Privacy-Preserving Optimal Parameter Selection for Collaborative Clustering

Maryam Ghasemian and Erman Ayday(✉)

Case Western Reserve University, Cleveland, OH, USA
{maryam.ghasemian,erman.ayday}@case.edu

Abstract. This study investigates the optimal selection of parameters for collaborative clustering while ensuring data privacy. We focus on key clustering algorithms within a collaborative framework, where multiple data owners combine their data. A semi-trusted server assists in recommending the most suitable clustering algorithm and its parameters. Our findings indicate that the privacy parameter (ϵ) minimally impacts the server's recommendations, but an increase in ϵ raises the risk of membership inference attacks, where sensitive information might be inferred. To mitigate these risks, we implement differential privacy techniques, particularly the Randomized Response mechanism, to add noise and protect data privacy. Our approach demonstrates that high-quality clustering can be achieved while maintaining data confidentiality, as evidenced by metrics such as the Adjusted Rand Index and Silhouette Score. This study contributes to privacy-aware data sharing, optimal algorithm and parameter selection, and effective communication between data owners and the server.

Keywords: Clustering · Privacy · Differential Privacy · Membership Inference Attack · Data Mining · Machine Learning

1 Introduction

Clustering, a fundamental technique in unsupervised machine learning, involves identifying patterns in unlabeled data. This process includes feature selection, measuring data similarity, and evaluating algorithms [21,35]. There are several types of clustering algorithms: partitioning based [25,34], distribution based [19, 22], density based [3,6,31], and hierarchical [11,24]. Our study concentrates on selecting the optimal hyperparameters for key representative clustering algorithms from each category, within a privacy-preserving collaborative framework. Specifically, we explore K-Means (partitioning-based), Hierarchical Clustering (HC, hierarchical), Gaussian Mixture Models (GMM, distribution-based), and DBSCAN (density-based).Choosing the right parameters is crucial as it directly impacts the accuracy and effectiveness of the clustering results, thereby influencing the insights derived from the data while maintaining privacy.

Motivated by the fact that clustering algorithm perform better with larger amount of data and that datasets are typically distributed across different parties, cooperative clustering and collaborative clustering [7] techniques have

J. Garcia-Alfaro et al. (Eds.): ESORICS 2024 Workshops, LNCS 15263, pp. 13–32, 2025.
https://doi.org/10.1007/978-3-031-82349-7_2

been popular. In cooperative clustering, each party generates its own clustering results, and a final clustering is performed via a post-processing step once individual processes are completed. In contrast, collaborative clustering aims to leverage the contributions of multiple parties by exchanging information about local data, current hypothesized clustering, or algorithm parameters to benefit each other's computations. Due to privacy concerns of the parties, privacy-preserving algorithms have been proposed during collaborative clustering, which aim to protect the sensitive information in each parties' local dataset. However, depending on the type of clustering (partitioning-based, distribution-based, density-based, or hierarchical clustering), parties need to decide on some common input parameters. Selection of such parameters significantly effect the accuracy of the clustering algorithm and existing privacy-preserving collaborative clustering techniques assume such parameters are pre-selected. On the other hand, such parameters typically depend on the distribution of the federated dataset of the parties and they should be determined in a privacy-preserving way before the collaborative clustering. In addition, parties also need to decide the type of the clustering algorithm depending on their federated dataset as different types of algorithms perform differently in particular datasets. To fill this gap, we focus on a server-assisted scenario for collaborative clustering, aiming to evaluate server-provided input parameters and clustering algorithms. We experiment with K-Means, Hierarchical Clustering, Gaussian Mixture Models, and DBSCAN using a labeled numeric dataset, assessing results with metrics like Adjusted Rand Index (ARI) and Silhouette Score.

Using a semi-trusted server enhances privacy and helps select optimal clustering algorithms and parameters without burdening data owners with large computational resources. Differential privacy techniques safeguard data throughout the process. Our findings show that this approach effectively maintains data privacy while delivering high-quality clustering, evidenced by ARI and Silhouette Scores. The Randomized Response mechanism efficiently preserves data structure while protecting privacy.

In this work, we make the following contributions to the context of collaborative clustering with hyper parameter recommendation:

1. Privacy-Preserving and Efficient Communication: We introduce a novel privacy-preserving step in the collaborative clustering process, where data owners share parts of their datasets with the server after applying the randomized response (RR) mechanism to add noise to their respective datasets. This step enhances privacy protection by concealing sensitive information while still allowing for meaningful analysis. Additionally, we establish a seamless communication framework between the data owners and the server, ensuring privacy-preserving data sharing. Unlike previous works that primarily rely on pre-selected clustering parameters and then apply encryption techniques in distributed or collaborative clustering, our approach goes beyond by addressing the challenge of parameter selection by determining the optimal clustering algorithm along with the respective hyper-parameters and incorporating the randomized response (RR) mechanism to introduce noise and safeguard sensitive information during data sharing.

2. Optimal Algorithm Selection: The server plays a crucial role in identifying the optimal clustering algorithm and its corresponding hyper-parameters. By employing various methods, the server evaluates different algorithms and provides data owners with recommendations for achieving the best clustering results. This step helps alleviate the burden of algorithm selection and parameter tuning for data owners.
3. Server-Data Owner Interaction: The server communicates chosen algorithms and parameters back to the data owners, ensuring that all parties are aligned with the recommended strategies. This facilitates a coordinated effort that enhances both accuracy and efficiency.

