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*—This research addresses the challenge of community detection in federated learning environments where data is non-independent and identically distributed across clients. We propose a CFL (Clustered Federated Learning) approach that groups clients into communities based on their model similarities during training. The proposed method is based on the integration of three fundamental elements, namely: the Louvain clustering algorithm, a model similarity measurement system, and a strategy for attributing aggregated models to clients. The primary benefit of this approach is its capacity to discern client communities without the need for pre-existing information, while simultaneously enhancing task performance. The Cifar10 dataset was used to conduct a comprehensive analysis of the method's response to various factors, including the degree of data distribution imbalances, different model initialization approaches, varying client participation rates, and different strategies for assigning clients to clusters. Our evaluation extends beyond traditional metrics by encompassing both model accuracy and clustering quality. When compared to existing CFL methods on an image classification problem, our approach demonstrates advantages through continuous clustering throughout training, flexible client reassignment between groups, and maintained model quality. The approach integrates smoothly with standard federated learning frameworks and improves both task performance and community detection. The results illustrate the efficacy of our clustering approach in identifying relevant communities of related target classes. Finally, the conducted experiments have identified specific avenues for further research that will extend the proposed global framework. The code associated to this work can be found at *https://github.com/albenoit/DeepLearningTools

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

I. Introduction

Federated Learning (FL) has emerged as a promising approach for privacy-enhanced, secure decentralized machine learning [18]. Yet, the inherent heterogeneity and diversity of data across clients poses significant challenges, limiting the effectiveness of a single federated model [11], [16]. Recent advances in model personalization [23] and particularly *Clustered Federated Learning* (CFL) [5] have addressed this challenge by grouping clients based on similarity criteria, creating intermediate models that bridge local and global models. While these methods have demonstrated improved task performance by leveraging both similar local model aggregation and global knowledge, they primarily focus on performance optimization rather than exploring the potential for community detection. Furthermore, crucial questions remain unanswered regarding the effectiveness of model similarity measures, clustering strategies, and result stability across various factors such as client participation rates and model initialization. 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. The identification of communities during training opens new avenues for model optimization, particularly in challenging scenarios involving *non-iid* data and data streams, by optimizing community models for clients with similar data distributions. We explore this direction that continuously detect client communities along training, and we integrate it into a clustered federated learning approach. Using a controlled dataset, we evaluate our method's clustering effectiveness while providing insights into result variability across different methodological choices and initializations. Our framework illustrated in fig. 1 introduces a community model attribution method that seamlessly integrates with state-of-the-art approaches in 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 clients' data.
2) Community models detection 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 structured as follows: the second section gives an overview of state-of-the-art CFL methods. The third and fourth sections present the problem statement and our contributions, respectively. The fifth section elaborates on the performance evaluation, explores the sensitivity of our approach to critical factors such as data non-iidness and provides a comparison of attribution strategies according to their hyperparameters. Finally, the last section discusses the performance and scalability of our proposal and conclude the paper.

This research was supported by the 2024 AFREU project, funded by Savoie Mont Blanc University, and facilitated by the MUST datacenter, jointly maintained by Savoie Mont Blanc University and CNRS (French National Centre for Scientific Research).

II. Related work

The initial proposal of Federated Learning [18] assumes that a single model can be applied to all clients, even in cases where their data distributions are non-iid. However, later studies [11] have demonstrated that FL is highly sensitive to data distribution-related bias issues. Indeed, the averaging of client models with markedly disparate distributions may result in suboptimal optimization of the global model. Furthermore, in realistic scenarios, clients and their data evolve over time, creating new forms of bias that need to be addressed. Bias mitigation is thus an active research direction with promising approaches including model aggregation optimization [4] and appropriate data sampling strategies as presented in [7], [17]. Following this direction, Personalized Federated Learning [23] has been recently proposed as a means of providing each client with relevant shared models. Among the various methodologies, those based on CFL isolate groups of clients with a view to mitigating bias and providing 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 from the literature.

In their work [7], Fraboni et *al.* propose client clustering as a means of mitigating bias. In contrast to the other presented works, this study considers client sampling clustering with the aim of reducing communication costs with the server while increasing the number of clients represented and reducing the variance of the client weights. Hierarchical clustering is applied to the gradient between each client and the global model.

