ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230451

$\mathcal{MFC}$: A $\mathcal{M}$ultishot Approach to $\mathcal{F}$ederated Data $\mathcal{C}$lustering

Jaglike Makkar[a;*], **Bhumika** [a], **Shweta Jain**[a] **and Shivam Gupta**[a;**]

[a]Indian Institute of Technology Ropar, India

Abstract. The work explores the federated data clustering problem. The primary goal is to perform k-means clustering of data distributed over multiple clients while preserving privacy during an exchange with the central server. Existing solutions to unsupervised federated data clustering are either computationally challenging or effective only in heterogeneous regimes, i.e., when the number of clusters per client (k_z) is less than the total number of clusters (k) (specifically, $k_z \leq \sqrt{k}$). Moreover, existing one-shot approaches assume that the information about k_z is available for each client. In this paper, we propose two multi-shot approaches which we call $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$, that perform well on both heterogeneous and non-heterogeneous regimes, i.e., are independent of the underlying client data distribution. Both $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$ stand out as they do not rely on prior knowledge about k_z. We theoretically bound the closeness of the local centers obtained by $\mathcal{MFC}$ to that of the optimal global centers and prove that under some well-separability assumption, the centers will be close enough. $\mathcal{MFC}_{\mathcal{H}}$ improvises $\mathcal{MFC}$ by only sharing a single cluster center from each client, thus ensuring more privacy. Our theoretical analysis shows that when at least $O(k^2 \log k)$ clients are involved, centers obtained by $\mathcal{MFC}_{\mathcal{H}}$ will closely approximate optimal global centers. Experiments on synthetic and real-world datasets validate the proposed approaches' efficacy showcasing lower objective costs in non-heterogeneous regimes while having comparable performance in heterogeneous regimes. In addition, as a byproduct $\mathcal{MFC}$ exhibits higher device-level fairness in terms of the individual objective cost compared to existing state-of-the-art algorithms. The code is publicly available at https://github.com/shivi98g/MFC.
Keywords: Machine Learning, Clustering, Federated Learning.

1 Introduction

Federated learning (FL) is a machine learning paradigm introduced by Google Inc. in 2016 [38]. The approach enables multiple devices (or clients) to contribute jointly to model training without sharing sensitive data [59]. In client-server federated learning, model training is delegated to different clients that process their local data, and solely model updates are communicated with a central authority (server). The resulting global model is obtained by consolidating local updates and is shared back with clients [38]. Due to its enhanced privacy and security, the approach has been adopted widely in recommender systems [56], next-word prediction [47], and healthcare [1].

Federated learning is a well-studied problem in supervised learning settings [24, 53, 58] with applications ranging from fraud detection [57], credit risk assessment [28], and speech recognition [15]. However, while some applications involve generating labels implicitly (using user interaction), several others generate a large amount of unlabelled data. The amount of unlabelled data is exponentially growing, and explicit (or manual) labeling is tedious and costly for users. This leads to the potential use of FL in applications such as healthcare and medical diagnosis [1]. Motivated by these applications, in this work, we focus on unsupervised learning in federated setting.

In unsupervised learning (UL), clustering is a powerful tool that finds inherent structures in data and identifies them as clusters [54]. The existing literature investigates various clustering approaches, such as hierarchical [39], density [8], and center-based methods [2, 33]. The current work will focus on center-based clustering due to its computational efficiency and easy scalability [5]. Federated clustering can help cluster patient data across multiple hospitals for better treatment [1], reduce fraudulent financial activities across banks [57], and improve customer market segmentation for enhanced recommendations [56]. As a running example, let us consider the potential of federated clustering (FC) in customer market segmentation. FC allows business organizations to analyze customer bases from multiple sources while preserving privacy, resulting in more accurate and reliable clusters. These clusters can help better understand their customers and provide them with more personalized products and policies [42, 45, 17].

In general, clustering problems in FL can be divided into two types, client clustering [12, 26] and data clustering [16, 31]. The goal of client clustering is to partition the clients into clusters based on the heterogeneity of the data available at clients so as to perform model training while achieving communication efficiency. We highlight that our work focuses on data clustering which aims to cluster the data points present across the clients in the federated network such that traditional clustering loss is minimized. Just like any FL setting, clients are not allowed to share their data points with the server but can share limited information such as model updates. In center-based clustering, clients primarily share their local centers with the server [16]. The server then aggregates these centers to find the global cluster centers that best represent the data available at the local clients.

The work in federated data clustering is in the formative stages. The initial attempt to explore the partitioning of data points scattered across multiple devices into k clusters is provided in [16]. The authors propose an algorithm (k-FED) based on the famous offline Llyod's heuristic (usually named offline k-means). The algorithm uses the Llyod heuristic locally on each device and then computes the final clustering in only one round of communication. Though it is communication efficient, the performance of k-FED is highly dependent on the data distribution across clients. We further show that k-FED

* Jaglike and Bhumika contributed equally in the work.
** {2019csb1092,2019csb1152,shwetajain,shivam.20csz0004}@iitrpr.ac.in

suffers significant fluctuations in the clustering cost across different client devices. The subsequent work by [13] also suffers similar challenges while having high computational complexity in some instances. To tackle these issues, we propose our algorithm, called Multishot Federated Clustering ($\mathcal{MFC}$), which performs well in both heterogeneous and non-heterogeneous settings and utilizes multiple rounds of interaction to achieve much better cost as opposed to the existing works. In particular, our algorithm satisfies the following properties:

- **Independent of Data Distribution:** $\mathcal{MFC}$ does not rely on the data distribution across clients, unlike k-FED. This is possible because $\mathcal{MFC}$ ensures that each cluster's gaussian distribution gets represented at the global level with a high probability. Further, if any of the initial global centers does not correctly represent a Gaussian distribution, there is a high probability of finding a correct representation by using multiple shots.
- **Multishot Approach:** $\mathcal{MFC}$ intends to utilize the benefits of the multishot approach. While a single-shot method might be more efficient in terms of communication costs, the multi-shot algorithm is able to achieve both, lower clustering cost and fairness. Further, it is observed experimentally that the number of iterations required to *converge* to a low cost are countably few.
- **Theoretical Guarantees:** We show that under well separability assumptions (similar to existing literature [3, 16]), we have, $||\theta_i^{(z)} - \mu(T_i)||_2 \leq c\sqrt{k}\frac{||X-C||_2}{\sqrt{n_i^{(z)}}}$, where c is a positive constant. Here, $\theta_i^{(z)}$ denotes the local cluster centers, $\mu(T_i)$ denotes the global optimal cluster centers, X represents the complete dataset matrix, C represents the global center matrix and $n_i^{(z)}$ represents the number of points of cluster i on device z. The result shows that the local and global centers are significantly close in $\mathcal{MFC}$, $\mathcal{MFC}_{\mathcal{H}}$.
- **Improved Fairness:** $\mathcal{MFC}$ not only minimizes the overall clustering cost but also promotes cost fairness across devices. By sending the maximum-cost cluster center from each device to the server, even if a device initially suffers high cost, it will soon reduce and become more equitable with other devices. We validate this experimentally on real-world datasets.
- **Addition/Removal of new devices:** When a new device joins the network, either to replace failed device or as a new participant, its cluster assignment can be computed independently without involving other devices. The device can receive the global centers and determine its clustering assignment. In future shots (or iterations), the device will send its highest-cost cluster to update the model.