In summary, our study contributes to privacy-aware data sharing, optimal algorithm and hyper-parameter selection, and effective communication between data owners and the server. The results revealed that the amount of noisy data shared and the privacy budget (ϵ) did not significantly affect the server's algorithm and parameter recommendations. However, an increase in the privacy budget was found to elevate the risk of membership inference attacks, suggesting a trade-off between privacy protection and attack vulnerability.

2 Related Work

Our study reviews privacy-preserving approaches in distributed and collaborative clustering, categorized by algorithm types, introduced in Sect. 1. Existing methods typically use predefined algorithms and hyperparameters, while our contribution dynamically identifies optimal clustering algorithms and hyperparameters to enhance collaborative clustering performance in a privacy-aware manner.

Bi et al.'s PriKPM scheme [5] introduces a privacy-preserving k-prototype clustering method using additive secret sharing to handle mixed data types in cloud environments, addressing privacy concerns. This framework ensures clustering privacy through secure processing by dual servers, validated by experiments demonstrating computational efficiency and accuracy.

Wang et al. [33] propose a privacy-preserving k-means clustering model for IoT, using multi-key fully homomorphic encryption for secure cloud-edge computations. The model optimizes resource use and ensures data privacy through secure communication protocols, demonstrating the feasibility of privacy-sensitive cloud-edge collaborations with minimal overhead.

Further contributions include Jagannathan and Wright's [20], as well as Baby et al.'s [4], protocols for privacy-preserving distributed K-Means clustering, designed for data partitioned arbitrarily. These protocols maintain data confidentiality while following the K-Means algorithm's iterative nature, allowing secure computation of cluster centers and distances without data exposure.

Additionally, Lin et al. [22] present an expectation maximization-based strategy for private clustering across distributed sites, utilizing secure summation to protect horizontally partitioned data. Liu et al. [23] offer privacy-preserving

Table 1. Overview of Adversary and System Models in Related Works

Reference	Clustering Algorithm	System Model	Adversary Model	Privacy Technique
Bi et al. [5]	k-Prototype	Cloud-based with dual servers	Semi-honest adversary	Additive Secret Sharing
Wang et al. [33]	k-Means	IoT ecosystem with cloud-edge collaboration	Semi-honest adversary	Multi-Key Fully Homomorphic Encryption
Jagannathan [20], Baby et al. [4]	k-Means	Arbitrarily partitioned data, distributed	Honest-but-curious adversary	Secure Multiparty Computation (SMC)
Lin et al. [22]	Expectation Maximization	Distributed sites with horizontally partitioned data	Honest-but-curious adversary	Secure Summation
Liu et al. [23]	DBSCAN	Distributed with various data partitions	Honest-but-curious adversary	Additive Homomorphic Encryption
Meng et al. [24]	Hierarchical Clustering	Two-party model	Semi-honest adversary	Homomorphic Encryption and Garbled Circuits
Our Work	Multiple (K-Means, HC, GMM, DBSCAN)	Semi-trusted server in collaborative clustering	Semi-honest server, honest-but-curious data owners	Local Differential Privacy, Randomized Response

DBSCAN techniques for data distributed in various ways, employing a Multiplication protocol based on additive homomorphic encryption for secure clustering.

Meng et al. [24] introduce privacy-preserving hierarchical clustering algorithms, emphasizing a two-party model that employs homomorphic encryption and garbled circuits. Their approach provides a dendrogram depicting the clustering process, enriched with detailed merge metadata.

These diverse approaches share a common goal of enhancing privacy in collaborative clustering, yet they employ fixed algorithms and parameters. Our study seeks to advance this domain by focusing on adaptive parameter selection to achieve optimal clustering results, reflecting a significant leap toward balancing privacy preservation and analytical utility in collaborative settings. To provide a clearer comparison of the various approaches, Table 1 summarizes the adversary models and system models considered in the related works discussed above.

3 Background

In this section we review some background and definitions of different clustering algorithms and clustering evaluation metrics as well as the local differential privacy.

3.1 Clustering Algorithms

This study explores four clustering algorithms: partitioning-based, distribution-based, density-based, and hierarchical [3,6,11,19,22,24,25,34]. K-Means, a widely used unsupervised algorithm, partitions data into K non-overlapping clusters by minimizing distances between data points and centroids [25,34]. Gaussian Mixture Models (GMM) handle clusters with varying sizes and correlations by assuming data is generated from a mixture of Gaussian distributions [19,22]. DBSCAN identifies clusters of arbitrary shapes based on data density and automatically detects outliers, without needing to predefine the number of clusters, though it is sensitive to its parameters: neighborhood size (*Eps*) and minimum points (*minpoint*) [3,6]. Hierarchical clustering creates a tree of clusters without a pre-specified number, using either a bottom-up or top-down approach. It is useful for hierarchical data but is computationally intensive and varies with the linkage criterion used [14,15,17,24,36].

Table 2. Table of symbols and notations.

