowing for the evaluation of clustering relevance while providing insights on results variability related to sensitive method choices and initialization. The proposal relies on a community models attribution method to each client that can be associated with complementary state-of-the-art methods such as client sampling [7] and model aggregation methods [25] that aim at mitigating bias. Our contributions are summarized as follows:

1) A regular client clustering enabling for community detection on the server side with no prior on client behaviours.
2) Community dedicated aggregated models generation added to a global model.
3) A community model client attribution strategy.
4) An evaluation of the approach on a controlled dataset to quantify both task performance and clusters quality.

This article is organized as follows: the first section explores similar CFL methods of the state-of-the-art. The second section describes our problem. The third section presents our contributions. The fourth section evaluates the sensitivity of our approach to critical factors such that the data non-iidness. It then compares several attribution strategies according to several hyperparameters. The last section concludes this work.

Identify applicable funding agency here. If none, delete this.

II. Related work

The initial proposal of Federated Learning [18] assumes that a single model can fit all clients, even if their data distribution is non-iid. But averaging client models that have very different distributions might lead to a poor optimization of the global model. In addition, as for realistic scenarios, clients and their data evolve in time thus creating new bias that has to be mitigated. Federated Learning is indeed very sensitive to the data distribution [11]. Bias mitigation is thus an active research direction for instance relying on model aggregation optimization [4] and appropriate data sampling strategies as presented in [7], [17]. Following this direction, Personalized Federated Learning [23] have recently been proposed to provide each client with relevant shared models. Among the diversity of the approaches, Clustered Federated Learning (CFL) based ones isolate groups of clients to mitigate bias and to provide relevant models to similar client subsets. Such approaches rely on a variety of similarity measures as well as optimization objectives. This section presents several recent approaches of the literature.

[7] consider client clustering as a solution to mitigate bias. Different from the other work presented hereafter, this work considers client sampling clustering in order to reduce communication costs with the server while increasing clients' representatives and reducing variance of the client's weight aggregation. Hierarchical clustering is applied on the gradient between each client and the global model.

Sattler et *al.* introduce *Clustered Federated Learning* [21] as a new federated multitask learning method. It improves performance by grouping clients into clusters with jointly trainable data distributions, achieving greater or equal performance than conventional federated learning under privacy constraints. In this article, clients are bipartionned after each round by comparing the cosine similarity of their gradient updates. Clients are thus grouped with respect to their convergence directions but the proposed method remains task performance guided and does not study the resulting partitioning.

The IFCA algorithm [8] is a Clustered Federated Learning method that aims to make FL more efficient when clients own non-iid data. Contrary to the CFL literature where clustering is more likely to be made on the server side, Ghosh et *al.* argue that making clusters on the client's side reduces the computational cost of the server. IFCA clients therefore identify their cluster by themselves by selecting the model that minimizes their loss. It implies an increase of computational cost on the clients that have to test several models before identifying their cluster. Finally, IFCA expects the number of clusters to be known in advance which limits its applicability in real scenarios. In addition, the communication cost is increased as clients must receive a set of models from the server instead of a single one.

FlexCFL [5] is a CFL technique that aims at grouping clients by their gradient updates similarities. It implements a specific mechanism for allocating a model to new clients for a better scalability. This approach is restricted to supervised learning since client's migration from one cluster to another is triggered when their data label distribution evolves above a threshold. Such policy can be challenging to calibrate and may imply the communication of the client label distributions. FlexCFL uses a similarity metric called *Euclidean Distance of Decomposed Cosine Similarity* which decomposes the updates into m direction using *Singular Value Decomposition* then compute a cosine similarity. The main advantage of this metric is to avoid the concentration phenomenon coming from the model parameter dimensions. In addition, they also propose a method to cluster clients according to their gradient based on EDC that can handle a change in data. FlexCFL is compared to IFCA and shows a better optimization of the global model on several datasets (MNIST, FashionMNIST, FEMNIST) and models (CNN, MLP, MCLR). As for IFCA, the number of clusters obtained with FlexCFL must be known in advance while clustering quality is not studied.

Briggs et *al.* propose hierarchical clustering to group clients relying on the similarity of their local updates [3]. While previous methods cluster clients after each round, this method first train models without any clustering step for n communication rounds then use the hierarchical clustering. After clusters are separated, they are trained separately until the end of the training. Clustering after n rounds allows clients to converge to a global shared solution before being clustered but may produce more similar clusters and introduce a bias related to the global model. Authors show that in a non-iid setting, beginning clustering in the first rounds of communication is preferable. Unlike IFCA or FlexCFL, this method does not expect the number of clusters to be known, thus providing more flexible. However, datasets are expected to remain static. Similarly, Espinoza Castellon et *al.* [6] propose a client partitioning strategy that is applied a single time at a late round along a regular federated learning session. Relying on model cosine similarity clustering, this allows clients to be grouped after the global model has converged and let them specialize in the remaining rounds while still sharing their knowledge their similar neighbours. The choice of the clustering round remains delicate while the bias introduced by the global model must be mitigated in the last rounds.