Sattler et *al.* introduce *Clustered Federated Learning* [21] as a new federated multitask learning method. The method 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 bipartioned 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 enhance the efficiency of FL when clients possess non-iid data. In contrast with the prevailing approach in the CFL literature whereby clustering is typically conducted on the server side, Ghosh et *al.* posit that cluster formation on the client's side can reduce the computational burden on the server. Consequently, IFCA clients are required to identify their cluster by themselves by selecting the model that minimizes their loss. This results in an increased computational cost on the clients as they must test several models before identifying their cluster. Furthermore, IFCA necessitates the number of clusters to be known in advance which limits its applicability in real scenarios. Additionally, 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 to group clients based on their similarities in gradient update patterns. A particular mechanism is put in place for the allocation of a model to new clients, with the objective of improving scalability. This approach is limited to supervised learning, as the migration of clients from one cluster to another is triggered when the distribution of data labels in the client's cluster evolves above a predefined threshold. The calibration of such a policy may prove challenging, and it may entail the communication of client label distributions. FlexCFL employs a similarity metric, termed *Euclidean Distance of Decomposed Cosine Similarity* which decomposes the updates into m direction using *Singular Value Decomposition*, subsequently computing a cosine similarity. The primary benefit of this metric is that it circumvents the concentration phenomenon that arises from the dimensions of the model parameters. Furthermore, they propose a method for clustering clients based on their gradients, using EDC to handle changes in data. FlexCFL is compared to IFCA and demonstrates superior global model optimization on multiple datasets (MNIST, FashionMNIST, FEMNIST) and models (CNN, MLP, MCLR). However, as with IFCA, the number of clusters obtained with FlexCFL must be known in advance, and the quality of the clustering is not evaluated.

In their study, Briggs et *al.* propose hierarchical clustering as a means of grouping clients based on the degree of similarity observed in their local updates [3]. In contrast to previous methods which cluster clients after each round, this approach first train models without any clustering step for n communication rounds and subsequently employs hierarchical clustering. Next, clusters are trained separately until the end of the training. The application of clustering after a specified number of rounds allows clients to converge to a global shared solution before being clustered. However, this approach may result in the formation of clusters that are more similar in nature and potentially introduce a bias related to the global model. The authors demonstrate that in a non-iid setting, initiating the clustering process in the first communication rounds is more advantageous. In contrast to IFCA or FlexCFL, this method does not require the number of clusters to be predetermined, thus offering enhanced flexibility. Nevertheless, the datasets are expected to remain static throughout the training process. Similarly, Espinoza Castellon et *al.* [6] propose a client partitioning strategy that is applied a single time, in a late round along a regular federated learning session. The use of model cosine similarity based clustering enables the grouping of clients after the global model has reached a point of convergence, allowing them to specialize in the remaining rounds while sharing their knowledge with their similar neighbours. The selection of the clustering round remains a challenging aspect of this process, particularly in light of the potential of bias introduced by the global model.

The works presented in this section are focused on the optimization of a single or a set of shared models, with a variety of strategies employed. It is notable that none of the aforementioned works assess the quality of their clustering.