Symbol	Description
D_i	Dataset of each data owner i
ND_i	Noisy data of each data owner i produced as a result of RR
f_{NDi}	Portion of the noisy data, ND_i, shared with server from each data owner i
RR	Randomized Response mechanism
ϵ, eps	epsilon, Privacy Parameter
Eps	Epsilon, Maximum distance between clusters in DBSCAN
k	Number of clusters
ARI	Adjusted Rand Index
CH	Calinski-Harabasz Index
Homo	Homogeneity of the clusters
Comp	Completeness

3.2 Evaluation Metrics for Clustering Algorithms

This section outlines the evaluation metrics used to assess the effectiveness of the proposed privacy-preserving collaborative clustering approach. To measure the performance of our approach, we use the following metrics, each selected for its capability to capture various dimensions of clustering quality and privacy preservation:

Adjusted Rand Index (ARI): Measures the similarity between two clusterings, with scores ranging from -1 (independent clusterings) to 1 (perfect agreement). *Higher* ARI values indicate better clustering performance.

Silhouette Coefficient Score: Evaluates cluster cohesion and separation, with scores ranging from -1 to 1. *Higher* values indicate better-defined clusters.

Calinski-Harabasz Index (CH): Measures clustering quality based on the ratio of between-cluster dispersion to within-cluster dispersion. *Higher* CH values indicate better separation between clusters.

Classification Accuracy: We also added classification accuracy to our evaluation framework, a metric that measures the proportion of correct predictions. Although unusual in unsupervised learning tasks like clustering, it helps evaluate how well cluster assignments match predefined labels when known. This metric is key in scenarios with known data classifications, allowing for direct comparison between our privacy-preserving clusters and actual categories.

Table 2 contains a list of symbols and notations used throughout this paper.

3.3 Local Differential Privacy and Randomized Response Mechanism

Local Differential Privacy (LDP) [8,10] is a more restricted form of traditional differential privacy [9]. Unlike traditional differential privacy, LDP does not rely on a trusted third party and provides a higher level of data protection for users. In LDP, each user modifies their own data before sharing them with a data aggregator. The aggregator only sees the perturbed data, ensuring privacy. An algorithm A satisfies ϵ-local differential privacy (ϵ-LDP) if, for any input values $v1$ and $v2$: $Pr[A(v1) = y] \leq e^{\epsilon} Pr[A(v2) = y]$, This condition holds true for all possible outputs of the algorithm A. The randomized response mechanism is commonly used to achieve $\epsilon - LDP$ [12]. In this mechanism, an individual reports the true value of a single bit of information with probability p and flips the true value with probability $1-p$, following the $(ln\frac{p}{1-p}) - LDP$ property. Although initially defined for binary inputs (e.g., yes/no), the randomized response mechanism can be generalized. To achieve ϵ-LDP, the generalized randomized response mechanism [18] shares the correct value with probability $p = \frac{e^{\epsilon}}{(e^{\epsilon}+m-1)}$ where m is the number of possible states. Each incorrect value is shared with the probability. $q = \frac{1}{(e^{\epsilon}+m-1)}$. A data aggregator collects the perturbed values from individuals and aims to calculate the frequency of values in the population while preserving privacy.

4 System and Threat Models

In this section, we provide an explanation of the system and threat model for privacy-preserving hyper-parameter identification for collaborative clustering.

4.1 System Model

In the proposed system model, the party who aims to collaborate in clustering with other data owners is referred to as the "data owner" (or researcher), while the server represents a third party that assists the data owners in identifying the

optimal clustering algorithm and hyper-parameters. Our approach focuses on the preliminary stages before actual clustering occurs in a collaborative environment. Our objective is to identify the optimal algorithm and input parameters for collaborative clustering among multiple data owners who wish to maintain data privacy. As discussed, different types of clustering algorithms perform differently depending on the type and distribution of the datasets, and hence it is crucial to identify the optimal clustering algorithm type beforehand. Once these optimal conditions are determined, clustering can then be executed using one of the existing algorithms mentioned in Sect. 3.1. In this context, data owners selectively share differentially private data with a semi-trusted server. This server plays a crucial intermediary role, analyzing the noisy data to recommend the most suitable clustering algorithm and corresponding hyper-parameters for the data received from data owners.

4.2 Threat Model

In this section, we outline the considered threats in our proposed scheme, which involve both the server and the data owners.

Server: In this study, the server is considered semi-honest, indicating it might engage in malicious activities, such as extracting sensitive information from the datasets of the individual parties (data owners), but it honestly follows the protocol execution. The server's role is pivotal, yet poses a risk of privacy violations. Privacy attacks like membership inference [28–30], deanonymization [26,27,29], and attribute inference [13,29] are concerns. Membership inference attacks aim to determine whether a specific record is in the dataset. Deanonymization attacks link anonymized data to actual identities using external information. Attribute inference attacks deduce sensitive attributes from observed data. In our setting, the most relevant is membership inference, where the server tries to determine if a specific record is part of one of the data owners' datasets, leading to privacy breaches. Our proposed scheme prevents this by sharing only a small, differentially-private portion of the dataset (f_{NDi}), which makes deanonymization more complex and significantly reduces the threat of membership inference.