Works presented in this section are focusing on the optimization of a single or a set of multiple shared models relying on a variety of strategies. None of them actually assess the quality of their clustering. Sattler et *al.* note that CFL exposes a new privacy issue as it seems possible to infer information about clients from their models at each round [21]. In this work, we push further this idea and aim at both maximizing task performance and discovering communities of clients from their models during the Federated Learning process.

III. Problem statement

Searching for a general CFL approach with minimum prior knowledge on the data and client behaviours, we consider a configuration with standard methods for model aggregation with no private information communication between clients

and the server apart from the local model and no communication across clients. This thus build a baseline that can be further improved with refinements related to aggregation, client sampling and other complementary approaches discussed in the previous section. We then evaluate the relevance of the proposed strategies with respect to their own configuration, model initialization, as well as client data distribution and participation rates.

A. General configuration

We focus on non-convex optimization problems addressed with neural network and build upon the centralized federated learning as defined by [18]. We assume that a set K of clients participate to the optimization of the same model relying on the same optimization criteria but different training data distributions. Clients are connected to a single central server that receives and aggregates client models at each communication round. Regarding client selection, all or a random subset $Q_t \subset K$ with $Q_t \neq \emptyset$ of the clients participate to a given communication round t and provide the server with their updated local models.

Community models W^{C_i} are computed as the average of the local model weights w^k of a given subset of clients C_i, with $C_i \subset K$ and $C_i \neq \emptyset$ defined in eq. 1.

$$W_{t+1}^{C_i} = \frac{1}{|C_i|} \sum_{k \in C_i} w_t^k \tag{1}$$

With such base aggregation, at a given round, each client has the same contribution to their community model W^{C^i} and the server does not need to know about client dataset behaviours and consider them equally. Regarding client sampling for each round, we consider the standard uniform random sampling approach but other refined strategies such as [7] can be used flawlessly. Finally, data distribution of each client remains constant along a given experiment but can vary from an experiment to another (*Cf.* evaluation section V).

B. Server side priors

With no prior on the client communities and their data behaviours, the objective of the server is to both help clients maximize their task performance and detect their communities. Finally, clustering should not introduce bias and reduce clients task performance by locking them into a given cluster. This constraint also opens further perspectives related to federated learning on data streams and time evolving data [17].

IV. Communities detection

Community detection is a fundamental task aiming at identifying groups of entities that are densely connected within themselves but sparsely connected to other groups [9]. We consider here federated clients as entities and search for their partitioning with respect to their model similarities thus considering such metric as edges in the client graph.

We then propose a client clustering and community model computation process applied on the server side all along the optimization process in the aim to avoid case study dependent clustering hyperparameter search as for [3], [6]. Such strategy also allows the monitoring of client community attributions along the federated optimization process, anticipating applications related to security such as poisoning attacks detection [6].

In this section, we present our client community detection approach that involves the choice of a similarity metric, a scalable client partitioning and the synthesis of community models as well as a global model.

A. Similarity metric

The common methods to evaluate the similarity between models are representational and functional similarity metrics [12], which respectively compare each neuron activation and output. The most common used are CCA and CKA [13], two representational metrics. However, reference data inputs are required to generate neuron responses, relying on one common set of carefully selected and unbiased samples to compare multiple models. Following the server side priors detailed in the previous section and the general difficulty to collect relevant, unbiased and privacy preserving centralized data, we do not consider such approach. This leaves the option of using distance metrics instead, that will assess the distance of models according to their parameters. Such metrics are low computation cost and the most typical ones reviewed in [12] are L-norms, cosine distance and Procrustes disparity, that can be aggregated all along deep models. In addition, the *deep relative trust* or *trusted distance* introduced with the *Fromage* optimizer [1] expresses the functional distance between models with similar structures. Its upper bound follows equation 2 with L the number of model layers, $w_{a,l}$ and $w_{b,l}$ the parameters of models a and b at layer l.

$$trusted(a,b) = \prod_{l=1}^{L} \left(1 + \frac{||w_{a,l} - w_{b,l}||_F}{||w_{a,l}||_F}\right) - 1 \tag{2}$$

Along a preliminary study, we compared such metrics and consider the trusted distance as the most relevant, with stable results, but this choice is not critical in this study as other metrics such as cosine distance classically used in CFL approaches yield similar results despite being less selective. Distance values are next mapped to the range [1,0] before any other transformation allowing for distance comparison and in order to provide normalized values to the following processes. Distance values are normalized and transformed in similarity values according to eq. 3:

$$f(v) = 1 - \frac{v - min}{max - min} \tag{3}$$

v being the value to normalize and min and max respectively the lowest and highest distance across all the client model pairs at a given round.