2 Related Work

A considerable amount of literature is available on supervised federated learning [24, 53, 58, 43, 18, 27, 34, 6, 44, 23]. The research on unsupervised learning (UL) is still in its early stages [13, 16] and gaining attention due to the increasing availability of unlabelled data. As a result, many experts consider it to be the next big frontier in artificial intelligence [14]. Clustering is one of the powerful UL paradigms. It deals with partitioning a set of data points into different groups (called clusters) such that data points within the same cluster are more similar than they are to points in other clusters [54]. We categorize the existing literature into four parts. Initially, we discuss various prevalent centralized clustering approaches, focusing on the centroid-based algorithms. Due to the close resemblance between federated and distributed clustering, next, we will delve into current advancements in parallel and distributed clustering. Towards the end, when it comes to clustering in FL, we explore existing literature in two distinct methods: client clustering and data clustering.

Centralized Clustering: Centralized clustering involves storing and processing all data in a single machine. The algorithms operate on entire data at once and group them into clusters. It is beneficial in scenarios when the number of data points is relatively small or computational resources are not a limiting factor. Popular centralized clustering methods involve centroid-based algorithms (the most popular being, k-means [2, 33], k-median [10], and k-center [22]), distribution-based (such as gaussian mixture models [60]), density-based (such as DBSCAN [29]) and hierarchical clustering algorithms [39]. In this work, we primarily focus on center-based clustering techniques. The general idea of these algorithms is to represent each cluster via a center and assigns each data point to its nearest center. The closeness to the center is measured by different norms, for example, for k-means L_2 norm, for k-median, L_1 norm and for k-center L_∞ norm. Our algorithm primarily focuses on k-means.

Distributed Clustering: To handle data that is large enough not to be stored or computed by a single machine, distributed environments are commonly used [21]. In distributed clustering, the data is partitioned into subsets and is processed simultaneously across multiple machines. The results from each machine are subsequently combined to form global clustering of the entire data. Not only distributed clustering enhances scalability and speed, but it also mitigates single-point failures in real-world deployments. Work is also carried out in decentralized clustering to remove dependence on a single central authority leading to a fully connected network [40]. [61] and [19] discuss the parallel implementations of the k-means. There are works on parallel implementations of other algorithms such as k-medoids [46], hierarchical [41], and DBSCAN [51]. However, data exchange in these environments can raise privacy concerns among users, hence there is a strong need to develop privacy preserving algorithms i.e, federated setting.

Federated Data Clustering: Data clustering in a federated setting poses several challenges that are not present in traditional distributed clustering. Some of the main challenges are data privacy and security, heterogeneity of data and devices, limited communication between devices, synchronization, consensus, handling device addition and dropout, etc. Some of the key related works in this area include [16], and [13], which propose a federated clustering algorithm that can handle heterogeneous data and devices in a distributed environment. [13] can handle both IID (Independent and Identically Distributed) and non-IID data. However, in most of these works, the clustering performance heavily depends on the data distribution across devices. They also do not leverage the benefit of multiple rounds of communication between the server and devices to find a better solution.

A parallel line of work in privacy-preserving distributed clustering is achieved using cryptographic techniques while still allowing for accurate clustering [31] . In SecFC [31], the clients perform local center updates and share encrypted distance vectors using lagrange encoding back to the server. The server is then responsible for gathering all secret distance codes from the clients and carrying out the following iteration updates. Though the algorithm leverages the benefits of encryption-decryption to protect data privacy, such methods require significant computation overhead and communication costs. The sharing of distance vectors can hinder the scalability of the technique. A different line of work includes the work by [55] that attempts to achieve global cluster centers by generating synthetic data at server instead of using original data. Similarly [32] draws inspiration from differential privacy which is quite different from our work.

Federated Client Clustering: In federated client clustering, clients are clustered together to intelligently choose a subset of clients for

the client update step. This direction of work does not deal with the clustering of datapoints from each client but rather involves the clustering of clients themselves [26, 25, 52, 35] . Such client clustering methods lead to sampling lower number of clients and achieving high accuracy of the global model. We emphasize that all these methods still solve the classification problem in a federated setting.

3 Preliminaries

We consider a federated setting in which $[Z]$ devices (or clients) are involved, and each device (denoted by $z \in [Z]$) wants to partition their local data $X^{(z)} \subseteq \mathbb{R}^d$ into k sets (called *clusters*). We further denote the set of all datapoints by $X = \cup_{z\in[Z]} X^{(z)}$ and i^{th} datapoint/row of X by X_i. To address the federated clustering problem, algorithmic designers aim to produce a set of global centers μ=$\{\mu_j\}_{j=1}^k$ that best represent each device's local data $X^{(z)}$. It should be further noted that the points with a particular client may not come from all the k centers. We denote the number of clusters at a device z by $k_z \leq k$. Furthermore, to learn these global centers, each device z will undergo local training to find a local set of k centers, denoted by $\theta^{(z)}$=$\{\theta_j^{(z)}\}_{j=1}^k$, and an assignment function $\phi^{(z)} : X^{(z)} \to \mu$ that maps each data point in $X^{(z)}$ to the closest global center. Also, let $||\cdot||_2$ denote the L_2 norm and $\mathbb{I}(\cdot)$ denote the indicator function. The goal of federated k-means is to find the global centers μ to minimize:

Definition 1 (Clustering Cost)**.** *The cost of k-means federated clustering with respect to the data points X and centers μ with $\phi^{(z)}$ as the assignment function at each client is given by:*

$$\sum_{z\in[Z]} \sum_{x\in X^{(z)}} \sum_{\mu_j\in\mu} \mathbb{I}(\phi^{(z)}(x) = \mu_j)\ ||x - \mu_j||_2^2$$

The above cost is primarily the k-means clustering cost computed by calculating the distance of each point to the global centers achieved. It should be noted that the local data accumulated over time on different devices (or clients) may follow identical probability distributions or originate from entirely distinct distributions (i.e., be heterogeneous). Such distributions have been defined in previous works in terms of the heterogeneity of data. A network is considered heterogeneous if the number of clusters on a device $k_z \leq \sqrt{k}$ for all $z \in [Z]$. Our work attempts to establish a generalized algorithm and proof system that does not solicit certain distribution patterns of data. In contrast, a distribution is said to be homogeneous if $k_z = k$.

Considerable research has studied center separation guarantees for clustering problems across various data distributions [3, 36, 9]. If the data originates from a mixture of k Gaussian one can apply the algorithm in [3] which clusters all data points accurately with high probability under some separability assumptions. In this paper, we also use the algorithm provided by [3] and explain it here for completeness. The algorithm begins by projecting the device's local data onto singular vectors [1]. The initial set of cluster centers is intelligently initialized, after which a standard Lloyd algorithm [33] for k-means is executed. The Algorithm 1, will act as a building block for our federated $\mathcal{MFC}$ that works for both heterogeneous and homogeneous settings while providing theoretical guarantees on global centers obtained. We now discuss $\mathcal{MFC}$ in detail.