Data Owners: In our system model, we assume that the parties involved in the collaborative clustering are honest but curious. This means that while they trust each other and do not engage in malicious behavior, they may still be interested in learning about each other's data. This assumption is based on the fact that other literature (such as those in Sect. 2) has already addressed the challenges posed by malicious or semi-honest data owners in collaborative clustering using privacy-enhancing techniques like homomorphic encryption. In our work, we specifically focus on the task of selecting the optimal algorithm and hyper-parameters for the clustering process. By concentrating on this aspect, we aim to improve the efficiency and effectiveness of collaborative clustering while assuming a cooperative environment among the data owners.

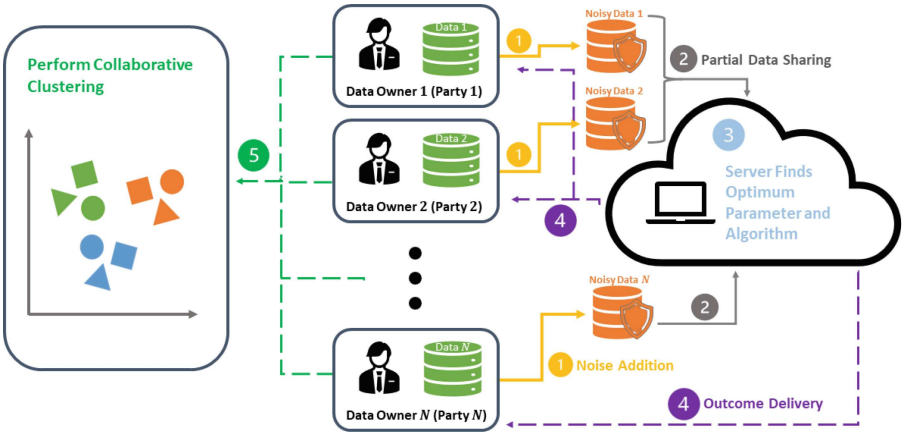


Fig. 1. Comprehensive five-step process, highlighting the interaction between multiple data owners and the server. We show how data are shared, processed for noise addition (to achieve differential privacy), and then utilized in a collaborative clustering algorithm, all while maintaining strict privacy protocols. In step (1), data owners add noise to part of their datasets using randomized response (RR). Data owners send a portion of their noisy data to the server in step (2). In step (3), the server applies various methods to find the optimum algorithm with its corresponding hyper parameter(s), and the server provides its outcome (algorithm and parameter) to the data owners in step (4). Finally, the data owners perform collaborative clustering based on server suggestions in step (5).

5 Proposed Solution and Framework

Our proposed system model and framework, as shown in Figure 1, encompass five fundamental steps:

Step 1-Noise Addition to Datasets: Data owners ($DO_{1\rightarrow N}$) add noise to their datasets (to achieve differential privacy) through randomized response (RR) ($\{D_1, D_2, ..., D_N\} \rightarrow \{ND_1, ND_2, ..., ND_N\}$). In this process, we utilize a generalized version of the RR mechanism as mentioned in Sect. 3.3, allowing data owners to use perturbed data directly without encoding. The number of possible states for each feature (attribute) can vary according to the specific domain.

Step 2-Data Sharing with the Server: Data owners transmit a portion of their noisy data (f_{NDi}) to the server. During this step, data owners share their perturbed data with the server, enabling it to analyze the data and provide recommendations for the clustering process.

Step 3-Server-Based Algorithm and Parameter Selection: The server selects the best clustering algorithm and its hyperparameters using collaborative clustering, where multiple data owners keep their data private with the Generalized Randomized Response (RR) mechanism. Each owner sends noisy data to a semi-trusted server, which combines the datasets and uses methods like the

elbow method and silhouette method [37] to determine optimal parameters for algorithms such as K-Means, hierarchical clustering, Gaussian mixture models. For DBSCAN, it sets the *Eps* value using the k-Nearest Neighbors algorithm and adjusts the *minpoint* parameter based on data dimensionality, following different recommendations from prior research [14,32].

One of the challenges the server faces is the absence of ground truth data. To address this, the server uses internal performance evaluation metrics that do not require ground truth, such as the Silhouette Coefficient and the Calinski-Harabasz (CH) index. These metrics objectively measure the effectiveness of different algorithms, guiding the server in its selection process.

Here is the selection mechanism that server adapts to select the optimum clustering algorithm and its corresponding parameters for the data it received from data owners: The input to the selection algorithm includes a combined dataset from all data owners (*data*), a list of candidate clustering algorithms (*algorithms*), and a threshold parameter set to 0.1 (α). The output is the optimal clustering algorithm (*best_algorithm*) and its corresponding parameters (*best_parameters*). The procedure begins by initializing *max_silhouette* to $-\infty$, *best_algorithm* to *None*, *best_parameters* to *None*, and *best_ch_index* to $-\infty$. It evaluates all algorithms, updating *max_silhouette* if the Silhouette score is higher. The algorithm then sets a silhouette threshold ($max\_silhouette - \alpha$) and selects algorithms within this range with the highest CH index, updating *best_algorithm*, *best_parameters*, and *best_ch_index* accordingly.

Step 4-Communication of Recommendations: The server communicates the recommended clustering algorithm and its parameters to the data owners, based on the analysis of the shared data.