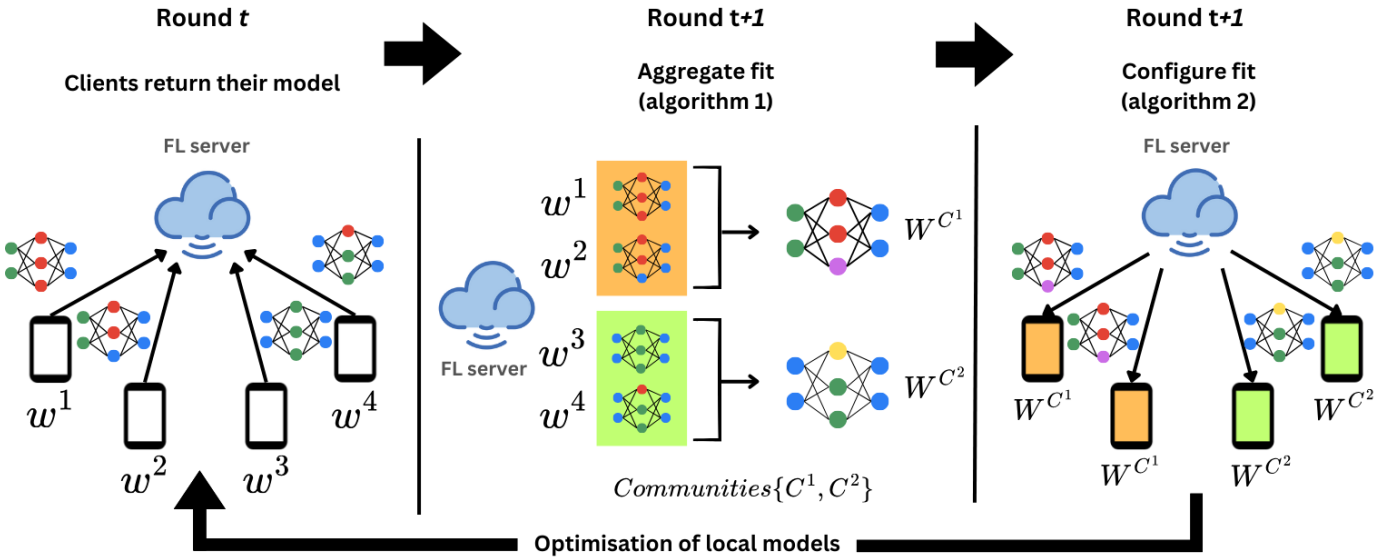


Fig. 1: Overview of our CFL community detection approach over a round. Firstly, clients participating in the previous round send their trained model to the server. Then the server runs the aggregate fit function described in the algorithm 1 and detects client communities. Finally, the server executes 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.

Sattler et *al.* highlight that CFL presents a novel privacy concern, as it appears feasible to infer information about clients from their models at each round [21]. In this study, we extend this concept and seek to achieve both optimal task performance and the discovery of communities of clients from their models throughout the Federated Learning process.

III. Problem statement

In the pursuit of a comprehensive approach to CFL that does not require minimal prior knowledge of data and client behavior, we focus on a setup that uses standard model aggregation methods. In this configuration, only the local model updates are communicated from clients to the server, with no client-to-client communication or exchange of private information. This setup establishes a baseline that can later be enhanced through improvements in aggregation, client sampling, and other complementary techniques outlined in the previous section. We then assess the effectiveness of these strategies in relation to their specific configuration, model initialization, and factors such 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 architecture 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 subsets 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 behaviors 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

In the absence of prior knowledge regarding client communities and their data behaviors, the objective of the server is twofold: firstly, to assist clients in optimizing their task performance and, secondly, to facilitate the detection of their communities. It is imperative that clustering does not introduce bias and result in a reduction in the client task performance due to the client being locked into a specific cluster. This constraint also presents opportunities for further research in the area of federated learning on data streams and time-evolving data, as discussed in [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]. In this approach, the federated clients are considered as individual entities and aim to be partitioned based on similarities in their model parameters. This similarity metric serves as the edges in a client graph, facilitating the grouping of clients with comparable models.

We then introduce a client clustering and community-based model computation process, performed on the server side throughout the optimization process. The objective of this approach is to eliminate the necessity for case-specific clustering hyperparameter search as for [3], [6]. Furthermore, this strategy allows for the continuous monitoring of client community assignments throughout the federated optimization process, which can facilitate the detection of security threats, such as detecting poisoning attacks. [6].

This section presents our approach to detecting client communities, which involves selecting a similarity metric, partitioning clients in a scalable manner and synthesizing the community models and a global model.

A. Similarity metric

The most common methods for evaluating the similarity between models are representational and functional similarity metrics [12], which respectively compare each neuron activation and output. The most commonly 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. Given the server-side priors detailed in the previous section and the general difficulty of collecting relevant, unbiased and privacy-preserving centralised data, we do not consider such an approach. As an alternative, distance metrics may be employed to evaluate the dissimilarity between models according to their parameters values. Such metrics are characterized by a low computational cost. The most typical ones reviewed in [12] are *L-norms*, *cosine distance* and *Procrustes disparity*. Furthermore, the concept of *deep relative trust* or *trusted distance* introduced with the *Fromage* optimizer [1] provides a means of expressing the functional distance between models with similar structures. The upper bound is given by equation 2 where L is the number of model layers, $w_{a,l}$ and $w_{b,l}$ are the parameters of models a and b at layer l.

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

In a preliminary study, we compared various similarity metrics and identified the trusted distance as the most relevant due to its stable results. Nevertheless, this choice is not crucial for the purpose of the study, as alternative metrics, such as the cosine distance that is commonly employed in CFL, yield comparable results, albeit with slightly less selectivity. Subsequently, the distance values are transformed into similarity values mapped to the range [1,0] by applying eq. 3 prior to any other transformation. This facilitates comparison and provides normalized values for the following processes:

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

where v is the distance value to be normalized while min and max are respectively the lowest and highest distance observed across all the client model pairs in a given round.

Normalized similarity values may be subjected to further post-processing in order to exert an influence on the clustering of the client graph. Three main approaches are possible, their choice impacting on cluster composition : "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 as the exponent value increases, similarity values decrease, which can make it challenging to distinguish between low and high similarities if the exponent becomes too large.