Normalized chosen distance measures are then converted to similarity values in order to comply with the following client graph clustering approach. Three main approaches are possible, their choice impacting on cluster composition :

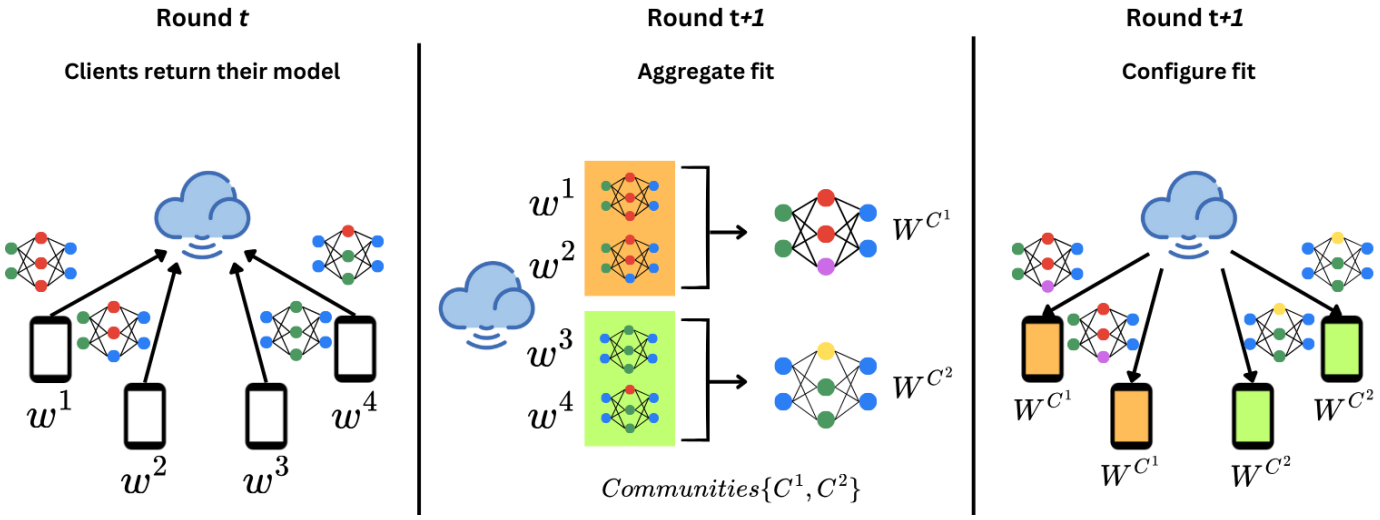


Fig. 1: Overview of our CFL community detection approach over a round. Firstly, clients participating in the previous round send their trained model to the server. Then the server runs the aggregate fit function described in the algorithm 1 and detects client communities. Finally, the server executes the function configure fit from the algorithm 2 to sample and initialize the new set of participating clients. In this example, returned models in configure fit are the nearest community model of each client. It therefore matches the 1NN algorithm.

"exaggerating" small distances (e.g. $x \rightarrow x^{1/2}$), exaggerating great distances (e.g. $x \rightarrow x^2$), or linearly transforming the distance matrix. Exaggerating great distance by the following transformation $x \rightarrow x^3$ has been chosen to facilitate the differentiation between low and high similarities. Note that the higher the exponent value is, the lower similarity values, possibly making low to high similarities more difficult to distinguish.

B. Client partitioning method

The identification of client groups can be considered as a clustering problem when considering the full client adjacency matrix by computing similarities between each client pairs. It can also be formulated as a community detection problem if only a subset of the client distances is considered. Full adjacency matrix can be a computational bottleneck and can be difficult to obtain in the case study of a cross-device federated learning case study with a large number of clients but remains acceptable in case of a cross-silo FL setup with fewer clients. Also, the partitioning algorithm introduces an additional cost. Then the choice of such algorithm can be guided by the expected clustering quality, computational cost requirements as well as scaling capabilities, and hyperparameter search cost. Several methods are proposed in the literature for clients clustering [19]. CFL state-of-the-art papers usually rely on hierarchical clustering as for [3], [5]. In this work and as for [6], we consider the Louvain community detection (e.g. Louvain [2], Leiden [24]) that is appropriate with a large range of client numbers while introducing few hyperparameter search. Its main hyperparameter so-called "resolution", controls the size of the clusters and thus the communities fragmentation. Small resolutions leads to numerous small clusters, while high resolutions leads to few big clusters. In the following, the partitioning at a given round yields P, the list of client communities. Each community i presenting a community model noted W^{C_i} computed from eq. 1.

C. Clients' model attribution

Finally, considering the overall CFL process, a global strategy for clients partitioning and community cluster attribution to each client must be defined. Following our problem statement, minimum knowledge on client behaviours being available, the server is almost blind and should apply a careful clustering that do not bias clients by locking them into a non-relevant cluster. We then propose a flexible approach inspired from [5] that applies client partitioning regularly along the optimization process. However our approach differentiates by applying clustering after each round instead of introducing a tuned trigger that relies on a label shift measure. We also compare a variety of model attribution strategies that would allow any client to move to the most relevant community model at any time along the process, their choice impacting on task performance and clustering quality. We consider the 3 following aggregated model attribution strategies applied after the client sampling step of each federated round t to provide each participating client q with a personalized aggregated model w^q_{t+1}:

1) AVG: participating clients receive the global model W_t that is the unweighted average of the set of community models $W_t^{C_i}$. Each community thus contributes with the same weight to build the global model. This approach resembles the standard FedAVG but does not weight client nor community contributions thus being naively more ethical.

