DCFL: Dynamic Clustered Federated Learning under Differential Privacy Settings

Andrea Augello, Giulio Falzone, and Giuseppe Lo Re

In International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)

DOI: https://doi.org/10.1109/PerComWorkshops56833.2023.10150285

Publisher: IEEE

DCFL: Dynamic Clustered Federated Learning under Differential Privacy Settings

Andrea Augello
Department of Engineering
University of Palermo
Palermo, Italy
andrea.augello01@unipa.it

Giulio Falzone
Department of Engineering
University of Palermo
Palermo, Italy
giulio.falzone@unipa.it

Giuseppe Lo Re
Department of Engineering
University of Palermo
Palermo, Italy
giuseppe.lore@unipa.it

***Abstract*—Federated Learning (FL) allows training machine learning models on a dataset distributed amongst multiple clients without disclosing sensitive data. Each FL client, however, might have a different data distribution, with a detrimental effect on the performance of the trained model. In this paper, we present a dynamic clustering algorithm (DCFL) that allows the server to cluster FL clients based on their model updates, letting the server adapt to changes in the data distribution and supporting the addition of new clients. Moreover, we propose a novel distance metric to estimate the distance between model updates by different clients. We evaluate our approach in a wide range of experimental settings, comparing it against the standard *FedAvg* algorithm and divisive clustering on the EMNIST dataset. Our approach outperforms the baselines, yielding higher accuracy and lower variance for the participating clients.**

***Index Terms*—Federated Learning; Differential Privacy; Dishomogeneous data distribution; Dynamic clustering**

I. Introduction and related works

Federated Learning (FL) is a machine learning technique that trains an algorithm across multiple decentralized clients holding local data samples without exchanging sensitive data [1]. Local models are trained on local data samples, and only parameters (either model weights or computed gradients) are exchanged. FL, over the course of multiple communication rounds, can train a global model without collecting private data from the users producing the training data. Compared to traditional paradigms where all the data are available to a central entity training the model, FL preserves user privacy.

Nonetheless, private data can still be leaked: model parameters can disclose information on the presence of specific entries in the training data. Inference attacks can detect whether a specific user participated in the FL scheme and whether the targeted user trained the model with some specific records using nothing but the updates [2].

To protect the federated clients from disclosure of their private data, local Differential Privacy (DP) has been proposed as a solution [3]. DP mechanisms add random noise to the results of computation performed on the data that needs to be protected from disclosure [4]. In the case of FL, this computation corresponds to the client updates. Before transmitting the locally trained models, clients perform norm clipping to the gradient and add Gaussian or Laplacian noise, so that privacy is preserved even if the server aggregating model updates is compromised [5]. Clearly, there are some trade-offs between the privacy gains and the reduction in model accuracy due to the noisier updates [6].

FL is gaining popularity in human activity recognition [7] and mobile crowdsensing applications [8], [9] as it allows to perform learning tasks without centrally collecting the user data. In these contexts, data collected by various participating clients can have dishomogeneous quality [10] and might have different statistical properties for each client [11], potentially worsening the trained model performance or prevent convergence altogether [12]. Moreover, in some cases, the data might be distributed in a severely non-Independently and Identically distributed (IID) way. In these non-IID cases, some users might prefer to abstain from participating to the FL scheme. Indeed, if the user data distribution differs from the majority, the user would not have any benefit by participating in a FL scheme [13] and their presence might harm the overall performance of the trained model. This issue can be mitigated adopting weighting schemes more sophisticated than the commonly used *FedAvg* [14], as to reduce the variance of the model accuracy achieved on the local datasets by the federated clients [15]. This objective is sometimes achieved through the use of a root dataset in the aggregating server: the root dataset can be used to obtain useful information about the quality of the received model updates and to optimize the next training round [16].