B. Client partitioning method

The process of identifying client groups can be conceptualised as a clustering problem, whereby the complete client adjacency matrix is constructed through the calculation of similarities between each pair of clients. Moreover, it can be formulated as a community detection problem if only a subset of the client distances is considered. In cross-device scenarios with a large number of clients, the full adjacency matrix may become a significant computational burden. Nevertheless, it remains a viable proposition in a cross-silo FL configuration with a reduced number of clients. Furthermore, the partitioning algorithm entails an additional cost in comparison to the basic form of FL. The selection of an appropriate algorithm should be informed by a number of factors, including the desired clustering quality, the computational cost requirements, the potential for scalability, and the cost of hyperparameter tuning. A number of methods for client clustering have been proposed in the literature, as surveyed in [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 wide range of client numbers while requiring few hyperparameter searches. The main advantage of Louvain clustering, compared with traditional methods such as K-Mean, is its ability to create clusters without knowing the number of groups to be created. The Louvain clustering algorithm aims to optimize the modularity of a graph, which measures the strength of division of a network into communities. Modularity quantifies the difference between the actual number of edges within communities and the expected number of edges if the network were randomly connected. Its main hyperparameter termed "resolution", regulates the size

of the clusters and thus the fragmentation of the communities. Low-resolution values lead to numerous small clusters, while high values lead to a limited number of large 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, a comprehensive strategy for client partitioning and community cluster attribution must be established when considering the entirety of the CFL process. In light of the aforementioned problem statement and the availability of a minimum level of knowledge regarding client behaviors, the server is effectively blind and thus requires a meticulous clustering process that avoids any biasing of clients by locking them into a non-relevant cluster. We then put forward a flexible approach inspired by [5] that applies client partitioning on a regular basis throughout the optimization process. However, our approach differs in that clients clustering occurs after each round, rather than relying on a tuned trigger that relies on a label shift measure. Furthermore, we examine a variety of model attribution strategies that would allow any client to move to the most relevant community model at any time during the process. The choice of strategy therefore has an impact on task performance and clustering quality.

We consider three aggregated model attribution strategies, which are applied after the client sampling step of each federated round *t*. Each strategy aims at providing participating clients q with a personalized aggregated model w_{t+1}^q.

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.
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 based on a semi-soft assignment strategy [15] shown in eq. 4. It computes the average of neighboring community clusters, introducing more flexibility through the parameter β. This parameter modulates the influence of neighboring clusters, while an early cut-off effect is introduced by selecting the nearest neighbors, which reduces the influence of distant or 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. 1 illustrates the global workflow of our contribution over a single round, this process being repeated along training rounds.

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

Algorithm 1 Model aggregation step for a FL round t

procedure AGREGATE($clientUpdates$)
Require: $clients$ the graph of known 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

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

V. Evaluation

A. Experimental setup

This section presents an approach to evaluating our contribution in a manner that is comparable to that of state-of-

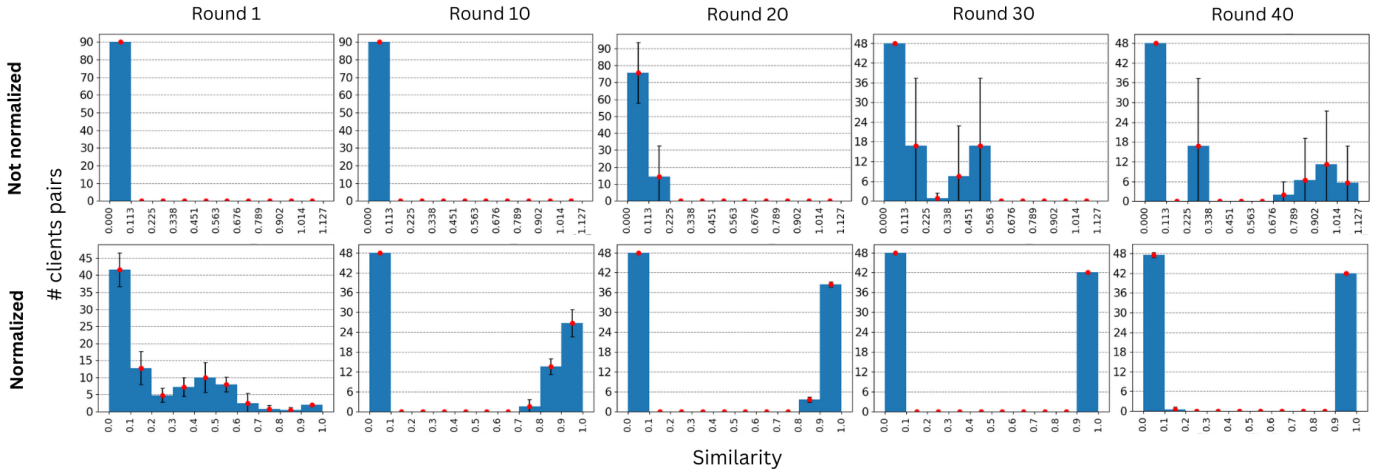


Fig. 2: Client similarity distributions over 2 FL optimization trials without (top row) or with (bottom row) normalization. The histograms are averaged from 4 experiments with different random seeds. For visualization and as for the applied process, top row shows the similarity values calculated from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