[1] The Singular Value Decomposition (SVD) of a matrix A is a factorization represented by $A = U \sum V^T$, where the columns of V are referred to as right-singular vectors (or simply singular vectors).

Algorithm 1: Local k-means [3] (centralized)

Input: The matrix X and number of clusters k
Part 1: Finding initial centers: Project X onto the subspace spanned by the top k singular vectors. Run any standard approximation algorithm for the k-means problem on the projected matrix $\hat{X}$, and obtain k centers $\{\theta_1, \theta_2, \ldots, \theta_k\}$.
Part 2: Set
$S_r \leftarrow \{\hat{X}_i : ||\hat{X}_i - \theta_r||_2 \leq \frac{1}{3}||\hat{X}_i - \theta_s||_2, \forall s \in [k]\}\ \forall r \in [k]$ and $\theta_r \leftarrow \theta(S_r), \phi(x) \leftarrow \theta_r\ \forall x \in S_r$.
Part 3: Repeatedly run Lloyd steps until convergence: Set
$U_r \leftarrow \{\hat{X}_i : ||\hat{X}_i - \theta_r||_2 \leq ||\hat{X}_i - \theta_s||_2, \forall s \in [k]\}\ \forall r \in [k]$ and $\theta_r \leftarrow \theta(U_r), \phi(x) \leftarrow \theta_r\ \forall x \in U_r$
Return: Cluster assignment ϕ and their means
$\theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$

4 Multishot Federated Clustering ($\mathcal{MFC}$)

$\mathcal{MFC}$ first runs Algorithm 1 on local data of each client $z \in [Z]$ to obtain a set of k initial local centers denoted by $\theta^{(z)}$. Each client then transmits its respective set to the server and the server applies the Lloyd k-means algorithm [33] on collected set S. The obtained global centers μ are sent back to clients for further update.

After the initial handshake between the client and server, they engage in multiple rounds of communication as follows- Clients use the global centers to update their local assignment functions $\phi^{(z)}$ by re-assigning each data point $x \in X^{(z)}$ to the nearest center. The local cluster centers are updated by taking the mean of points assigned to each center. The client sends the local cluster center $\theta_{\text{max}}^{(z)}$ back with the maximum clustering cost on local data. Conversely, when the server receives the maximum cost centers, it recalculates the global centers μ by using Lloyd's k-means on the previous global centers and $\theta_{\text{max}}^{(z)}$ from all clients. After finding the updated global centers, they are returned to the clients. This iterative process is repeated for a few rounds until convergence is reached. The complete algorithm for $\mathcal{MFC}$ is described in Algorithm 2. We now provide theoretical bounds on obtained global cluster centers.

5 Theoretical Results for $\mathcal{MFC}$

This section lays the theoretical foundation and key results that form the backbone of $\mathcal{MFC}$. We prove the correctness of our algorithm by showing that the centers obtained from $\mathcal{MFC}$ are *close* to the oracle clustering. We do so by providing a series of lemmas to prove our main Theorem 1 that bounds the distance between these centers.

5.1 Assumptions

Before presenting the theoretical proofs, we carefully outline the assumptions on which our analysis is based. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ denote the optimal clustering of datapoints X with means $\{\mu(T_1), \mu(T_2), \ldots, \mu(T_k)\}$. Let n_i denote the cardinality of T_i. Let $T_i^{(z)} \subseteq T_i$ denote the set of datapoints of cluster T_i that are available on client z. Let $n_i^{(z)}$ denote the number of points in $T_i^{(z)}$. Let $X^{(z)} = \cup_{i=1}^k T_i^{(z)}$ denote the set of datapoints at client z. Further, we will denote the dataset by the matrix $X \in \mathbb{R}^{n\times d}$ with the i^{th} row representing the datapoint X_i. Let us further define a mapping $\gamma : X \to [k]$ which associates each data point to their optimal cluster, i.e. $\gamma(X_i) = j$, iff $X_i \in T_j$. Similarly, let us define $\gamma^{(z)} : X^{(z)} \to [k]$ such that $\gamma^{(z)}(X_i^{(z)}) = j$, iff $X_i^{(z)} \in T_j^{(z)}$.

Algorithm 2: Multishot Federated Clustering ($\mathcal{MFC}$)

Client initially executes:

1 On each client $z \in [Z]$, run Algorithm 1 with local data $X^{(z)}$ and k to find local cluster centers $\theta^{(z)}$.
2 All clients $z \in [Z]$ shares center set $\theta^{(z)}$ with server.

Server initially executes:

1 Receives set of centers $\theta^{(z)}$ from all devices $z \in [Z]$, to construct $S = \theta^{(1)} \cup \theta^{(2)} \ldots \cup \theta^{(z)}$.
2 Apply Llyod k-means clustering [33] on set S, to find k global centers $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$.
3 Sends back global centers μ to all clients $z \in [Z]$ for further local training.

Client updates:

1 All clients $z \in [Z]$ receive global centers μ from the server.
2 Each client z updates their local assignments function $\phi^{(z)}$ according to μ, i.e., $\forall x \in \hat{X}^{(z)}$, $\phi^{(z)}(x) \leftarrow \{\mu_i : ||x - \mu_i||_2 \leq ||x - \mu_j||_2 \; \forall \mu_j\}$.
3 Updating local cluster sets $\theta^{(z)}$ by computing the mean of cluster assignments.
4 Sends back local cluster center suffering maximum clustering cost (Definition 1) to server **(i.e., Server updates)**. Let us denote it using $\theta^{(z)}_{\max}$.

$$\theta^{(z)}_{\max} = \underset{\theta_i}{\operatorname{argmax}} \sum_{x \in X^{(z)}} \mathbb{I}(\phi(x) = \theta_i)||x - \phi(x)||_2^2$$

Server updates:

1 Receives maximum cost centers $\theta^{(z)}_{\max}$ from all $z \in [Z]$.
2 Update $S = S \cup \{\theta^{(1)}_{\max}, \ldots, \theta^{(k)}_{\max}\}$
3 Apply Llyod k-means clustering on the S, to find k global centers $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$.
4 Sends back global centers μ to all clients $z \in [Z]$ for further local training **(i.e., Client updates)**.

Let $C \in \mathbb{R}^{n \times d}$ denote the matrix such that each i^{th} row correspond to the optimal center of i^{th} data point, i.e. $C_i = \mu(T_{\gamma(X_i)})$. Let $C^{(z)}$ matrix be the corresponding C matrix but defined only for datapoints in $T^{(z)}$ with centers defined on local datasets i.e., $\{\mu(T^{(z)}_1), \mu(T^{(z)}_2), \ldots, \mu(T^{(z)}_k)\}$.

A 1. *The non-empty subset of the datapoints on device z belonging to the global cluster T_i, denoted by $T^{(z)}_i$ is sufficiently large. That is, there exists a sufficiently small constant $0 < \epsilon < 1$ such that $n^{(z)}_i \geq \frac{8\sigma_i^2}{\epsilon^2}\left(ln(\frac{1}{\delta}) + \frac{1}{4}\right) \; \forall \; i$ for a given $0 < \delta < 1$. We will clarify the requirement on sufficiently small in Lemma 4.*

Next, we define the notion of well-separability of clusters, and such assumption is a standard in the clustering literature [3, 16].