This approach can mitigate the issues caused by non-IID data with *label distribution skew*, meaning that each client is assumed to hold similar types of local training data but not every client holds data relative to every class. This type of non-IID distribution is common in IoT scenarios [17]. Nevertheless, even if all clients can achieve similar model performances, training a single model with non-IID data can still prevent the clients from obtaining an optimal model [18], especially in the case of *full overlapping attribute skew* where the feature distribution varies across clients, and *label preference skew* where different client might associate different labels to the same feature values. This last type of non-IID distribution is particularly common in crowdsourced data labeling scenarios [12].

To address this issue multiple models can be trained for subsets of the federated clients. For instance, when the global

model perform poorly on the local datasets of a subset of clients compared to the others, a distinct model to be trained separately can be instantiated for these clients [19].

Similar loss values for all the clients in a cluster, however, do not guarantee a correct clustering: a standard FL model reaches a stationary point when the magnitude of the average model update tends to zero. Yet, if the data distributions of the clients are different, the model updates of each client might be non-negligible even if they cancel out. The global model could be an equally poor fit for all clients, making it impossible to determine an effective clustering structure by only analyzing the loss. Cosine similarities between clients updates conveys richer information about how each client's ideal model differs from the global one, allowing to more accurately determine how to cluster clients [18]. Such an approach entails iteratively bipartitioning one of the existing clusters whenever the associated model is close to a stationary point but the maximum update module is still high. In the case of noisy updates, as in the case of DP, this bipartitioning approach can perform incorrect cluster assignments [20] and cannot recover from its mistake in the following iterations. Cosine similarity can also be used to cluster clients in a single step using an agglomerative clustering technique [17], but retains most of the aforementioned issues.

Another downside of this approach is that many communication rounds are needed to achieve a correct clustering structure. For this reason, agglomerative clustering has been proposed for FL [21]. After some initial training with a single global model the distances between clients updates are used for clustering.

The agglomerative clustering approach, however, requires the participation of all the federated clients to be successful, does not support the inclusion of additional clients after the clustering phase, and cannot handle data distribution shifts. FlexCFL [22] has been proposed to overcome these issues, supporting newcomer clients and cluster migrations. The assignment of a client to a cluster is performed according to the cosine between their update to a global model and the direction of the clusters models. Even so, this technique requires prior knowledge on the number of clusters, which cannot change during training.

In this work, we present an original model update distance measure and an alternative aggregation algorithm for clustering FL clients with non-IID data. Our Dynamic Clustered Federated Learning (DCFL) approach can handle the inclusion of additional clients during training, cluster migration, client data distribution shifts, and remains effective even under the noisy conditions typical of differential privacy. The remainder of the paper is organized as follows: in Section II we present the proposed client clustering algorithm, in Section III we provide experimental results. Finally, Section IV concludes the paper discussing improvement areas and future development plans.

II. Proposed approach

A. Distance measure

The euclidean distance between vector endpoints does not take into account whether these vectors are converging to the same point or getting further away. This makes the euclidean distance unsuitable to cluster update vectors that may be computed starting from different models. Cosine distance, on the other hand, only considers the angle between local models from a given reference point, ignoring all other spatial relationships, such as the positions of the global models of the clusters.

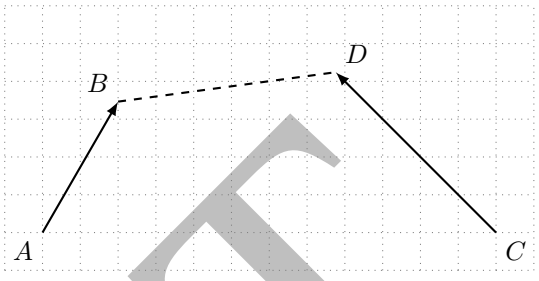

Fig. 1. Example of a possible arrangement of update vectors

In order to address for these issues, we introduce a novel distance measure based on both a divergence measure and the euclidean distance of the models. To illustrate this distance measure we make use of the notation in Fig. 1. The divergence measure is given by Eq. 1 and accounts for both the magnitude and the angle between the vectors considering their position in space.

$$\omega = \frac{|AB|\cos\alpha + |CD|\cos\beta}{|AB| + |CD|} \tag{1}$$

This measure lies in the [-1,1] interval. In accordance with intuition, the highest divergence value, 1, is given by $\alpha = \beta = 0$. This occurs when the two vectors lie on the same line and point outward. Symmetrically, when the two vectors lie on the same line and point toward each other, $\alpha = \beta = \pi$ and the divergence measure is -1. The divergence is 0 when $\overline{AB} \parallel \overline{CD}$ and the vectors are not getting further apart.