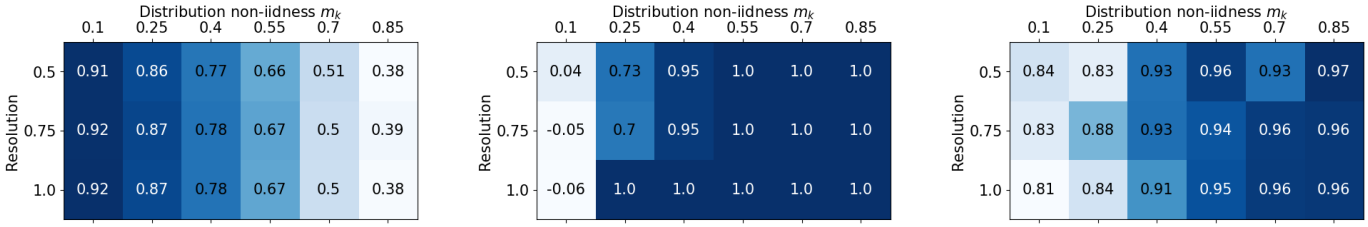
the-art methods. The objective is to identify a relevant case study in which realistic communities can coexist without the need for artificial exaggeration of their behaviors, which could be easily identified by high-capacity models. We therefore consider the classification problem related to the standard Cifar10 dataset [14] with preserved image orientations and color distributions, without introducing any related data augmentation. Furthermore, we apply a client sampling policy on the training dataset to create communities based on semantic label distribution shifts, with the aim of verifying the capability of CFL approaches to detect such communities while maximizing classification performance. This is done by relying on the underlying label ontology of the dataset.

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*. The Cifar10 dataset comprises 5,000 training images and 1,000 test images per class. It is assumed that there are two superclasses representing higher levels of the dataset labels ontology: 'animals' and 'vehicles'. Consequently, the standard 10-class recognition problem is extended to include the additional goal of detecting these two superclasses or communities while clients are training with different class 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 for conducting all the proposed experiments at a reasonable cost, which totals 8,000 CPU hours, not including calibration trials. These experiments were conducted on a single Intel Xeon(R) Gold 5118 CPU @ 2.30GHz, which was equipped with 40GB of RAM and used the TensorFlow 2.14.1 and Flower 1.9.0 libraries within our open-source framework.

The following section assesses the task performance and cluster quality of the models on the official Cifar10 validation set, a collection of balanced 10k samples, 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]. The Silhouette score, meanwhile, evaluates how closely clients are associated with their cluster and how distant they are from other clusters [20]. Each experiment is conducted over 40 rounds, with 10 clients, and the results are averaged over four 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: The first experiment entails a comparison of the aforementioned metrics in accordance with varying data distributions and Louvain resolutions. The objective is to ascertain which data distribution and resolution are most advantageous for community detection. Subsequently, an exaggerated imbalanced data context is constructed based on the Cifar10 dataset. In this context, 10 clients participate *to each* of the training rounds of an FL session, with each client having a specific majority class with varying importance 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 $m_k = 1.0$, it is ensured that each client owns a single class that is distinct from those owned by other clients. This represents the most non-iid case in this configuration.