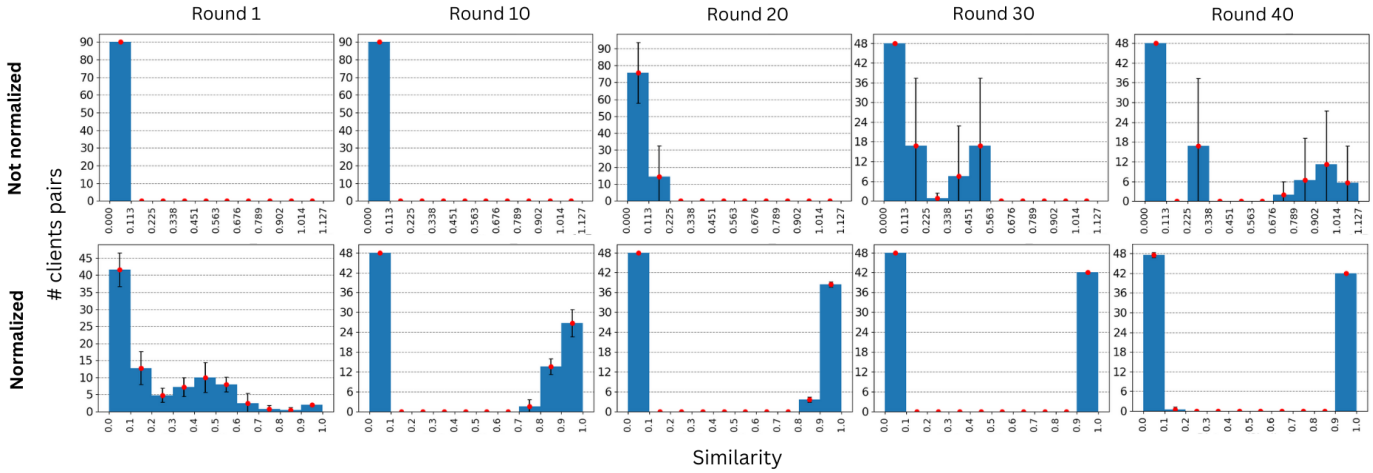


Fig. 2: Client similarity distributions over 2 FL optimization trials without (top row) or with (bottom row) preliminary distance normalization (eq.3) The histograms are averaged from 4 experiments, respectively for the top and bottom row. For visualization and as for the applied process, top row shows the similarity values calculated from the raw distance measure v with $f(v) = (max - v)^3$, max being the greatest distance value between all client models at a given round. Normalized distance enables faster convergence to the expected cluster separation along the FL process.

2) 1NN: relying on the same model distance, here the trusted distance, a given client receives its nearest community cluster. Note that in case of a new unknown client, the global model W_t is attributed as an initialization.
3) WNN: an intermediate approach, the semi-soft assignment strategy [15] shown in eq. 4 that computes an average of neighbouring community cluster. It introduces more flexibility with the β parameter that modulates the influence of the neighbours and an early cut-off effect brought by the selection of the nearest neighbours that reduce the influence of long-range clusters. For each selected client q, their k-nearest community models, N_q, are considered and their individual distance to each client q is used as a weighting factor $u_{q,j}$ to compute the personalized aggregated model $wnn(q, N_q)$ at round t (we do not explicitly mention t in the following formulation). We consider $\beta = 1.0$ as the default configuration. Again, in case of a client newcomer, the global model W_t is sent instead.

We then present the aggregation and the model attribution algorithms in respectively alg. 1 and alg. 2 using the same notations. Those two algorithms are called sequentially as for the classical Federated Learning process. Fig. IV illustrates the workflow of the proposed contribution over a single round, this process being repeated along training rounds.

$$u_{q,j} = \frac{exp\left(-\beta trusted(w_q, W^{C_j})\right)}{\sum_{l \in N_q} exp\left(-\beta trusted(w_q, W^{C_l})\right)}$$
$$wnn(q, N_q) = \frac{1}{|N_q|} \sum_{j \in N_q} u_{q,j} W^{C_j} \quad (4)$$

Algorithm 1 Model aggregation step for a FL round t

procedure AGREGATE($clientUpdates$)
Require: $clients$ the graph of know clients, may be empty
Ensure: $clients$, P, W_t and each W^{C_i} to be updated
  $clients \leftarrow updateClientGraph(clientUpdates)$
  $P \leftarrow applyLouvainClustering(clients)$
  for each $C_i \in P$ **do**
    $W_t^{C_i} \leftarrow \frac{1}{|C_i|} \sum_{k \in C_i} w_t^k$ ▷ Equation 1
  end for
  $W_t \leftarrow \frac{1}{|P|} \sum_{i \in P} W_t^{C_i}$ ▷ Same as equation 1
end procedure

V. Evaluation

A. Experimental setup

In this section, we present an approach to evaluate our contribution with comparable state-of-the-art methods. We search for a relevant case study where realistic communities can coexist without exaggeration of their behaviours that could be easily spotted by high capacity models. We thus consider the classification problem related to the standard Cifar10 dataset [14] with preserved image orientations and colour distributions. Further, relying on the underlying label ontology of the dataset, we apply a client sampling policy on the training dataset to create communities based on semantic label distribution shifts and verify the capability of CFL approaches to detect them while maximizing classification performance.

Cifar10 is a challenging toy dataset that allows for relevant evaluation of community detection along with a difficult target classification problem. It gathers 10 classes: *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship* and *truck*. We assume there are two super classes representing higher levels of the dataset labels ontology: *animals* and *vehicles*.