Definition 2 (Well-separability). *A pair of target clusters T_i and T_j are said to be well separated if they satisfy*

$$||\mu(T_i) - \mu(T_j)||_2 \geq p\sqrt{k}||X - C||_2 \left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}}\right)$$

where p is a large constant and n_i is number of points in i^{th} cluster.

A 2. *The centers of the oracle clustering μ_i are well separated.*

We first provide the important results along with their proofs, followed by the magnitude of the center separation in Lemma 4.

5.2 Proofs

In the first result, we show that the centers from the localized datasets at each device (if optimal clustering would have been known) is close to that of global centers. For this, we use vector Bernstein inequality provided in [30]. We restate the lemma here for completeness.

Lemma 1. *(Vector Bernstein Lemma [30]) Let $x_1, x_2, \ldots, x_n$ be d-dimensional independent vector-valued random variables s.t. $\mathbb{E}[x_i] = 0$, $||x_i||_2 \leq \mu$ and $\mathbb{E}[||x_i||^2] \leq \sigma^2$, then $\forall \epsilon : 0 < \epsilon < \frac{\sigma^2}{\mu}$, we have*

$$\mathbb{P}\left(\left|\left|\frac{1}{n}\sum_{i=1}^{n} x_i\right|\right|_2 \geq \epsilon\right) \leq exp\left(-n\frac{\epsilon^2}{8\sigma^2} + \frac{1}{4}\right)$$

Using the above Lemma on each data point present in $T^{(z)}_i$, and defining $x_j = T^{(z)}_{ij}$, where $T^{(z)}_{ij}$ is the j^{th} data point in $T^{(z)}_i$, we get:

Lemma 2. *Let us denote σ_i^2 to be an upper bound on the standard deviation of i^{th} global cluster i.e. $\mathbb{E}[||T^{(z)}_{ij} - \mu(T_i)||^2] \leq \sigma_i^2$. If $n^{(z)}_i \geq \frac{8\sigma_i^2}{\epsilon^2}\left(\ln(\frac{1}{\delta}) + \frac{1}{4}\right)$, then $||\mu(T^{(z)}_i) - \mu(T_i)||_2 \leq \epsilon$ with probability at least $1 - \delta$.*

Proof. Substitute $exp\left(-n\frac{\epsilon^2}{8\sigma^2} + \frac{1}{4}\right) \leq \delta$ in Lemma 1. □

Lemma 3 (Dennis et. al [16])**.** $||X^{(z)} - C^{(z)}||_2 \leq 2\sqrt{k}||X - C||_2$

Now, further let $n^{(z)}_{max} = \max_i\left(n^{(z)}_i\right)$ and to have sufficiently small requirement of ϵ consider α such that $\epsilon \leq \alpha\sqrt{k}||X - C||_2\left(\frac{1}{\sqrt{n^{(z)}_{max}}}\right) \leq \frac{\alpha}{2}\sqrt{k}||X - C||_2\left(\frac{1}{\sqrt{n^{(z)}_i}} + \frac{1}{\sqrt{n^{(z)}_j}}\right)$.

Lemma 4. *Let $\gamma = \frac{8\sigma_i^2}{n_i\epsilon^2}\left(\ln\left(\frac{1}{\delta}\right) + \frac{1}{4}\right)$ and if each pair of global clusters are well separated, i.e.*

$$||\mu(T_i) - \mu(T_j)||_2 \geq p\sqrt{k}||X - C||_2\left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}}\right)$$

and p' is chosen such that $p' \geq (p\sqrt{\gamma} - \alpha)$, then the means of the local subsets of the clusters ($\mu(T^{(z)}_i)$) are also well separated with probability at least $1 - \delta$. Here γ provides an upper bound on fraction of points present on any client z i.e, $\gamma \leq n^{(z)}_i/n_i$.

Proof. Using triangular inequality and Lemma 1,

$$\begin{aligned}||\mu(T_i) - \mu(T_j)||_2 &\leq ||\mu(T^{(z)}_i) - \mu(T_i)||_2 + ||\mu(T^{(z)}_j) - \mu(T_j)||_2 \\ &\quad + ||\mu(T^{(z)}_i) - \mu(T^{(z)}_j)||_2 \\ &\leq 2\epsilon + ||\mu(T^{(z)}_i) - \mu(T^{(z)}_j)||_2 \end{aligned} \tag{1}$$

As each pair of global clusters is well separated, we have

$$\begin{aligned}||\mu(T^{(z)}_i) - \mu(T^{(z)}_j)||_2 &\geq p\sqrt{k}||X - C||_2\left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}}\right) - 2\epsilon \\ &\geq p\sqrt{k}||X - C||_2\sqrt{\gamma}\left(\frac{1}{\sqrt{n^{(z)}_i}} + \frac{1}{\sqrt{n^{(z)}_j}}\right) - 2\epsilon \\ &\geq (p\sqrt{\gamma} - \alpha)\sqrt{k}||X - C||_2\left(\frac{1}{\sqrt{n^{(z)}_i}} + \frac{1}{\sqrt{n^{(z)}_j}}\right)\end{aligned}$$

Using Lemma 3 and sufficiently small requirement of ϵ i.e, the fact that $p' \geq (p\sqrt{\gamma} - \alpha)$, the lemma follows. □

Lemma 5 (Awasthi-Sheffet [3])**.** *If each pair of local subsets of clusters are well separated, then after performing Algorithm 1,*

$$||\theta_i^{(z)} - \mu(T_i^{(z)})||_2 \leq \frac{25}{p'} \frac{||X^{(z)} - C^{(z)}||_2}{\sqrt{n_i^{(z)}}}$$

It should be noted that the above holds only when well-separability is upheld. Specifically, in Algorithm 1, we create k clusters, while in reality, the data points on a device may come from only $k_z (\leq k)$ clusters. But we can observe that if well-separability of the obtained centers will be violated, that will be the case only when both of these centers belong to the same cluster. In fact, the multiple centers, if obtained, will be close to the center that would have been obtained if $k = k_z$. It can be shown by using CLT again, by considering the two centers to be two different sample means of the same population (i.e. target global cluster).

To account for this, we introduce a mapping $\beta^{(z)} : [k] \to [k]$ which essentially maps the index of the local cluster center obtained by Algorithm 1 to the index of the closest global center. Thus, we have $\beta^{(z)}(r) = \text{argmin}_{i \in [k]} ||\theta_r^{(z)} - \mu(T_i^{(z)})||_2$. Thus, as a direct consequence of Lemma 5, we get

$$||\theta_r^{(z)} - \mu(T^{(z)}_{\beta^{(z)}(r)})||_2 \leq \frac{25}{p'} \frac{||X^{(z)} - C^{(z)}||}{\sqrt{n^{(z)}_{\beta^{(z)}(r)}}} \tag{2}$$

Theorem 1 (Main Theorem)**.** *Assuming well separability holds, we have*

$$||\theta_r^{(z)} - \mu(T_{\beta^{(z)}(r)})||_2 \leq c\sqrt{k} \frac{||X - C||_2}{\sqrt{n^{(z)}_{\beta^{(z)}(r)}}},$$

where c is a positive constant.