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

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

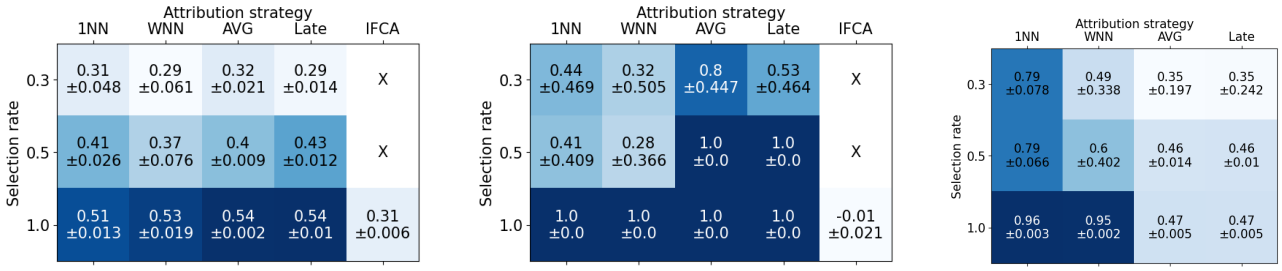
Fig. 3a illustrates the resulting classification accuracy measures as a function of the degree of non-identity controlled by m_k and the Louvain resolution on the balanced validation set. The degree of non-iidness has a significant impact on the accuracy of the classification, whereas the Louvain resolution exerts no notable influence. However, in contrast to accuracy, ARI exhibits a markedly higher value when training on non-iid data, as illustrated in Fig. 3b. Furthermore, the clusters are more dense and better separated when the client data is non-iid and the resolution is higher, as illustrated by Silhouette scores in Fig. 3c. The results demonstrate that the optimal hyperparameters for community detection are a Louvain resolution of 1.0 and a data distribution parameter of $m_k = 1.0$. A higher Louvain resolution enables the formation of fewer, larger clusters, which is an effective approach for detecting only two communities, as is the case for the specific evaluation conditions under consideration. Additionally, the value of $m_k = 0.7$ represents a compromise between enhanced model optimisation and a diminished Silhouette score. In conclusion, it can be stated that when the data is more balanced across clients, the local models are more similar, which in turn makes community detection more challenging. Furthermore, it is noteworthy that the clustering of models becomes more straightforward when clients are biased towards a specific majority class. This ultimately aligns with the clustering of the animals and vehicles superclasses. Our CFL approach is then capable of differentiating between high-level semantic categories, even in the absence of explicit training and despite significant variability in the training data, even at the class level. This is a promising result that will require further validation in other contexts through subsequent studies. From a data-driven perspective, this outcome demonstrates that when clients exhibit a pronounced bias towards a specific class (i.e., when m_k is high), the local models of a given superclass tend to converge towards a similar solution. This occurs despite the considerable diversity of the objects' backgrounds across images that can be similar across superclasses. Our results complement those of previous work, such as IFCA [8], which demonstrated the capability of CFL to differentiate clusters artificially created by image rotations. Furthermore, our findings confirm the potential privacy issue discussed in [21], as the server is aware of the community model to which a given client belongs.

Furthermore, we investigated the influence of the model similarity measure formulation on the separability of the community. Fig. 2 illustrates the evolution of similarity distributions along the federated rounds comparing two trials with either non-normalized or normalized distances. It can be observed that, with normalized distances, the distributions converge more rapidly to the two expected modes, which illustrate high intra-cluster and low inter-cluster distances.

2) Model attribution and selection rate: In this section, the relevance of each proposed model attribution method described in section IV-C is compared with a varying ratio of participating clients. For the purposes of this analysis, we consider a non-iid training data distribution for each client with $m_k = 0.7$, as discussed in the previous section. In order to ensure consistency across experiments, the client selection rates remain constant. Louvain resolution is set to 1.0 for each method. Finally, we add to the comparison the results provided by the IFCA state-of-the-art paper [8].

The original IFCA method has been tested on rotated Cifar10 images with the objective of improving the global model. As detailed in the original paper, the images were successfully partitioned into two clusters: one comprising regular images and the other comprising rotated images. Nevertheless, the clustering of images according to their semantic superclass represents a distinct and potentially fruitful avenue for further investigation, which could be effectively addressed through the aforementioned approach. In the course of our experiments, we employed the IFCA method with the identical dataset described in the preceding sections. It thus should be noted that the IFCA method is not compared in the same conditions as those described in the original paper, as the distribution of data among clients is entirely distinct. In these conditions, as illustrated in Fig. 4a and 4b, the results demonstrate that the global model produced by IFCA is less accurate and that the clustering produced is not as relevant as that produced by other methods. Given that IFCA produced inferior results to other approaches in favourable conditions, further testing was not conducted on lower client selection rates.

Moreover, we assess our approach in comparison to the AVG methodology, which incorporates a late clustering phase that emulates the methodology proposed by Espinoza-Castellón [6]. It is notable that this 'late' attribution strategy necessitates a considerable input of domain expert effort in optimization with respect to the round at which clustering is applied. This renders the approach case study dependent,



(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.

thereby increasing the cost due to the calibration requirements. In our experiments, we have applied clustering at round 10, in accordance with the recommendations of this approach, which coincides with the point at which models begin to converge and metrics reach at least 50% of their maximum value.