This divergence measure is used to scale the euclidean distance of the end points so that, given a fixed $|BD|$, if two vectors are pointing toward each other they are considered closer than if they are getting further apart. Moreover, it is desirable that the distance between two parallel vectors with the same length is not scaled, and still corresponds to their euclidean distance. Finally, the distance measure is given by Eq. 2

$$\mathrm{d}(\vec{AB}, \vec{CD}) = |BD| \cdot e^{2\omega} = |BD| e^{2\frac{|AB|\cos\alpha + |CD|\cos\beta}{|AB|+|CD|}} \tag{2}$$

If $|BD| = 0$, meaning that the two vectors have the same end point, α and β are not meaningfully defined. The divergence index, however, is always in the [-1,1] interval, and thus in this special case the distance is just zero. This distance measure assigns distance equal to zero to all the vectors with the same end point, regardless of their origin.

Another special case is when the update vectors have the same starting point. This is a common occurrence since when two clients belonging to the same cluster produce their update

vectors, these have the same origin. If the update vectors also have the same length, our divergence measure ω behaves similarly to the cosine distance D_C which, given two vectors at an angle γ, is $1 - \cos\gamma$. Without loss of generality, in the following we consider the example in Fig. 2 with $|AB| = |AC| = l$:

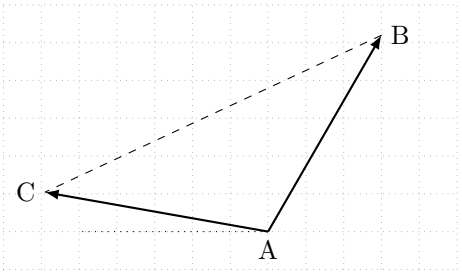

Fig. 2. Example of a generic arrangement of update vectors by clients belonging to the same cluster.

$$\omega = \frac{l(\cos\alpha + \cos\beta)}{2l} = cos(\alpha)$$

Since $\triangle ABC$ is isosceles $\cos\alpha = \cos\beta = \sin\frac{\gamma}{2}$, which can be expressed in terms of $\cos\gamma$ as

$$\omega = \sqrt{\frac{1-\cos\gamma}{2}}$$

Thus, under these assumptions, $D_C = 2\omega^2$.

As shown in Fig. 3, all else being equal, the greater the difference in length between the update vectors, the higher is the assigned divergence. Indeed, even if the angle between the vectors is small, if one is longer than the other and they have the same origin, overall they are getting further apart. Due to norm clipping typical of differential privacy, the update vectors will typically have similar lengths.

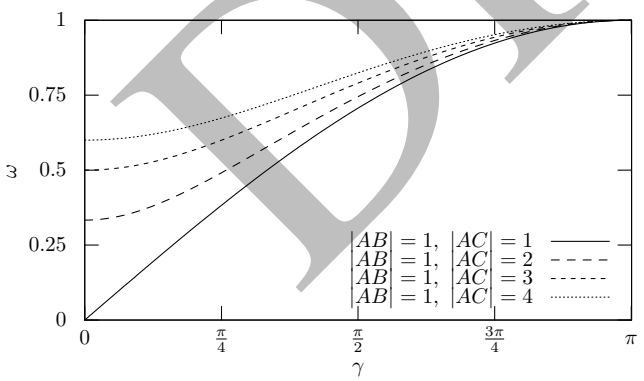

Fig. 3. Divergence ω of two vectors with the same origin varying their angle and relative length.