Proof. We prove this theorem in two steps. Firstly, using Equation 2 and Lemma 3, we have

$$||\theta_r^{(z)} - \mu(T^{(z)}_{\beta^{(z)}(r)})||_2 \leq \frac{50\sqrt{k}}{p'} \frac{||X - C||_2}{\sqrt{n^{(z)}_{\beta^{(z)}(r)}}} \tag{3}$$

Now, $||\theta_r^{(z)} - \mu(T_{\beta^{(z)}(r)})||_2 \leq ||\theta_r^{(z)} - \mu(T^{(z)}_{\beta^{(z)}(r)})||_2 + ||\mu(T^{(z)}_{\beta^{(z)}(r)}) - \mu(T_{\beta^{(z)}(r)})||_2$ (using triangular inequality)

$$\leq \frac{50\sqrt{k}}{p'} \frac{||X - C||_2}{\sqrt{n^{(z)}_{\beta^{(z)}(r)}}} + \epsilon \quad \text{(Using Equation 3 and Lemma 2)}$$

Substituting $c = \frac{50}{p'} + \alpha$, we get the required result. □

Beyond this, the multi-shot iterations only perform a Lloyd's heuristic and thus lowers the bound of this distance. In $\mathcal{MFC}$, all the clients shares all the k centers with the server. We will now describe a heuristic approach that shares only one center. This algorithm offers communication efficiency, particularly for large values of k and a high number of clients, while also providing increased privacy. We further show that if there are enough clients available and each client uniformly selects every data point from each distribution at random, then $\mathcal{MFC}_{\mathcal{H}}$ holds similar guarantees to that of $\mathcal{MFC}$.

6 $\mathcal{MFC}_{\mathcal{H}}$: A Heuristic approach

The only change that we do in $\mathcal{MFC}_{\mathcal{H}}$ is that instead of sharing all k cluster centers during the client initialization process in Algorithm 2, $\mathcal{MFC}_{\mathcal{H}}$ only shares one random center with the server. Our experimental results demonstrate that $\mathcal{MFC}_{\mathcal{H}}$ gives comparable performance to that of $\mathcal{MFC}$ but with much more communication efficiency and higher privacy. We now provide the theoretical analysis of $\mathcal{MFC}_{\mathcal{H}}$ which is similar to $\mathcal{MFC}$ but with an extra result. More specifically, we show that even when only one center is shared during the initialization phase, if there are enough clients, then the server will have at least one representation from each gaussian distribution. With this, one can simply follow the results provided in the previous section to bound the closeness of global and the obtained local centers.

Lemma 6. *If there are at least $k^2 log(k)$ devices then after first round (or shot) there will be at least one data point from each gaussian distribution at the server.*

Proof. As we know, devices share data points with the server for identifying centers; our goal is to determine the minimum expected number of client devices to be included in the network to ensure representation of points from all k gaussian's.

We can map this problem to the coupon collector problem (CCP) [20], which is a classical probability problem. In the CCP, there are a total of k different types of coupons, and if each coupon type is arriving uniformly at random, we need to find the expected number of purchases needed to collect each coupon at least once. More formally,

Claim 7 (Coupon Collector Problem (CCP) [20])**.** *The expected number of purchases to obtain a full collection of m distinct coupons is $m\mathcal{H}_m$ where $\mathcal{H}_m = \left(\frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{m}\right)$ is m^{th} Harmonic number.*

We will now apply principles from CCP to solve our problem. Let us treat each gaussian distribution to be a type of coupon. Given that we have k distributions, the number of points from each distribution can vary significantly. Further, if each client device had points from all k distributions, then the bound would be a straightforward application of CCP. However, clients may have data points from fewer than m distributions, specifically $k_z \leq k$. To handle this, we first determine the minimum number of devices needed to represent points from a particular distribution say (i^{th} distribution) on the server. Let us say that we need m devices that have points from i^{th} distribution on them. Applying CCP, we can say that if we have at least $m = k_z \log k_z \leq k \log k$ devices then there will be at least one point (or representation) from i^{th} distribution at the server.

To extend the bound to all k distributions, we can along similar lines say that if we have $\sum_{i=1}^{k} k \log k = k^2 \log k$ devices then all the distributions will be represented by $\mathcal{MFC}_{\mathcal{H}}$. □

With this bound, all the lemmas hold valid as in $\mathcal{MFC}$.

7 Experimental Analysis

In this section, we now validate the efficacy of our proposed approach on synthetic and *benchmarking* real-world datasets against different state-of-the-art (SOTA) methods. The datasets used in the study are:

- **Synthetic dataset**: The dataset is composed of ten bi-variate gaussian distributions $\{\mathcal{N}_i(\mu = 10i, \sigma^2 = 2)\}_{i=1}^{10}$, with 1000 points each, totaling 10000 points altogether. The dataset obeys our well-separability assumption. The code to generate data is available at https://github.com/shivi98g/MFC.