Step 5-Execution of Collaborative Clustering: Data owners apply the suggested algorithm and hyper-parameters for collaborative clustering. As discussed in Sect. 2, previous approaches often used encryption for distributed or collaborative clustering. In contrast, this study focuses on selecting the optimal algorithm and hyper-parameters, assuming mutual trust among data owners for clustering on the combined dataset. Further details are provided in Sect. 4.2.

By following these steps, our framework provides recommendations for the optimal clustering algorithm and its hyper-parameters when data owners wish to perform clustering in a collaborative environment.

6 Evaluation

6.1 Datasets

We use the Obesity dataset [2] (2,111 records, 17 features) and the Extended Iris dataset [1] (1,200 rows, 20 features) which is an enhanced version of the classic Iris dataset [16]. The Obesity dataset assesses obesity levels based on diet and physical condition, while the Extended Iris dataset provides detailed biological and ecological information about the iris flower. These datasets were chosen due to their varying characteristics and complexity, which provide a comprehensive evaluation of our proposed approach across different types of data distributions and clustering challenges.

6.2 Metric Significance and Evaluation Approach

ARI, Silhouette Score, classification accuracy, and Calinski-Harabasz Index (CH) provide a comprehensive performance view. ARI and Silhouette Score assess internal cluster consistency and separation, while classification accuracy offers external validation, and CH highlights cluster distinctness. Together, these metrics enable a thorough assessment of both the clustering effectiveness and the impact of privacy-preserving techniques on data utility. In our evaluation, we analyze these metrics under varying conditions of data perturbation and privacy budget settings to explore the trade-offs between clustering quality and privacy preservation. The goal is to achieve optimal hyper-parameter selection that balances these aspects effectively, demonstrating the practical utility of our approach in collaborative clustering scenarios.

6.3 Evaluation Results

The datasets were pre-processed by converting categorical variables to numerical values for analysis. To determine the optimal number of clusters, we applied the elbow and silhouette methods, as detailed in Sect. 5. Our experiments, particularly under varying privacy budgets (ϵ), aimed to identify the most effective method for our data. The results for *dataset 1* are shown in Table 3.

Given that *dataset 1* has 7 clusters and *dataset 2* has 3, our analysis shows that the elbow method outperforms the silhouette method in determining the optimal cluster count. Consequently, we use the elbow method for a more detailed analysis, aiding in the selection of the optimal k for clustering algorithms like K-Means, hierarchical clustering, and Gaussian mixture models.

Table 3. Comparison of Silhouette and Elbow Methods for Predicting the Optimal Number of Clusters (k): It highlights the superior performance of the Elbow method in predicting the optimal cluster count, leading to its selection for further analysis in this study.

ϵ	Baseline K	Silhouette K	Elbow K
0.0010	7	2	8
0.1000	7	2	8
1.00	7	2	8
5.00	7	2	7
10.00	7	2	7

Optimum Input Parameter Selection Results on Noisy Datasets: The experimental findings of this study are illustrated in Table 4 and Fig. 2. Table 4 offers a glimpse into the server's input parameter recommendations, based on

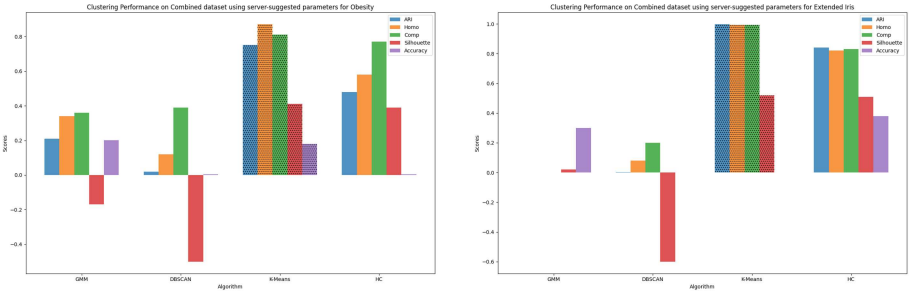
the analysis of 10% of the noisy data shared by the data owners, with a noise parameter (ϵ) set at 0.1. Figure 2, on the other hand, showcases the clustering outcomes derived from applying these server recommendations to the combined dataset. Notably, the results from this application highlight the superiority of the K-Means clustering algorithm for the combined dataset, a finding that resonates with the server's initial suggestion regarding the most suitable algorithm and hyper-parameter configuration. These findings and recommendations by the server are not merely data points, but they serve as critical guidance for the data owners. They enable the owners to align their clustering strategies with the server's insights, which are rooted in a meticulous analysis of optimal input parameters. This alignment is key to enhancing the effectiveness and accuracy of the clustering process in a collaborative, privacy-preserving data environment.

Table 4. Server Suggestions for Clustering Input Parameters: Recommendations for various clustering algorithms based on 10% shared noisy data ($\epsilon = 0.1$).