Unlike cosine distance, our divergence score can be negative if the vectors have different origins and are getting closer, making this measure more suitable than D_C for our aims.

B. DCFL algorithm

Given n clients participating to the FL scheme, at any given communication round each of them produces an update vector $\vec{\delta}$ with origin the model received from the FL server and ending on the model found after a local update step. For each pair (i, j) of clients then we can compute their distance as $\mathrm{d}(\vec{\delta}_i, \vec{\delta}_j)$. Once all the pairwise distances have been computed we use them to determine an appropriate clustering structure $\mathcal{C}$. We use an affinity propagation algorithm [23] to divide the clients in clusters without needing a priori knowledge on the optimal number of clusters.

Due to the possibility of distribution shift and the introduction of new clients, the clustering structure is subject to change. Thus, after each communication round is over, when all the model updates are available, a measure to determine whether the current clustering structure is still valid is required. Since we lack any ground truth relative to the correct clustering structure, an internal validity index is needed. In this work, we adopt the Dunn index [24], which is the ratio between the minimum intercluster distance and the maximum intracluster distance. As shown in Eq. 3 we compute the intercluster distance with single linkage and the intracluster distance as the maximum distance between two update vectors produced by clients belonging to the same cluster.

$$DI = \frac{\min\{\mathrm{d}(\vec{\delta}_a, \vec{\delta}_b) : \forall C_i \in \mathcal{C} \;\; \{a,b\} \nsubseteq C_i\}}{\max\{\mathrm{d}(\vec{\delta}_a, \vec{\delta}_b) : \{a,b\} \subseteq C_i \in \mathcal{C}\}} \tag{3}$$

Whenever $DI < 1$ we have that two pairs of updates $(\vec{\delta}_a, \vec{\delta}_b)$ and $(\vec{\delta}_c, \vec{\delta}_d)$ exist such that the conditions in Eq. 4 are verified.

$$\begin{cases} \mathrm{d}(\vec{\delta}_a, \vec{\delta}_b) < \mathrm{d}(\vec{\delta}_c, \vec{\delta}_d) \\ \nexists \; C_i \in \mathcal{C} : \{a,b\} \subseteq C_i \\ \exists \; C_i \in \mathcal{C} : \{c,d\} \subseteq C_i \end{cases} \tag{4}$$

It is worth noting that in principle it is possible that $a = c$. In such a circumstance, $\vec{\delta}_a$ is closer to updates from other clusters than some from its own, meaning that it could probably be assigned to the other cluster. In general, if DI is below this threshold we can assume that the clustering is not suitable for the current data distribution of the clients. Thus, we use this threshold on the Dunn index to assess whether to perform new clustering operations. If this condition is verified, then the latest update vectors are fed as input to the clustering algorithm to determine a new clustering structure.

Unlike sequential divisive clustering techniques, when a new clustering structure is produced, the members of any given cluster might originate from distinct clusters of the previous structure. Therefore, a new global model for each cluster must be computed and distributed to all cluster members. Many possible strategies can be employed to produce a cluster model, such as majority voting and robust aggregation. Here we decide to compute these models simply as the average of each member client model as shown in Eq. 5.

$$\mathcal{W}_{C_i} = \frac{1}{|C_i|} \sum_{j \in C_i} \mathcal{W}_j \tag{5}$$

Whenever a new client wants to join the FL system, the server needs to provide it with a suitable model. We assume that in the current clustering structure exists a cluster C_j such that its model $\mathcal{W}_{C_j}$ is a better fit for the new client than the model obtained by training a separate model from scratch with local training only, even if the overall data distribution for the members of the selected cluster could be different from that of the new client.