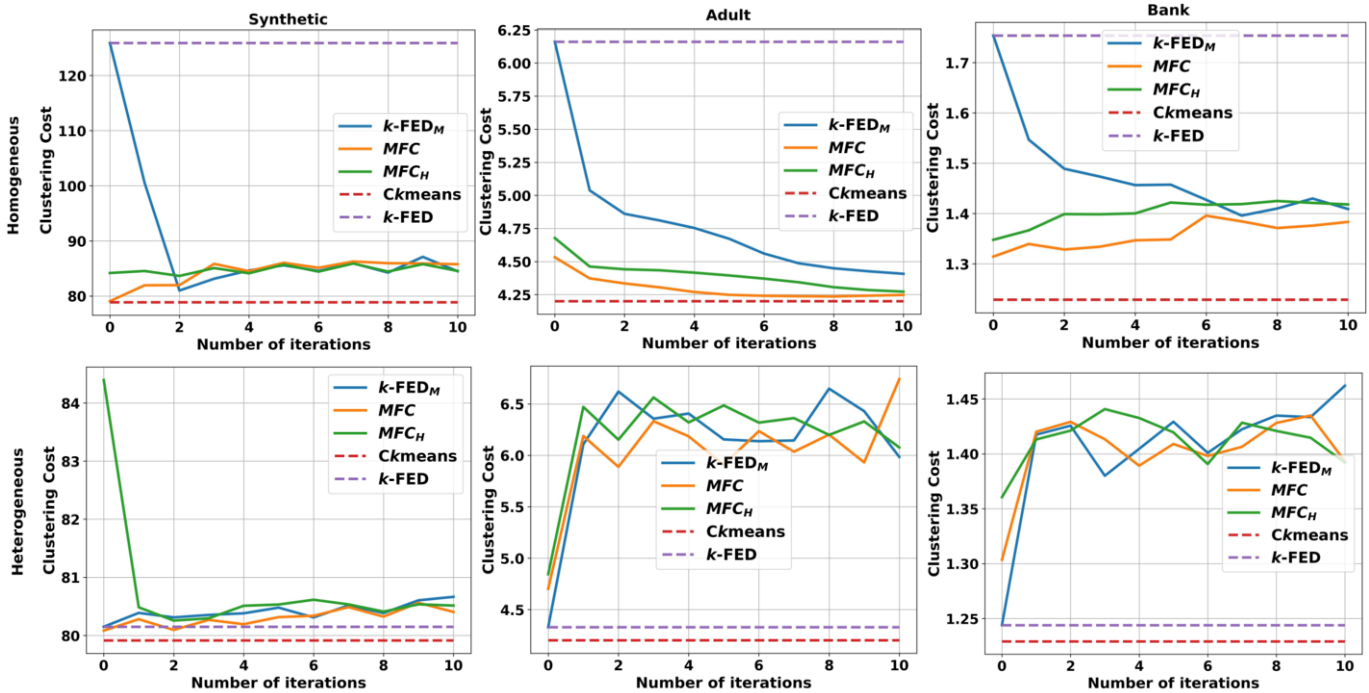

Figure 1: The plot in the first row shows the clustering cost for multiple shots (or iterations) in case of homogeneous distribution. The second shows cost analysis on different datasets in heterogeneous distribution. Each column corresponds to one dataset. (Best viewed in color).

- **Adult:** The 1994 census dataset contains 32,562 records of individuals. We evaluate performance on five attributes as a feature set: age, fnlwgt, education_num, capital_gain, and hours_per_week. These attributes are popular in clustering literature [11, 62, 4, 7]. The dataset is publicly available [49].
- **Bank:** A Portuguese bank marketing dataset of 41,108 records, with each record containing information on the customer's details. We use age, duration, campaign, cons.price.idx, euribor3m, nr.employed as a feature set consistent with clustering literature [11, 62, 4, 7]. The dataset is publicly available [50].

We evaluate the performance of $\mathcal{MFC}$, $\mathcal{MFC}_{\mathcal{H}}$ against following:

- **Centralized k-means (C$k$means)**: The centralized variation of famous Llyod heuristic also simply known as k-means [33].
- **k-FED**: The method assumes the knowledge of k_z on each client device. Each client utilizes this information and executes the Llyod heuristic to obtain local clustering. These are then accumulated over all clients at the server to compute global k-clustering.
- **k-FED$_M$**: To have a fair comparison with multishot approaches, we extend the single shot k-FED to the multishot scenario. In the first shot, k-FED$_M$ follows a similar procedure to k-FED to determine global clustering. For the remaining shots, it employs a technique akin to our $\mathcal{MFC}$.

Metric: We compare SOTAs on clustering cost (see Definition 1).
Experimental Setup: All experiments are executed on i5 8^{th} generation processor, 8GB RAM and Windows 10 with Python 3.8. We report the mean of performance across ten independent runs with different seed values from the set $\{0, 100, \ldots, 900\}$. All approaches do not require any additional hyper-parameters other than the target number of clusters (k) that are taken to 10 across all datasets. The data points are split across 50 client devices, and we examine two different data distribution regimes, namely homogeneous and heterogeneous described in Section 3. We limit $k_z = k\ \forall z \in [Z]$ for experimental study in homogenous setting. To distribute data points among clients in a heterogeneous setting, we randomly allocate each client device with data points from exactly k_z distributions i.e, a heterogeneity of 2 implies each client has points exactly from 2 of k distributions.

7.1 Analysis on Synthetic Datasets

We begin with an empirical analysis of different methods on the synthetic dataset in the first column of Fig. 1. The observations are summarized below for different data distribution settings.
Homogeneous setting: (1) $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$ perform close enough to centralized k-means. In contrast, k-FED experiences a considerable increase in cost in a homogeneous setting and needs multiple shots (i.e., k-FED$_M$) to attain the same level of performance as other SOTA, increasing communication overheads.
Heterogeneous setting: (1) We observe that the initial centers generated by k-FED and $\mathcal{MFC}$ at zeroth shot (iteration) have cost similar to centralized k-means. This can be attributed to k-FED's ability to perform effectively in heterogeneous settings and $\mathcal{MFC}$'s independence from data distribution across clients.
(2) The cost of $\mathcal{MFC}_{\mathcal{H}}$ is slightly higher compared to C$k$means as our heuristic prioritizes privacy by sharing only one center with the server from each client. This might lead to a slight deviation from target global centers in the initial stages, but cost gradually decreases as the number of iterations (shots) increases.
(3) Initially $\mathcal{MFC}_{\mathcal{H}}$ perform low compared to $\mathcal{MFC}$ but over multiple shots, both attain similar performance and stand against SOTA.

7.2 Analysis on Real-world Datasets

We now show empirical analysis of SOTA methods on two real-world datasets - Adult and Bank in the second and third column of Fig. 1.

Homogeneous setting: (1) The initial cost of $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$ are comparable to C$k$means showing its efficacy whereas k-FED fails to cluster in a homogeneous setting resulting in substantially high cost. (2) While in k-FED$_M$ performing multi-shot iterations, though reducing the cost, still results in a considerable gap in converged costs compared to C$k$means, $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$.

Heterogeneous setting: (1) At first glance initial outcome of k-FED seems promising as cost is comparable to C$k$means. However, it is crucial to note that in real-world datasets, it may not be trivial to priorly know if a single shot is adequate to attain convergence. This is because such datasets may not conform entirely to Gaussian distribution and may not satisfy the well-separability assumptions. Thus, may require multiple shots to converge.

(2) On the contrary, the initial costs of $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$ are not quite far from k-FED, and under multiple shots, both $\mathcal{MFC}_{\mathcal{H}}$ and k-FED$_M$ converges to akin equivalent cost values showing the efficacy of our approach on different datasets independent of data distribution.

7.3 Ablation Studies

Ablation Study on Homogeneity– We will now see how SOTA approaches perform as the homogeneity of data increases across clients. We in Fig. 2 observe that k-FED$_M$ performs well only when data is less homogeneous i.e., more heterogeneous whereas our approaches are not susceptible to any such distributional underpinnings.

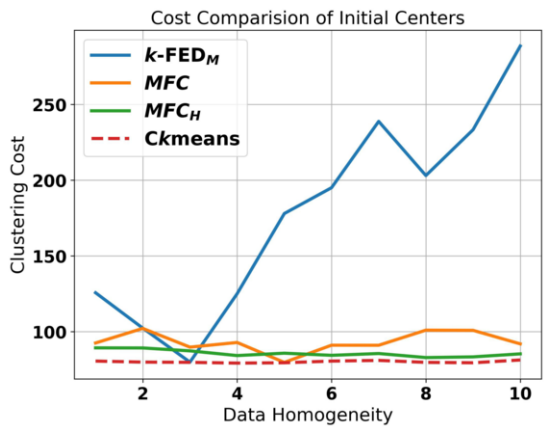

Figure 2: Cost comparison on different SOTA after single shot. (Best viewed in color).