Dataset	Algorithm	Data shared to Server	ϵ	K or Eps	Silhouette	CH
Dataset #1	GMM	10%	0.1	k = 8	0.34	301.30
	DBSCAN	10%	0.1	k = 10, Eps = 1	-	-
	K-Means	**10%**	**0.1**	**k = 8**	**0.36**	**318.13**
	HC	10%	0.1	k = 8	0.31	237.61
Dataset #2	GMM	10%	0.1	k = 3	0.23	46.88
	DBSCAN	10%	0.1	k = 6, Eps = 7	-	-
	K-Means	**10%**	**0.1**	**k = 3**	**0.36**	**61.92**
	HC	10%	0.1	k = 3	0.37	51.57

Effect of Privacy Parameter ϵ: We have examined the influence of different levels of ϵ, which perturb the data through the Randomized Response (RR) mechanism, on the server's ability to suggest input parameters for clustering algorithms. In this experiment, the server receives the same amount of data while varying the value of ϵ, and its suggestions are evaluated on the joint dataset without any noise. Experimental results, as shown in Tables 5 and 6, reveal a notable consistency in the server's recommendations.

Regardless of the ϵ value, the server consistently proposes around 7 clusters for the first dataset (Obesity dataset) and approximately 3 clusters for the second dataset (Extended Iris dataset). This consistency closely aligns with the established ground truth, indicating a marginal effect of the privacy parameter ϵ on the server's cluster count recommendations. However, it is important to note that the actual quality of the clusters formed is subject to the specific clustering algorithm employed. For instance, in the first dataset (Obesity dataset), clustering algorithms demonstrate varied effectiveness influenced by different privacy budgets (ϵ), shown in Table 5. K-Means excel, achieving high ARI values, reaching up to 1.0 when less noise introduced to data (higher ϵ), but maintain low



(a) **(b)**

Fig. 2. Visual Representation of Clustering Algorithm Performance Across Combined Datasets. This figure illustrates the performance metrics from Table 4 for various clustering algorithms—GMM, DBSCAN, K-Means, and Hierarchical Clustering (HC)—evaluated under conditions of 10% data sharing and a privacy parameter of $\epsilon = 0.1$. Performance metrics including Adjusted Rand Index (ARI), Homogeneity (Homo), Completeness (Comp), Silhouette Score, Calinski-Harabasz Index (CH), and Accuracy are plotted. Algorithms recommended by the server are highlighted with dots, showcasing their superior performance in comparison to others in each dataset scenario.

classification accuracy across all settings, indicating well-defined clusters that do not match predefined labels. Silhouette scores also improve with increased ϵ, suggesting clearer cluster definition. Hierarchical Clustering (HC) shows moderate and stable ARI values around 0.48 but face declines in accuracy under extreme privacy settings, hinting at potential misalignments with actual labels. Gaussian Mixture Models (GMM) record lower ARI and negative silhouette scores, suggesting less effective clustering and poor separation, with fluctuating accuracy that sometimes aligned with class labels under minimal privacy constraints. DBSCAN consistently performs poorly with very low ARI, negative silhouette scores, and minimal accuracy, indicating its unsuitability for this dataset due to its sensitivity to specific parameter settings and data density. In the second dataset (Extended Iris dataset), the performance of clustering algorithms vary significantly under different privacy settings as shown in Table 6. K-Means showcases excellent clustering with ARI values of 0.997 at low and high ϵ levels, though it drops at $\epsilon = 1$, reflecting its sensitivity to privacy settings, despite maintaining high silhouette scores for good cluster separation. However, its consistently low accuracy indicates a misalignment between the clusters and actual class labels. Hierarchical Clustering (HC) remains stable across all metrics and ϵ settings, achieving moderate to high ARI and silhouette scores, and comparatively better accuracy at 0.38, suggesting it aligns more closely with true labels. Gaussian Mixture Models (GMM) exhibit poor performance with negative ARIs and low silhouette scores, with only moderate accuracy, underscoring its challenges in this dataset under privacy constraints. DBSCAN performs poorly, with extremely low ARI, negative silhouette scores, and zero accuracy across all ϵ settings, confirming its unsuitability for the dataset. Overall, the K-Means algo-

rithm excels over others when the server's recommendation was $k = 3$, according to various evaluation metrics. Furthermore, the server's recommendations do not significantly deviate from the original data in both datasets. To understand the behavior of data points in dataset #1, we conducted an analysis by selecting two clusters from the original dataset and applying the RR mechanism with varying ϵ values. This investigation revealed that the RR mechanism effectively maintains the separation between clusters when present.