Testing all the cluster models on the client local dataset to find the one which minimizes the loss is unfeasible for a large number of clusters, since each cluster model would need to be transmitted. Instead, we assign the new client to an appropriate cluster by only transmitting two models and one update. Namely, to assign the newcomer $n+1$-th client to a cluster, the server first provides the client an average global model $\mathcal{W}_G = \sum_{C_j \in \mathcal{C}} \mathcal{W}_{C_j} / |\mathcal{C}|$. Then, after a round of local training, we compute the distance of the client update vector $\vec{\delta}_{n+1}$ relative to the existing clusters and add the client to the closest cluster. Finally, the cluster model is transmitted to the client. This is expressed more formally in Eq. 6

$$\mathcal{W}_{n+1} = \mathcal{W}_{C_j} : \quad j = \underset{i}{\operatorname{argmin}} \ \mathrm{d}\left(\vec{\delta}_{C_i}, \vec{\delta}_{n+1}\right) \tag{6}$$

where $\vec{\delta}_{C_i}$ is a vector with origin on the average global model $\mathcal{W}_G$ and pointing to $\mathcal{W}_{C_i}$.

A complete high-level overview of our DCFL algorithm is presented in Algorithm 1.

III. Experimental evaluation

In order to assess the performance of DCFL, we test it on the EMNIST dataset [25]. We simulate a label distribution skew by assigning samples of each label to clients following a Dirichlet distribution with $\alpha = 1$. Moreover, in order to also achieve attribute and preference skew, the images in the dataset of each client are rotated by a random angle, different for each client.

As we are interested in improving the performance of FL setups under highly dynamic conditions, every ten communication rounds we introduce a new client with its own unique data distribution. The training consists in a total of 60 rounds. Each local update entails one epoch of training on the local dataset of each client. Moreover, to account for client data distribution drift, starting from the fifth communication round and every ten rounds after that a client is randomly selected and its data distribution is altered by changing the rotation angle. Differential privacy is also used with $\delta = 10^{-5}$ and $\epsilon = 10$, and gradient norm clipping, which are commonly used parameters for relatively high privacy concerns [3].

In Fig. 4 we compare DCFL against the standard *FedAvg* algorithm and the divisive clustering proposed in [18] (CFL). All the algorithms are tested under the same conditions. We can see that any form of clustering outperforms *FedAvg*, yielding higher accuracy and lower variance for the participating clients. DCFL can also be seen to outperform the competing clustering algorithm, both in terms of overall final accuracy and performance during training.

Algorithm 1 DCFL algorithm

Input: an initial set of clients $\{1, \dots n\}$
Output: A clustering for the federated clients and a model for each cluster

$C_0 \leftarrow \{1, \dots, n\}$ *# assign all clients to* C_0
$\mathcal{W}_{C_0} \leftarrow \mathcal{W}_G \leftarrow$ random initialization
$\mathcal{C} = \{C_0\}$, Models$\leftarrow \{\mathcal{W}_{C_0}\}$
for each communication round **do**
  if client $n+1$ joins **then**
    $\mathcal{W}_{n+1} \leftarrow \mathcal{W}_G$
    $\vec{\delta}_{n+1} \leftarrow$ LocalUpdate$(n+1, \mathcal{W}_G)$
    $k \leftarrow$ FindClosest$(\vec{\delta}_{n+1}$, Models$)$
    $C_k \leftarrow C_k \cup \{n+1\}$, $\mathcal{W}_{n+1} \leftarrow \mathcal{W}_{C_k}$
  end if
  deltas $= \{\}$
  for each cluster $C_j \in \mathcal{C}$ **do**
    for each client $i \in C_j$ **do**
      $\vec{\delta}_i \leftarrow$ LocalUpdate$(i, \mathcal{W}_{C_j})$
      deltas = deltas $\cup \{\vec{\delta}_i\}$
    end for
    update $\mathcal{W}_{C_j}$ with the deltas of the clients in C_j
  end for
  update $\mathcal{W}_G$
  compute the distances between update vectors
  $DI \leftarrow$DunnIndex(distances)
  if $|\mathcal{C}| == 1$ **or** $DI < 1$ **then**
    $\mathcal{C}' \leftarrow$ Clustering(deltas)
    Models $\leftarrow$ GenerateNewModels$(\mathcal{C}, \mathcal{C}'$, Models$)$
    $\mathcal{C} \leftarrow \mathcal{C}'$
  end if
end for
return $\mathcal{C}$, Models