Ablation Study on Fairness – Ensuring that the loss (here, clustering cost) is nearly the same across all client devices is crucial in a federated setting. It ensures that all clients contribute equally to the training process; otherwise, some clients may find the global centers unfair. We examine the fairness of the proposed approach and SOTA methods by analyzing the spread of clustering cost among devices. The parameters used to measure fairness includes standard deviation and the inter-quartile range (IQR) of clustering cost at each device. The fairness decreases as the values of both these parameters increase. The results on fairness metrics for $\mathcal{MFC}$ and k-FED on Synthetic dataset are reported in Table 1. The results for $\mathcal{MFC}_{\mathcal{H}}$ are similar to $\mathcal{MFC}$ and is omitted due to space constraints. Results show that $\mathcal{MFC}$ performs better compared to k-FED showing it's efficacy.

k_z	k-FED			$\mathcal{MFC}$		
	Mean Cost	Std. Dev	IQR	Mean Cost	Std. Dev	IQR
1	149.672	175.013	108.448	**8.698**	**2.461**	**1.965**
2	**14.861**	2.849	4.097	16.604	**2.275**	**1.314**
3	**15.948**	3.239	**4.036**	24.601	**2.977**	5.537
4	**29.013**	6.679	9.227	30.952	**3.338**	**6.255**
5	111.481	66.454	125.552	**40.837**	**4.209**	**5.951**
6	101.676	31.527	40.746	**46.27**	**5.51**	**9.917**
7	120.338	65.106	115.941	**60.052**	**7.913**	**6.873**
8	102.149	28.149	41.383	**73.025**	**19.775**	**16.994**
9	225.712	61.122	58.384	**71.229**	**11.809**	**17.989**
10	182.989	48.496	56.827	**85.635**	**12.553**	**19.413**

Table 1: Fairness metrics on SOTA approaches over different levels of homogeneity ($k_z \leq k$) on Synthetic dataset (k = 10).

8 Conclusion

We proposed two data distribution-independent federated clustering algorithms - $\mathcal{MFC}$ and $\mathcal{MFC}_{\mathcal{H}}$. Unlike SOTA, both proposed methods do not require knowledge about the number of local clusters (k_z) on clients. We also show that $\mathcal{MFC}_{\mathcal{H}}$ ensures better privacy by sharing only a single center with the server. Further, we theoretically prove that under well-separability assumptions, centers obtained by our algorithms are within proximity of global centers. Our experimental analysis shows that proposed methods outperform SOTA. An immediate future direction can be detailed analysis of fairness on aspects like group and individual fairness [48, 37]. In addition, the adaptation methods for addressing problems such as client selection could be considered a promising area for future research [12].

Acknowledgement

The authors thank PMRF for funding Shivam. The research is also supported by the SERB, India, with grant number CRG/2022/004980.

References

[1] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier, 'Federated learning for healthcare: Systematic review and architecture proposal', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **13**(4), 1–23, (2022).

[2] David Arthur and Sergei Vassilvitskii, 'k-means++: The advantages of careful seeding', Technical report, Stanford, (2006).

[3] Pranjal Awasthi and Or Sheffet, 'Improved spectral-norm bounds for clustering', *Approximation, Randomization, and Combinatorial Optimization*, 37–49, (2012).

[4] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner, 'Scalable fair clustering', in *ICML*, (2019).

[5] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii, 'Scalable k-means+', *VLDB Endowment*, **5**(7), (2012).

[6] Sourasekhar Banerjee, Rajiv Misra, Mukesh Prasad, Erik Elmroth, and Monowar H Bhuyan, 'Multi-diseases classification from chest-x-ray: A federated deep learning approach', in *Advances in Artificial Intelligence: Australasian Joint Conference*, pp. 3–15. Springer, (2020).

[7] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani, 'Fair algorithms for clustering', *NeurIPS*, **32**, (2019).

[8] Panthadeep Bhattacharjee and Pinaki Mitra, 'A survey of density based clustering algorithms', *Frontiers of Computer Science*, **15**, 1–27, (2021).

[9] Johannes Blömer, Christiane Lammersen, Melanie Schmidt, and Christian Sohler, 'Theoretical analysis of the k-means algorithm–a survey', *Algorithm Engineering: Selected Results and Surveys*, 81–116, (2016).

[10] Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys, 'A constant-factor approximation algorithm for the k-median problem', in *ACM STOC*, pp. 1–10, (1999).

[11] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii, 'Fair clustering through fairlets', in *NeurIPS*, (2017).

[12] Yae Jee Cho, Jianyu Wang, and Gauri Joshi, 'Client selection in federated learning: Convergence analysis and power-of-choice selection strategies', *arXiv:2010.01243*, (2020).

[13] Jichan Chung, Kangwook Lee, and Kannan Ramchandran, 'Federated unsupervised clustering with generative models', in *AAAI Workshop on Trustable, Verifiable and Auditable Federated Learning*, (2022).

[14] Cornell. Unsupervised learning: Next frontier ai. https://www.cs.cornell.edu/content/unsupervised-learning-next-frontier-ai, 2023. [Online; accessed 03-May-2023].

[15] Xiaodong Cui, Songtao Lu, and Brian Kingsbury, 'Federated acoustic modeling for automatic speech recognition', in *IEEE ICASSP*, pp. 6748–6752, (2021).

[16] Don Kurian Dennis, Tian Li, and Virginia Smith, 'Heterogeneity for the win: One-shot federated clustering', in *ICML*, (2021).

[17] Sara Dolnicar, 'A review of data-driven market segmentation in tourism', *Journal of Travel & Tourism Marketing*, **12**(1), 1–22, (2002).

[18] Hung Du, Srikanth Thudumu, Sankhya Singh, Scott Barnett, Irini Logothetis, Rajesh Vasa, and Kon Mouzakis, 'Decentralized federated learning strategy with image classification using resnet architecture', in *IEEE CCNC*, pp. 706–707, (2023).

[19] Reza Farivar, Daniel Rebolledo, Ellick Chan, and Roy H Campbell, 'A parallel implementation of k-means clustering on gpus.', in *Pdpta*, volume 13, pp. 212–312, (2008).

[20] Philippe Flajolet, Daniele Gardy, and Loÿs Thimonier, 'Birthday paradox, coupon collectors, caching algorithms and self-organizing search', *Discrete Applied Mathematics*, **39**(3), 207–229, (1992).

[21] Mo Hai, Shuyun Zhang, Lei Zhu, and Yue Wang, 'A survey of distributed clustering algorithms', in *IEEE ICICEE*, pp. 1142–1145, (2012).

[22] Dorit S Hochbaum and David B Shmoys, 'A best possible heuristic for the k-center problem', *Mathematics of operations research*, **10**(2), 180–184, (1985).

[23] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown, 'Federated visual classification with real-world data distribution', in *ECCV*, pp. 76–92. Springer, (2020).

[24] Sirui Hu, Ling Feng, Xiaohan Yang, and Yongchao Chen, 'Fedssc: Shared supervised-contrastive federated learning', *arXiv:2301.05797*, (2023).

[25] Younghwan Jeong and Taeyoon Kim, 'A cluster-driven adaptive training approach for federated learning', *Sensors*, **22**(18), 7061, (2022).