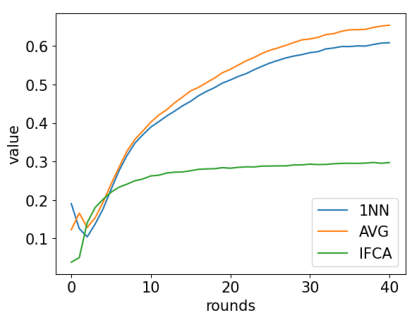
Figures 4a, 4b and 4c illustrate respectively, the accuracy, ARI and silhouette scores on the balanced Cifar10 validation set after the proposed CFL training process. The results presented here are the average of four trials conducted with different initialisations, along with the estimated standard deviation envelope range. It can be observed that the highest scores are obtained when all clients participate in each run. This is to be expected, given that all similarity edges in the client graph are constantly updated and the adjacency matrix is fully measured. This is a realistic assumption in a cross-silo scenario, but not in the case of cross-device case studies. Subsequently, a reduction in client participation rate is observed to result in a decline in performance indicators. It is evident that the client graph undergoes partial updates during each round, which in turn affects the convergence of clusters. In the case study presented here, a selection rate of 50% has a moderate impact on the accuracy and silhouette score for the 1NN and WNN strategies, while the ARI score is the most adversely affected. An examination of the ARI score envelopes for these methods reveals a notable sensitivity to model initialization seeds. This is evident given the limited number of clients. However, a smoothing effect may be anticipated if a larger number of clients were to be included in a larger-scale experiment, provided that the number of target classes remains below the number of clients. This hypothesis requires further investigation.

A comparison of the 1NN and WNN attribution methods reveals that the former yields superior metrics overall. A strong assignment to the closest cluster is beneficial for both clients in terms of task performance and clustering. The AVG and Late strategies achieve comparable accuracy but higher ARI. However, their silhouette scores are the lowest due to the reliance on a unique global model before local fitting by clients. The resulting clusters are narrow, which can introduce challenges in interpretation.

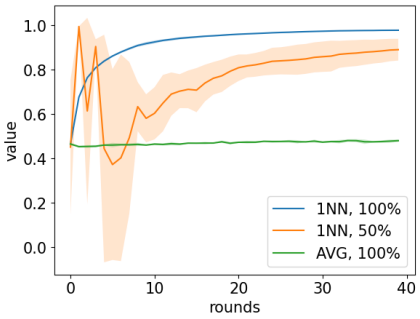
In conclusion, when considering case studies where late clustering approaches are not relevant or when high cluster separation requirements exist, the 1NN attribution method is a relevant approach, although it remains sensitive to initial conditions if the client selection rate is low. The AVG and Late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

3) On the variability of the results: This final section presents the influence of the model's initialization seed and the client sampling process on the variability of the results. With regard to centralized learning, such variability is a known phenomenon that may potentially compromise the reproducibility of the results. Such variability is more pronounced in the context of decentralized learning, where additional higher-level server and client coordination processes are involved. As already demonstrated in Fig. 4b the variability of the ARI values for the 1NN and WNN attribution methods is evident. In order to gain further insight, additional experiments were conducted with metrics monitored throughout the FL training sessions. The experiments relied on the following parameters: $m_k = 0.7$, Louvain resolution at 1.0 and client selection rate at 100%. The same experiment was repeated 4 times with different initialization seeds and the resulting standard deviation envelopes are reported. However, due to limitations in computational resources, it was not possible to conduct a greater number of trials.

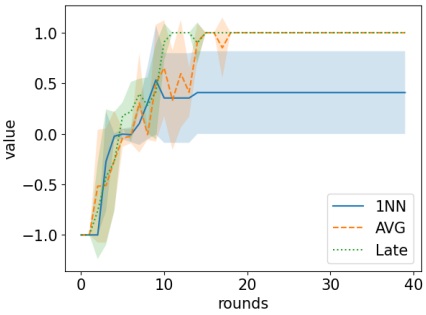
Fig. 5b compares the Silhouette curves along training trials for the 1NN strategy at 100% and 50% client participation ratios and with the AVG strategy with 100% client participation. As demonstrated in the preceding experiment, the AVG represents an effective approach for optimizing the global model and for clustering the clients. However, due to its intrinsic nature as a model aggregator, it is unable to effectively create strongly separated clusters. The Silhouette score of the AVG remains approximately 0.5 with minimal variation, whereas the 1NN strategy is capable of more effectively separating the clusters throughout the entire FL session, thereby facilitating the detection of client communities. Moreover, the 1NN strategy demonstrates minimal variability when all clients participate, which makes it suitable for use in cross-silo scenarios. However, it is important to note that the 1NN results exhibit variability when clients participate partially in



(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.

each FL round. The impact of client migration in the initial rounds on this variability is evident, 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. However, this is not the objective of the present study. Nevertheless, a comparison of the task performance of global models related to the 1NN, AVG and IFCA approaches is presented. As expected, Fig. 5a demonstrates the superiority of the AVG global model, while the 1NN approach reaches a slightly lower value. The IFCA approach obtains notably the lowest score in this setup. However, such performance evaluations should be extended in order to also consider model bias and potential ethical issues in FL, which extend beyond the scope of this work and provide avenues for future research.