Algorithm 2 Client sampling and model attribution

procedure CONFIGURE FIT(K, P, W_t)
Require: a client sampling method $SampleClients$, kNN relies on the trusted distance.
Ensure: a subset $Q \subset K$ start a round with an appropriate aggregated model.
    $Q \leftarrow$SampleClients(K)
    for each client $q \in Q$ **do**
        if AVG **or** q is newcomer **then**
            $w_{t+1}^q \leftarrow W_t$
        else
            if 1NN **then**
                $w_{t+1}^q \leftarrow$ kNN(q, P, k=1)
            else if WNN **then**
                nearest clusters $N_q \leftarrow$ kNN(q, P, k=3)
                $w_{t+1}^q \leftarrow$ wnn(q, N_q) ▷ Equation 4
            end if
        end if
        FitRound(q, w_{t+1}^q)
            ▷ client then fits starting with w_{t+1}^q
    end for
end procedure

The goal of the server is to detect these two communities while clients are training with different classes distributions. The model used for the experiments is MobileNet [10] which is a high-capacity model but remains frugal with modest performances on Cifar10 compared to newer models. This choice is, however, relevant to conduct all the proposed experiments at a reasonable cost, totalling 8000 CPU hours, not counting calibration trials, on Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz and 40Go of RAM, using the Tensorflow 2.14.1 and Flower 1.9.0 libraries within our open-source framework.

Models task performance and cluster quality are assessed in the following section on the official Cifar10 validation set, a collection of balanced 10k sample, through three metrics: model *accuracy*, the *Adjusted Rand index* (ARI) and the *Silhouette score*. Adjusted Rand index produces a score that assesses a clustering compared to an expected partitioning [22] while the Silhouette score evaluates how much clients are close to their cluster and how far they are from other clusters [20]. Each experiment is run on 40 rounds, with 10 clients and averaged on 4 runs with different initialization seeds. Each client has a majority class, the quantity of which can varies between experiments.

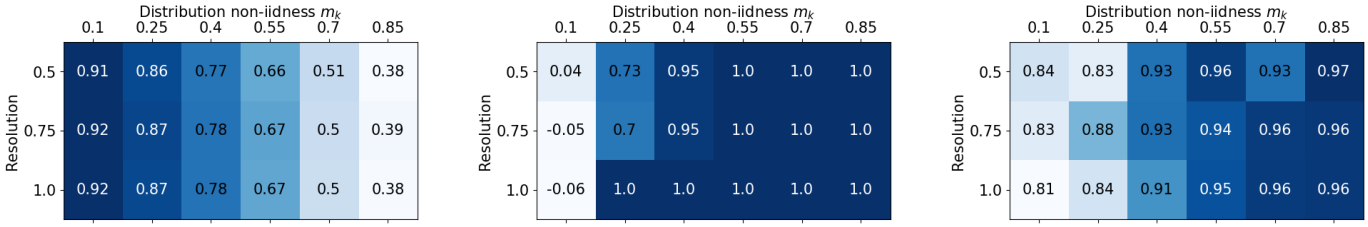
B. Results

1) Louvain resolution and data distribution: This first experiment plan compares the aforementioned metrics according to varying data distributions and Louvain resolutions. The aim is to show what data distribution and what resolution are the most profitable to the community detection. We then build an exaggerated imbalanced data context based on the Cifar10 dataset where 10 client train in an FL session while each has a specific majority class in its local training set. With the 10 class labels indexes in $L = \{l_0, l_1, ..., l_9\}$, we thus create a context where a given client k has a majority class l_k such that $P_k(l_k) = m_k$ while any other label $l_o \in L$ with $l_o \neq l_k$ has balanced ratio with the others such that $P_k(l_o) = (1.0 - m_k)/9$. By setting $m_k = 1.0$, each client therefore owns a single class that is different from other clients. It is the most non-iid case in this configuration. In this experiment, all the clients participate to each round.

Fig. 3a presents the classification accuracy measures with varying degrees of non-iideness controlled by m_k and the Louvain resolution on the balanced validation set. Accuracy is strongly impacted by the non-iidness but not by the Louvain resolution. But unlike accuracy, ARI is much higher when training on non-iid data, as shown in Fig. 3b. The clusters are also more dense and better separated when client data is non-iid and the resolution is higher as shown in Fig. 3c. Results show that the best hyperparameters in order to detect communities is Louvain resolution at 1.0 and a data distribution parameter $m_k = 1.0$. A higher Louvain resolution allows to make fewer but bigger clusters, which is very effective to detect only two communities as for our specific evaluation conditions. Also, $m_k = 0.7$ is a compromise between a better model optimization and a worse Silhouette score. In conclusion, first, when data is more balanced across clients, local models are more similar and community detection is harder, this could be expected. Also, a sticking point is the fact that model clustering becomes easier when clients are biased by a specific majority class. Their clustering finally matches the one of the animals and vehicles superclasses. Then our CFL approach is able to differentiate the high semantic level categories even if these are not explicitly shown along training and despite the strong variability in the data even at the class level. This is a strong result to be validated in over context in further studies. Coming back to the data, this means that when clients are strongly biased by a given class i.e. when m_k is high, then, the local models of a same superclass converge to a similar solution despite the fact that objects background is very diverse and resembling across superclasses. We actually reach subtle results that complete previous work results such as IFCA [8] that showed the capability of CFL to differentiate clusters of rotated and original images. This also confirms the potential privacy issue discussed in [21] since the server knows the community model a given client belongs to.