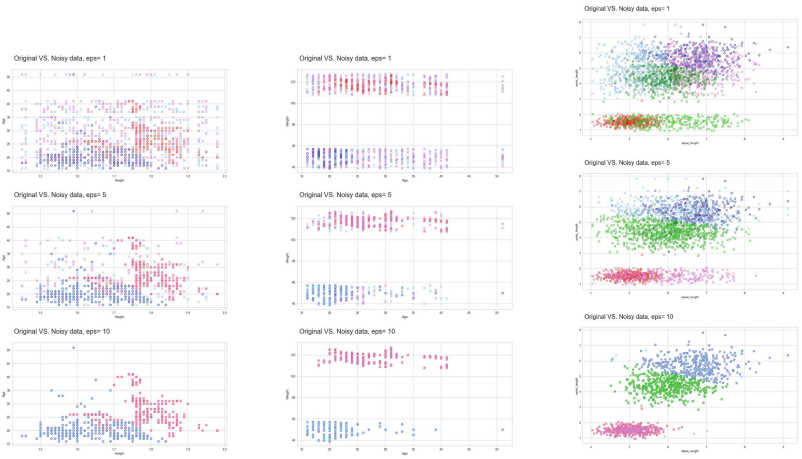


Fig. 3. (a): Contrast in dataset #1 with Overlapping Clusters (ϵ = 1, 5, 10): This part displays the differences between original ('O') and noise-modified ('X') data in closely positioned clusters, colored blue and red. **(b)**: Comparison in dataset #1 with Clear Cluster Gaps (ϵ = 1, 5, 10): Here, the focus is on the impact of the Randomized Response (RR) method on data (original 'O', noisy 'X') in maintaining cluster gaps despite noise variations, balancing privacy with data structure integrity. **(c)**: Original vs. Noisy Data in dataset #2 (ϵ = 1, 5, 10): This section compares original ('O') and noise-affected ('X') data at different privacy levels, using blue, red, and green to show cluster separation effectiveness via the RR mechanism. Note: Plots can be zoomed in for clearer visualization. (Color figure online)

A comparison of Figs. 3a and 3b illustrates that the distinction between two randomly selected clusters is retained even when the data is subjected to different ϵ values. This finding is significant as it demonstrates that despite lower ϵ values possibly leading to a more sparse appearance of the data, the server is still capable of accurately identifying two distinct clusters. This is because the RR mechanism ensures that data points are redistributed within a range akin to their original positions. Furthermore, an analysis of Table 4 shows that the server's recommendations for second dataset closely mirror the original data. An exploration involving a comparison of the original and RR-perturbed data points

across different ϵ values, as demonstrated in Fig. 3c, indicates that in two out of the three clusters in dataset #2, data points overlap without a clear gap, while the third cluster's points are notably distanced from the others. This observation reinforces the notion that the RR mechanism is capable of preserving existing gaps between clusters for various ϵ values.

These results underscore the RR mechanism's proficiency in safeguarding the intrinsic structure of the data while incorporating elements of privacy protection. By effectively maintaining the relative distances between data points, the server is enabled to provide precise recommendations for the number of clusters, despite the noise caused by different ϵ values. This highlights the RR mechanism's balance in protecting data privacy while ensuring the accuracy of clustering algorithm suggestions in a privacy-conscious data analysis setting.

Table 5. Differential Impact of Privacy Levels on Clustering Algorithms in the dataset #1. This table explores the performance variations (measured through ARI, Silhouette, and Accuracy) of four distinct clustering algorithms (K-Means, HC, GMM, DBSCAN) at different privacy budget levels ($\epsilon = 0.1, 1, 5$) with a consistent data sharing percentage (10%).

Algorithm	Shared	ϵ	K	ARI	Silhouette	Accuracy
K-Means	10%	0.1	k = 8	0.75	0.41	0.18
K-Means	10%	1	k = 8	0.75	0.41	0.18
K-Means	10%	5	k = 7	1	0.44	0.15
HC	10%	0.1	k = 8	0.481	0.39	0.005
HC	10%	1	k = 7	0.482	0.41	0.17
HC	10%	5	k = 8	0.482	0.41	0.005
GMM	10%	0.1	k = 6	0.185	−0.0143	0.201
GMM	10%	1	k = 8	0.2069	−0.072	0.05
GMM	10%	5	k = 6	0.2008	−0.007	0.14
DBSCAN	10%	0.1	k = 10	0.017	−0.504	0.005
DBSCAN	10%	1	k = 10	0.017	−0.504	0.005
DBSCAN	10%	5	k = 10	0.017	−0.504	0.005

Impact of Shared Data Volume on Server Suggestions: In exploring the influence of shared data volume on clustering algorithm suggestions for both datasets 1 and 2, the results consistently indicate that varying the proportion of data shared with the server does not significantly impact the server's recommendations for clustering input parameters. To investigate this, we conduct experiments where varying amounts of data are shared with the server while keeping the privacy parameter (ϵ) unchanged. This observation is consistent across both datasets and all tested algorithms, as shown in Tables 7 and 8.

Table 6. Influence of Privacy Settings on Clustering Recommendations in the dataset #2. This table details how varying privacy budgets (ϵ = 0.1, 1, 5) affect the recommendations for clustering parameters and subsequent algorithm performance (ARI, Silhouette, and Accuracy) for multiple clustering algorithms (K-Means, HC, GMM, DBSCAN), all with a consistent 10% data sharing arrangement.

Algorithm	Shared	ϵ	K	ARI	Silhouette	Accuracy
K-Means	10%	0.1	k = 3	0.997	0.52	0
K-Means	10%	1	k = 2	0.44	0.57	0.18
K-Means	10%	5	k = 3	0.997	0.52	0
HC	10%	0.1	k = 3	0.84	0.51	0.38
HC	10%	1	k = 3	0.84	0.51	0.38
HC	10%	5	k = 3	0.84	0.51	0.38
GMM	10%	0.1	k = 3	−0.0003	0.021	0.3
GMM	10%	1	k = 2	−0.0004	0.051	0.34
GMM	10%	5	k = 3	−0.0003	0.021	0.3
DBSCAN	10%	0.1	k = 6	0.003	−0.6	0
DBSCAN	10%	1	k = 6	0.003	−0.6	0
DBSCAN	10%	5	k = 6	0.003	−0.6	0

Table 7. Impact of Data Sharing Proportions on Clustering Algorithms' Performance in the dataset #1. This table evaluates how different proportions of data shared with the server (10%, 30%, 50%) influence the clustering outcomes (ARI, Silhouette, and Accuracy) for various algorithms (K-Means, HC, GMM, DBSCAN) at a fixed privacy parameter ($\epsilon = 0.1$).