[26] Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi, 'Fedvarp: Tackling the variance due to partial client participation in federated learning', in *UAI*, pp. 906–916, (2022).

[27] Ce Ju, Dashan Gao, Ravikiran Mane, Ben Tan, Yang Liu, and Cuntai Guan, 'Federated transfer learning for eeg signal classification', in *IEEE EMBC*, pp. 3040–3045, (2020).

[28] Deep Kawa, Sunaina Punyani, Priya Nayak, Arpita Karkera, and Varshapriya Jyotinagar, 'Credit risk assessment from combined bank records using federated learning', *International Research Journal of Engineering and Technology (IRJET)*, **6**(4), 1355–1358, (2019).

[29] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady, 'Dbscan: Past, present and future', in *IEEE ICADIWT*, pp. 232–238, (2014).

[30] Jonas Moritz Kohler and Aurelien Lucchi, 'Sub-sampled cubic regularization for non-convex optimization', in *ICML*, pp. 1895–1904, (2017).

[31] Songze Li, Sizai Hou, Baturalp Buyukates, and Salman Avestimehr, 'Secure federated clustering', *arXiv:2205.15564*, (2022).

[32] Yiwei Li, Shuai Wang, Chong-Yung Chi, and Tony Q. S. Quek. Differentially private federated clustering over non-iid data, 2023.

[33] Stuart Lloyd, 'Least squares quantization in pcm', *IEEE transactions on information theory*, **28**(2), 129–137, (1982).

[34] Sascha Löbner, Boris Gogov, and Welderufael B Tesfay, 'Enhancing privacy in federated learning with local differential privacy for email classification', in *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS International Workshops, Revised Selected Papers*, pp. 3–18. Springer, (2023).

[35] Renhao Lu, Weizhe Zhang, Yan Wang, Qiong Li, Xiaoxiong Zhong, Hongwei Yang, and Desheng Wang, 'Auction-based cluster federated learning in mobile edge computing systems', *IEEE Transactions on Parallel and Distributed Systems*, **34**(4), 1145–1158, (2023).

[36] Yu Lu and Harrison H Zhou, 'Statistical and computational guarantees of lloyd's algorithm and its variants', *arXiv:1612.02099*, (2016).

[37] Sepideh Mahabadi and Ali Vakilian, 'Individual fairness for k-clustering', in *ICML*, (2020).

[38] 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*, pp. 1273–1282, (2017).

[39] Fionn Murtagh and Pedro Contreras, 'Algorithms for hierarchical clustering: an overview', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **2**(1), 86–97, (2012).

[40] Elth Ogston, Benno Overeinder, Maarten Van Steen, and Frances Brazier, 'A method for decentralized clustering in large multi-agent systems', in *AAMAS*, (2003).

[41] Sanguthevar Rajasekaran, 'Efficient parallel hierarchical clustering algorithms', *IEEE transactions on parallel and distributed systems*, **16**(6), 497–502, (2005).

[42] Somula Ramasubbareddy, T Aditya Sai Srinivas, K Govinda, and SS Manivannan, 'Comparative study of clustering techniques in market segmentation', *Springer ICICSE*, 117–125, (2020).

[43] Sogo Pierre Sanon, Rekha Reddy, Christoph Lipps, and Hans Dieter Schotten, 'Secure federated learning: An evaluation of homomorphic encrypted network traffic prediction', in *IEEE CCNC*, pp. 1–6. IEEE, (2023).

[44] Yuris Mulya Saputra, Dinh Thai Hoang, Diep N Nguyen, Eryk Dutkiewicz, Markus Dominik Mueck, and Srikathyayani Srikanteswara, 'Energy demand prediction with federated learning for electric vehicle networks', in *IEEE GLOBECOM*, pp. 1–6, (2019).

[45] John A Saunders, 'Cluster analysis for market segmentation', *European Journal of marketing*, **14**(7), 422–435, (1980).

[46] Hwanjun Song, Jae-Gil Lee, and Wook-Shin Han, 'Pamae: parallel k-medoids clustering with high accuracy and efficiency', in *ACM SIGKDD*, pp. 1087–1096, (2017).

[47] Joel Stremmel and Arjun Singh, 'Pretraining federated text models for next word prediction', in *Springer FICC*, pp. 477–488, (2021).

[48] **Shivam Gupta**, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain, 'Efficient algorithms for fair clustering with a new notion of fairness', *Data Mining and Knowledge Discovery Journal*, 1–39, (2023).

[49] UCI. Adult dataset (census). https://archive.ics.uci.edu/ml/datasets/Adult, 1994. [Online; accessed 15-August-2021].

[50] UCI. Bank dataset. https://archive.ics.uci.edu/ml/datasets/Bank+Marketing, 2014. [Online; accessed 15-August-2021].

[51] Yiqiu Wang, Yan Gu, and Julian Shun, 'Theoretically-efficient and practical parallel dbscan', in *ACM SIGMOD*, pp. 2555–2571, (2020).

[52] Xiao-Xiang Wei and Hua Huang, 'Edge devices clustering for federated visual classification: A feature norm based framework', *IEEE Transactions on Image Processing*, (2023).

[53] Jeffry Wicaksana, Zengqiang Yan, Dong Zhang, Xijie Huang, Huimin Wu, Xin Yang, and Kwang-Ting Cheng, 'Fedmix: Mixed supervised federated learning for medical image segmentation', *IEEE Transactions on Medical Imaging*, (2022).

[54] Rui Xu and Donald Wunsch, 'Survey of clustering algorithms', *IEEE Transactions on neural networks*, **16**(3), 645–678, (2005).

[55] Jie Yan, Jing Liu, Ji Qi, and Zhong-Yuan Zhang. Federated clustering with gan-based data synthesis, 2022.

[56] Liu Yang, Ben Tan, Vincent W Zheng, Kai Chen, and Qiang Yang, 'Federated recommendation systems', *Federated Learning: Privacy and Incentive*, 225–239, (2020).

[57] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu, 'Ffd: A federated learning based method for credit card fraud detection', in *Big Data, Services Conference Federation*. Springer, (2019).

[58] Minghao Ye, Junjie Zhang, Zehua Guo, and H Jonathan Chao, 'Federated traffic engineering with supervised learning in multi-region networks', in *IEEE ICNP*, pp. 1–12, (2021).

[59] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao, 'A survey on federated learning', *Knowledge-Based Systems*, **216**, (2021).

[60] Yi Zhang, Miaomiao Li, Siwei Wang, Sisi Dai, Lei Luo, En Zhu, Huiying Xu, Xinzhong Zhu, Chaoyun Yao, and Haoran Zhou, 'Gaussian mixture model clustering with incomplete data', *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, **17**(1s), 1–14, (2021).

[61] Yufang Zhang, Zhongyang Xiong, Jiali Mao, and Ling Ou, 'The study of parallel k-means algorithm', in *World Congress on Intelligent Control and Automation*, volume 2, pp. 5868–5871, (2006).

[62] Imtiaz Masud Ziko, Jing Yuan, Eric Granger, and Ismail Ben Ayed, 'Variational fair clustering', in *AAAI*, volume 35, (2021).