In addition, we studied the impact of model distance normalization before similarities computation and the effect on the community's separability. Fig. 2 shows the evolution of similarity distributions along the federated rounds comparing two trials with either not normalized or normalized distances. One observes, with normalized distances, the faster convergence of the distributions to the 2 expected modes that illustrate high intra-cluster and low inter-cluster distances.

2) Model attribution and selection rate: The relevance of each proposed model attribution method described in section IV-C is compared with a varying ratio of participating clients. We consider a non-iid training data distribution for each client with $m_k = 0.7$. Client selection rates remain constant along



(a) Accuracy (b) Adjusted Rand Index (c) Silhouette score

Fig. 3: Metrics for 1NN attribution model strategy according to Louvain resolution and data distribution.

runs for each experiment. Louvain resolution is set to 1.0 for each method. We add to the comparison the results provided by the IFCA state-of-the-art paper [8].

IFCA is a CFL method tested on rotated Cifar10 images to improve the global model. The original paper reports a successful partitioning of images in two clusters: one for regular images and one for rotated images. However, clustering images according to their semantic superclass is a completely different task that can be interesting to test with such approach. In our experiments, the IFCA method uses the exact same dataset described in previous sections. Then IFCA is not compared in the same conditions as the original paper since the clients' data distribution is solely different. In these conditions, as reported in Fig. 4a and 4b, results show that IFCA's global model is less accurate and that the clustering produced is not as relevant as other methods. Since IFCA produced worse results than other approaches in favourable conditions, we did not test it further on lower selection rates.

Also, we compare with the AVG approach followed by a late clustering step that mimics the work of [6]. Called 'Late' attribution strategy, this approach is of interest but actually imposes a strong expert choice or an optimization on the round selection where clustering is applied and is then case study dependent. It is thus more costly as it does not rely on a single training run because of such calibration requirement. In our experiments, we apply clustering at round 10 when models start converging and metrics reaching at least 50% of their maximum value.

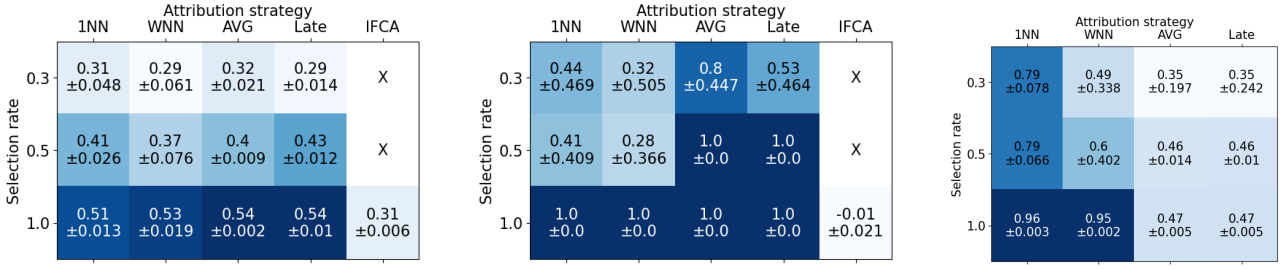
Fig. 4a, 4b, 4c present respectively the accuracy, ARI and silhouette scores on the balanced Cifar10 validation set after the proposed CFL training process. We report the average results over 4 trials with different initialization and the estimated standard deviation envelope range. One can first observe that highest metric scores are obtained when all the clients participate to each run. This can be expected since all the similarity edges in the client graph are always up to date and the adjacency matrix is fully measured. This is realistic in a cross-silo scenario but not in the case of cross-device case studies. Then, all the performance indicators have a tendency to decrease as the client participation rate decreases. Indeed, the client graph is partially updated along rounds and impacts on the convergence of clusters. In our case study, a selection rate of 50% moderately impacts on the accuracy and silhouette score for the 1NN and WNN strategies while the ARI score is the worst. Looking at the ARI score envelopes on these methods, we observe a high sensitivity to model initialization seed. This is visible since we rely on few clients but a smoothing effect can be expected if relying on a high number of clients in a larger scale experiment but maintaining a number of target classes lower than the number of clients. Such hypothesis is to be tested in another study.

Comparing 1NN and WNN attribution methods, one observes that 1NN yields overall better metrics. Then a strong assignment to the closest cluster actually helps both clients task performance and their clustering. The AVG and Late strategies can obtain similar accuracy but higher ARI. However their silhouette scores are the lowest since such approaches rely on the attribution of a unique global model before clients local fitting. Their obtained clusters are then very narrow which can introduce interpretation challenges.

In conclusion, when considering case studies where late clustering approaches are not relevant or when high cluster separation requirements exist, then the 1NN attribution method is relevant but remains sensitive to initial conditions if client selection rate is low. The AVG and Late approaches remain preferable if low cluster separability is acceptable.