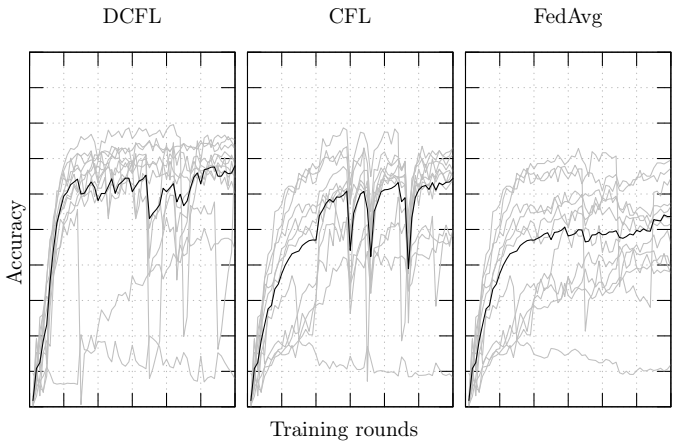

Fig. 4. Comparison of the performance during training of the competing approaches. The accuracy on the dataset of each client is shown in gray, the overall average is shown in black.

In order to quantitatively assess the difference in performance we report in Table I the mean and the standard deviation of the accuracy achieved by the clients at the end of the 60 rounds and the average of these values during the whole training. We can see that, in accordance with the results in Fig. 4, our approach yields a higher accuracy than the alternatives since the first communication rounds. Moreover, the outcome for all the clients is far more homogeneous, with half and one third of the standard deviation of the other two approaches respectively, confirming the soundness of our approach.

TABLE I
COMPARISON OF THE PROPOSED APPROACH TO BASELINE IN TERMS OF AVERAGE ACCURACY AND CLIENT VARIANCE.

Technique	**Final**		**Average**	
	Avg. accuracy	Std. dev	Avg. accuracy	Std. dev
DCFL	0.72	0.04	0.61	0.08
CFL [18]	0.64	0.08	0.50	0.12
FedAvg	0.54	0.12	0.44	0.13

On top of assessing the performance of the DCFL algorithm in this highly dynamic setting, we are interested in confirming that it does not harm performance under more static conditions. To evaluate the impact of the differences in the dynamic non-IID setting, in Fig. 5 we show multiple combinations of simulation parameters. Tests were performed in the following settings:

- No new clients, no drift, no differential privacy;
- New clients, drift, no differential privacy;
- New clients, no drift, differential privacy;
- No new clients, drift, differential privacy.

DCFL is shown to always match if not outperform the alternative algorithms in all the tested settings.

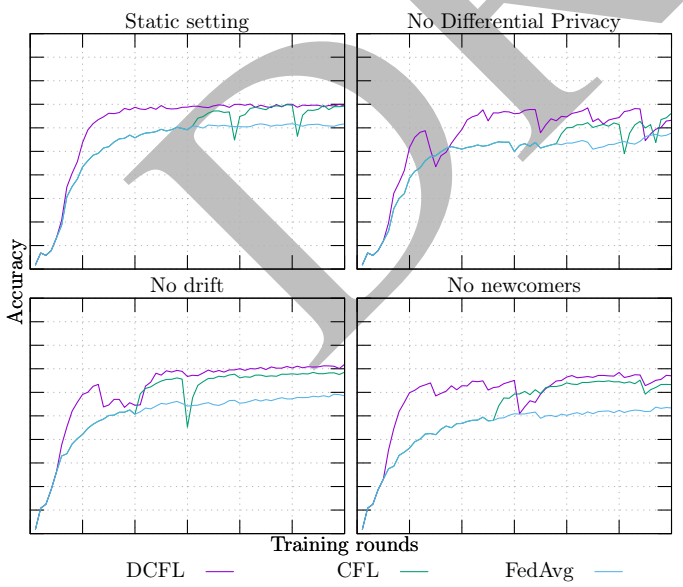

Fig. 5. Difference in accuracy during training of all the approaches varying the test conditions.