Furthermore, as demonstrated in the preceding evaluations, the 1NN and AVG strategies are equally effective in identifying the anticipated communities when all clients are selected in each communication round. However, the 1NN strategy was unable to detect communities when only 50% of clients participated in each round, as illustrated Fig. 4b. More into the details, fig. 5c illustrates the evolution of the ARI metric along the FL process in this scenario. First, both the AVG and Late strategies demonstrate a rapid convergence towards the optimal detection of the expected communities. Following the clustering phase applied at round 10, the later cluster personalization phase maintains the communities until the conclusion of the process. The AVG approach demonstrates comparable behavior but is capable of reporting minor changes, as illustrated at round 20 for a single trial. With regard to the 1NN approach, ARI converges rapidly but reaches a low ARI value close to 0 with a substantial and consistent variability. This illustrates the instability of the clustering generated by similarity measures with varying degrees of freshness. One avenue for future research could be the improvement of the partitioning method in such conditions.

Finally, the efficacy of community models is assessed and their performance is benchmarked against that of the global

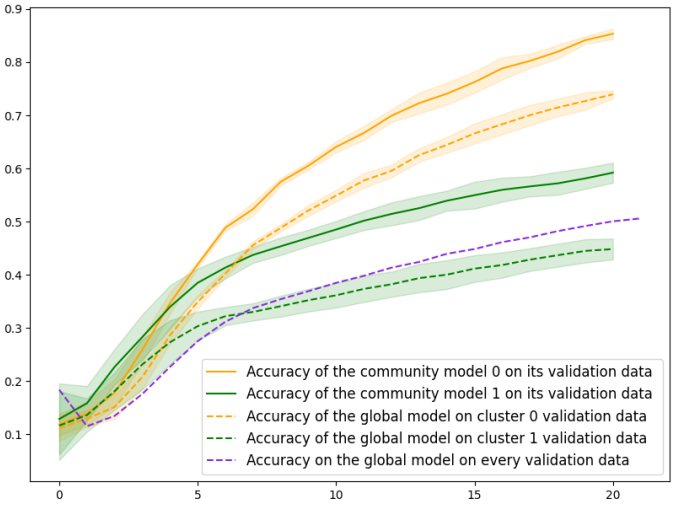


Fig. 6: $m_k = 0.7$, 1nn, 100% selection rate, 1.0 resolution

model. The aim is to determine the relevance of the community models on the validation data of their respective clients and to compare them with the global model. Fig. 6 illustrates the progression of the global and community models on sets of validation data throughout the training rounds. It can be observed that the proposed CFL strategy result in a consistent improvement in model performance, thereby demonstrating that client migration does not lead to significant performance fluctuations along the optimization process. Furthermore, it is evident that the community models exhibit superior performance compared to the global model when evaluated on the validation data specific to their respective clients.The global model performance evaluated on the whole (merged) validation data provides a lower performance. These findings highlight the necessity of optimizing community models in addition to a global model when non-iid data may not permit a relevant aggregation of local models. The global model remains a valuable resource, offering an initial model for new clients until their appropriate community is identified. The proposed CFL strategy presents new avenues for research, including the development of reliable methods for community

detection in CFL, the stabilization of client models, and the investigation of convergence towards a satisfactory solution while allowing clients to change community.

VI. Conclusion

This work proposes a clustered federated learning approach with the objective of optimizing both client task performance and clients' community detection. The proposed methodology involves bidirectional exchange between clients and the central server, where client local models and community models are iteratively updated throughout the federated learning process. In the course of our experiments, we considered a reference data partitioning strategy based on the underlying ontology of the Cifar10 dataset. The ontology can be identified by the server when the label distribution is non-iid on the client local datasets. This confirms the potential for privacy issues but also allows for bias mitigation as each community can be processed equally. The variety of experiments also identifies further specific research directions, thus enabling the extension of the proposed global approach.

Firstly, a large-scale experiment will allow the method to be evaluated in a cross-device case study, thereby facilitating further investigation into the impact of the ratio between clients and communities at scale. In addition, the effectiveness of refined client clustering strategies can be evaluated to improve community detection convergence. In particular, the relevance and stability of the clustering provided by the Louvain algorithm can be enhanced and compared to other scalable methods. Another avenue for further investigation is the mitigation of bias, as well as the enhancement of privacy and security based on community detection. Finally, the relevance of the approach in more complex scenarios, such as Cifar100, is to be studied.

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.