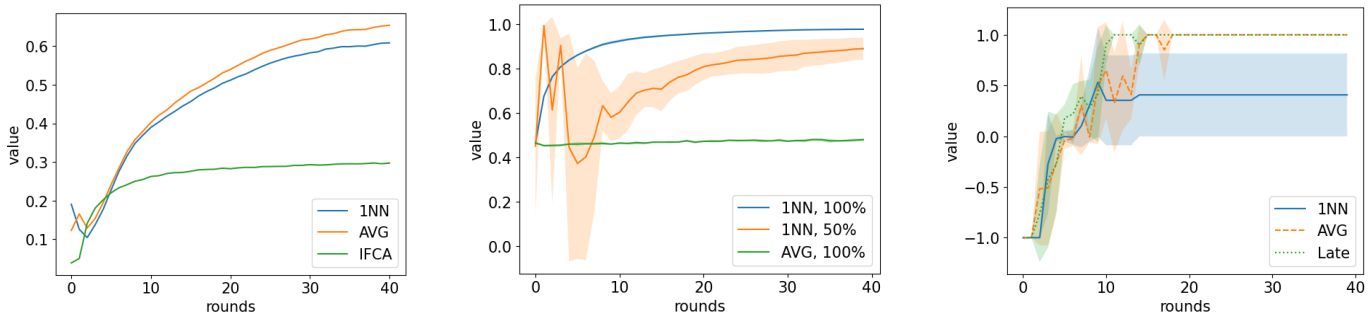
3) On the variability of the results: In this last section, we show the variability of the results with respect to model and computational context initialization seed. As for centralized learning, the variability of the performance levels is also related to model and data pipeline initialization seeds that compromise the reproducibility of the results. Despite the strong current development efforts on the involved mainstream libraries such as Numpy and Tensorflow, results reported in papers are difficult to reproduce. Such variability is stronger when it comes to decentralized learning involving additional higher level (virtualized) server and client coordination libraries. We already showed in Fig. 4b the variability of the ARI values for 1NN and WNN attribution strategies. We thus report, in new experiments, refined monitoring metrics measured along a set of FL training sessions corresponding to m_k = 0.7, a Louvain resolution at 1.0 and a client selection rate at 100%. The same experiment is repeated 4 times with different seeds and we compute the standard deviation envelopes, a higher number of trials being too costly.

Fig. 5b compares the Silhouette curves along training trials for the 1NN strategy at 100% and 50% client participation ratios and with the AVG strategy with 100% client participa-



(a) Accuracy (b) Adjusted Rand Index (c) Silhouette scores

Fig. 4: Impact of model attribution strategies and client selection rate (constant along rounds) with strong data non-iidness ($m_k = 0.7$) and Louvain resolution set to 1.0.



(a) Accuracy with a 100% selection rate (b) Silhouette score (c) ARI with a 50% selection rate

Fig. 5: Performance evaluation of community model attribution strategies along the FL optimization process with varying client selection rates. Data non-iidness is strong (m_k = 0.7) and Louvain resolution is 1.0.

tion. As shown in the previous experiment, AVG is an efficient method for optimizing the global model and at clustering the clients. However, by its very nature of model aggregator, it fails at creating strongly separated clusters. The Silhouette score of AVG remains around 0.5 with very low variability while the 1NN strategy is able to better separate the clusters along the entire FL session, making client communities easier to detect. Variability of 1NN with full clients participation is also very low which encourages its use in the case of cross-silo scenarios. However, we confirm the variability of the 1NN results when clients participate partially to each of the FL rounds. Clients migration in the early rounds clearly impact on variability while such variability is reduced and stabilizes later in the process.

Despite the relevance of the 1NN approach to provide each client with a personalized community model, as a side effect, the 1NN strategy can be less efficient when it comes to the optimization of the global model W_t. But this is not our initial goal in this work. Nevertheless, we initiate a comparison of the task performance of global models related to the 1NN, AVG and IFCA approaches. Fig. 5a indeed shows the superiority of the AVG global model while the 1NN approach converges earlier to a lower value. The IFCA approach obtains the lowest score in this setup. But such performance evaluation should actually take into account model bias considerations as well as potential ethical FL issues, going beyond this work and opening to future work directions.

Further, as shown in the previous evaluations, 1NN and AVG strategies are equally efficient to find the expected communities when all clients are selected at every communication rounds. However, 1NN failed community detection when only 50% of the clients participate to each round as reported in Fig. 4b. We then show in Fig. 5c the evolution of the ARI metric along the FL process in this scenario. First, the AVG as well as the Late strategies both converge rapidly to the perfect detection of the expected communities. After the clustering step at round 10, the Late clustering step perfectly isolates communities until the end of the process. The AVG shows similar behaviours but can report small changes as illustrated at round 20 for a single trial. Regarding the 1NN approach, ARI converges as rapidly but caps to a low ARI value close to 0 with a large and constant variability. This illustrates the instability of the clustering generated by similarities measures with varying degrees of freshness. An interesting future direction can relate to the improvement of the partitioning method in such conditions.