We are also interested in assessing how much of the performance improvements were due to the dynamic clustering approach and how much were due to the improved distance metric. Thus, we repeated the experiments in the dynamic setting using different distance metrics in place of the proposed one, namely: the cosine similarity, the euclidean distance, and our divergence metric. The results of these experiments are reported in Fig. 6 and Table II. The Euclidean distance gave the worse results, barely improving over the performance of vanilla *FedAvg*. As the training goes on the distance between the clusters became too large compared to the client updates, and the clustering structure could change anymore to accommodate changes in distributions. Using our divergence measure alone already somewhat improved the final accuracy. This measure, however, does not depend at all from the actual distance of the update vectors, only their direction, and thus sometimes clients that belong to different clusters were mistakenly joined as can be seen by the frequent drastic changes in accuracy experienced by the individual clients shown in Fig. 6. Using the standard cosine distance showed more stability in the performance experienced by the clients compared to our divergence measure, but the overall performance was quite similar and was still outperformed by our distance metric. The complete distance metric accounts for these shortcomings of the individual measures and yielded better results.

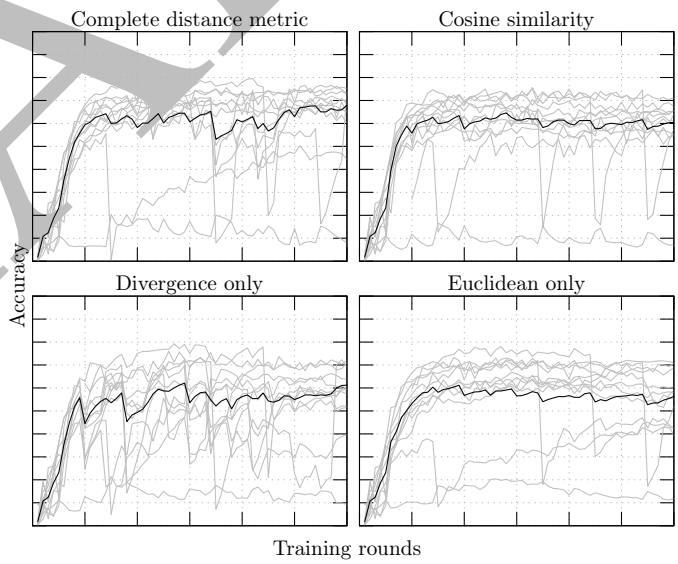

Fig. 6. Performance comparison of DCFL using the proposed distance metric compared to solely the divergence, the euclidean distance, and the cosine distance.

IV. CONCLUSIONS

In this work, we have presented DCFL, a federated learning framework to handle clients with heavily non-IID data distributions, distribution shifts, noisy updates caused by local differential privacy, and staggered joining of the clients to the FL setup. The proposed approach is based on a dynamic clustering algorithm and a novel metric used to assess the distance between the model updates provided by the clients.

TABLE II
PERFORMANCE OF DCFL VARYING THE ADOPTED DISTANCE METRIC

Distance	Final		Average	
	Avg. accuracy	Std. dev	Avg. accuracy	Std. dev
Complete	0.72	0.04	0.61	0.08
Divergence	0.61	0.09	0.51	0.12
Cosine	0.60	0.07	0.56	0.09
Euclidean	0.56	0.12	0.52	0.14

We evaluated DCFL on the EMNIST dataset showing its superiority compared with *FedAvg* and divisive clustered federated learning under a wide range of experimental conditions. Our evaluation also showed the advantages given by our distance metric compared to the ones commonly used to cluster federated clients.