Finally, community models are evaluated and their performances are compared to the one of the global model. The aim is to verify the relevance of the community models on the validation data of their related clients and compare with the global model. Fig. 6 shows the evolution of the global and community models on sets of validation data along the

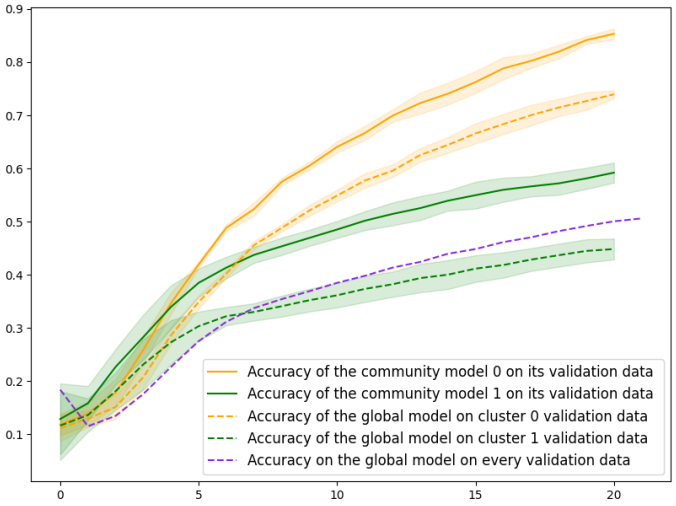


Fig. 6: $m_k = 0.7$, 1nn, 100% selection rate, 1.0 resolution

training rounds. One first observes that the proposed CFL strategy provides a regular increase in the model performances such that client migration does not introduce significant performance instability. One also observes that community models are better performing than the global model when tested on the validation data corresponding to their related clients. The global model performance evaluated on the whole (merged) validation data provides a lower performance. Such results show the importance of optimizing community models in addition to a global model when non-iid data might not allow for a relevant aggregation of local models. The global model is still of interest since it provides a good initial model for new clients before the identification of their appropriate community. The proposed CFL strategy opens new research perspectives on methods to facilitate the community detection in CFL (*e.g.*, how to compare models reliably?), on methods to stabilize the clients' models clustering and on contributions on model convergence to a satisfactory solution while allowing clients to change community.

VI. Conclusion

This work proposes a Clustered Federated Learning approach that maximizes both client task performance and clients community detection. We show that, relying on the sole communication of local and global models between clients and the central server, client community detection is possible. In our experiments, we consider a reference data partitioning relying on the dataset underlying ontology of Cifar10. This ontology can be identified by the server when the label distribution is non-iid on the client local datasets. This confirms a potential privacy issues but this also allows for bias mitigation as each community can be processed equally. Further experiments will extend this approach on large-scale case studies that can relate to both community detection and security taking advantage of community-based federated learning. Bias mitigation based on community detection is another interesting direction to explore.

VII. Acknowledgement

This work has been done thanks to the facilities offered by **masked following anonymous submission rule**.

References

[1] Jeremy Bernstein, Arash Vahdat, Yisong Yue, and Ming-Yu Liu. On the distance between two neural networks and the stability of learning. *Advances in Neural Information Processing Systems*, 33:21370–21381, 2020.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[3] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–9. IEEE, 2020.

[4] Yasmine Djebrouni, Nawel Benarba, Ousmane Touat, Pasquale De Rosa, Sara Bouchenak, Angela Bonifati, Pascal Felber, Vania Marangozova, and Valerio Schiavoni. Bias mitigation in federated learning for edge computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–35, 2024.

[5] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674, 2021.

[6] Fabiola Espinoza Castellon, Aurélien Mayoue, Jacques-Henri Sublemontier, and Cedric Gouy-Pailler. Federated learning with incremental clustering for heterogeneous data. In *IJCNN 2022 - 2022 International Joint Conference on Neural Networks*, page 10.1109/IJCNN55064.2022.9892653, Padoue, Italy, July 2022. IEEE, IEEE.

[7] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered Sampling: Low-Variance and Improved Representativity for Clients Selection in Federated Learning. In *ICML 2021 - 38th International Conference on Machine Learning*, online, United States, July 2021.

[8] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

[9] Riccardo Guidotti and Michele Coscia. On the equivalence between community discovery and clustering. In Barbara Guidi, Laura Ricci, Carlos Calafate, Ombretta Gaggi, and Johann Marquez-Barja, editors, *Smart Objects and Technologies for Social Good*, pages 342–352, Cham, 2018. Springer International Publishing.

[10] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[11] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[12] Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *arXiv preprint arXiv:2305.06329*, 2023.

[13] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[14] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[15] Lingqiao Liu, Lei Wang, and Xinwang Liu. In defense of soft-assignment coding. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, page 2486–2493, USA, 2011. IEEE Computer Society.

[16] Othmane Marfoq. *Tackling heterogeneity in federated learning systems*. Theses, Université Côte d'Azur, December 2023.

[17] Othmane Marfoq, Giovanni Neglia, Laetitia Kameni, and Richard Vidal. Federated learning for data streams. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 8889–8924. PMLR, 25–27 Apr 2023.

[18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[19] Chandan K Reddy and Bhanukiran Vinzamuri. A survey of partitional and hierarchical clustering algorithms. In *Data clustering*, pages 87–110. Chapman and Hall/CRC, 2018.

[20] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[21] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[22] Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.

[23] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

[24] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

[25] Yousef Yeganeh, Azade Farshad, Nassir Navab, and Shadi Albarqouni. Inverse distance aggregation for federated learning with non-iid data. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2*, pages 150–159. Springer, 2020.