Future works will perform tests on other standard datasets and will address settings where only a portion of the federated clients participate in each training round. Moreover, the impact of different privacy budgets will be investigated as well as the computation and communication overhead of the clustering scheme.

REFERENCES

[1] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond," *IEEE Internet of Things Journal*, vol. 8, pp. 5476–5497, Apr. 2021.

[2] R. Presotto, G. Civitarese, and C. Bettini, "Preliminary Results on Sensitive Data Leakage in Federated Human Activity Recognition," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 304–309, Mar. 2022.

[3] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[4] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation* (M. Agrawal, D. Du, Z. Duan, and A. Li, eds.), (Berlin, Heidelberg), pp. 1–19, Springer Berlin Heidelberg, 2008.

[5] Z. Lian, W. Wang, and C. Su, "COFEL: Communication-Efficient and Optimized Federated Learning with Local Differential Privacy," in *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021.

[6] Q. Yang, "Toward Responsible AI: An Overview of Federated Learning for User-centered Privacy-preserving Computing," *ACM Transactions on Interactive Intelligent Systems*, vol. 11, pp. 32:1–32:22, Oct. 2021.

[7] F. Concone, C. Ferdico, G. Lo Re, and M. Morana, "A Federated Learning Approach for Distributed Human activity recognition," in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 269–274, 2022.

[8] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the Air: Secure Federated Learning for UAV-Assisted Crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1055–1069, 2021.

[9] M. Cho and A. Mashhadi, "Caring Without Sharing: A Federated Learning Crowdsensing Framework for Diversifying Representation of Cities," in *Mobile and Ubiquitous Systems: Computing, Networking and Services* (T. Hara and H. Yamaguchi, eds.), (Cham), pp. 601–616, Springer International Publishing, 2022.

[10] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. Lo Re, "IncentMe: Effective Mechanism Design to Stimulate Crowdsensing Participants with Uncertain Mobility," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1571–1584, 2019.

[11] Y. Jiang, R. Cong, C. Shu, A. Yang, Z. Zhao, and G. Min, "Federated Learning Based Mobile Crowd Sensing with Unreliable User Data," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 320–327, 2020.

[12] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.

[13] S. Zehtabian, S. Khodadadeh, L. Bölöni, and D. Turgut, "Privacy-Preserving Learning of Human Activity Predictors in Smart Environments," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, 2021.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR, 20–22 Apr 2017.

[15] Y. Yeganeh, A. Farshad, N. Navab, and S. Albarqouni, "Inverse Distance Aggregation for Federated Learning with Non-IID Data," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning* (S. Albarqouni, S. Bakas, K. Kamnitsas, M. J. Cardoso, B. Landman, W. Li, F. Milletari, N. Rieke, H. Roth, D. Xu, and Z. Xu, eds.), Lecture Notes in Computer Science, (Cham), pp. 150–159, Springer International Publishing, 2020.

[16] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," *arXiv preprint arXiv:1806.00582*, 2018.

[17] P. Tian, W. Liao, W. Yu, and E. Blasch, "WSCC: A Weight Similarity Based Client Clustering Approach for Non-IID Federated Learning," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[18] F. Sattler, K.-R. Müller, and W. Samek, "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 3710–3722, Aug. 2021.

[19] Y. Sun, H. Esaki, and H. Ochiai, "Adaptive Intrusion Detection in the Networking of Large-Scale LANs With Segmented Federated Learning," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 102–112, 2021.

[20] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the Byzantine Robustness of Clustered Federated Learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8861–8865, May 2020. ISSN: 2379-190X.

[21] C. Briggs, Z. Fan, and P. 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)*, pp. 1–9, July 2020. ISSN: 2161-4407.

[22] M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, Y. Tan, and A. Ren, "Flexible Clustered Federated Learning for Client-Level Data Distribution Shift," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, pp. 2661–2674, Nov. 2022.

[23] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[24] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[25] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017.