Algorithm	Shared	ϵ	K	ARI	Silhouette	Accuracy
K-Means	10%	0.1	k = 8	0.75	0.41	0.18
K-Means	30%	0.1	k = 8	0.75	0.41	0.18
K-Means	50%	0.1	k = 8	0.75	0.41	0.18
HC	10%	0.1	k = 8	0.481	0.39	0.005
HC	30%	0.1	k = 8	0.481	0.39	0.005
HC	50%	0.1	k = 8	0.0.481	0.39	0.005
GMM	10%	0.1	k = 6	0.185	−0.143	0.201
GMM	30%	0.1	k = 8	0.175	−0.111	0.18
GMM	50%	0.1	k = 5	0.169	−0.001	0.23
DBSCAN	10%	0.1	k = 10	0.017	−0.504	0.005
DBSCAN	30%	0.1	k = 10	0.017	−0.504	0.005
DBSCAN	50%	0.1	k = 10	0.017	−0.504	0.005

For the first dataset, the K-Means algorithm maintains the same ARI, Silhouette, and Accuracy metrics across different data sharing proportions, suggesting that its performance remains stable despite changes in the volume of data shared. Similarly, Hierarchical Clustering (HC), Gaussian Mixture Models (GMM), and DBSCAN show consistent performance metrics across different data sharing amounts, further supporting the notion that the quality of clustering recommendations does not deteriorate with reduced data sharing. In the second dataset, similar patterns emerge. For instance, the K-Means algorithm and HC adjust their suggested number of clusters slightly depending on the data share, but the overall performance metrics such as ARI and Silhouette remain relatively stable. This trend continues with GMM and DBSCAN, which also show little variation in performance across different data sharing proportions.

These findings suggest that the server is capable of providing robust and reliable recommendations for clustering parameters regardless of the amount of data shared, enabling effective clustering outcomes even when data owners choose to share minimal data. This is particularly advantageous in scenarios where data privacy is a concern, as it allows data owners to restrict the amount of shared data without compromising the effectiveness of the clustering process. Overall, the server's ability to consistently suggest appropriate clustering parameters across varying data proportions demonstrates its effectiveness and reliability in guiding the clustering process under different data availability conditions.

Table 8. Analysis of Server Recommendations for Clustering Parameters Based on Data Sharing Amounts in the second Dataset. This table examines the influence of varying amounts of data shared (10%, 30%, 50%) on server-suggested clustering parameter (k) and their resulting ARI, Silhouette, and Accuracy metrics at a constant privacy parameter ($\epsilon = 0.1$).

Algorithm	Shared	ϵ	K	ARI	Silhouette	Accuracy
K-Means	10%	0.1	k = 3	0.997	0.52	0
K-Means	30%	0.1	k = 2	0.44	0.57	0.18
K-Means	50%	0.1	k = 2	0.44	0.57	0.18
HC	10%	0.1	k = 3	0.84	0.51	0.38
HC	30%	0.1	k = 2	0.55	0.52	0.66
HC	50%	0.1	k = 2	0.55	0.52	0.66
GMM	10%	0.1	k = 3	−0.0003	0.021	0.3
GMM	30%	0.1	k = 2	−0.0004	0.051	0.32
GMM	50%	0.1	k = 2	−0.0004	0.051	0.32
DBSCAN	10%	0.1	k = 6	0.003	−0.6	0
DBSCAN	30%	0.1	k = 6	0.003	−0.6	0
DBSCAN	50%	0.1	k = 6	0.003	−0.6	0

7 Privacy Analysis: Membership Inference Attack

Membership inference attacks (MIA) are techniques used to determine whether specific individual data was included in a dataset. These attacks pose significant privacy risks, especially when datasets contain sensitive information. Our goal is to minimize these risks for individuals whose data is part of a dataset shared with others. To enhance data privacy, only a portion of the dataset, even in its noisy format, is shared with the server. It has been observed that the likelihood of successful membership inference attacks is inversely related to the amount of noise added to the dataset. We divide the data into two groups to assess the impact of these attacks:

Case Group: This group contains data from specific number of individuals (150 for first dataset and 100 for second dataset) and represents the subset of the dataset that is shared with server, thus exposed to potential membership inference attacks.

Control Group: This group includes data that remains entirely internal and is not shared with the server. It serves as a benchmark to gauge the risk of data exposure. We address membership inference attacks by computing a threshold that determines whether an individual's data is likely part of the training dataset based on similarity between shared and unshared data. Similarity exceeding this threshold indicates a risk of data exposure through membership inference.

Our detailed analysis is visually represented in Fig. 4, demonstrating the impact of the privacy parameter (ϵ), with increased ϵ values reducing the noise and thereby increasing the risk of data identification.

Our findings show that as ϵ increases, the risk of membership inference attacks rises, indicating less noise leads to higher identification likelihood. Therefore, limiting shared data and augmenting it with noise is essential to reduce

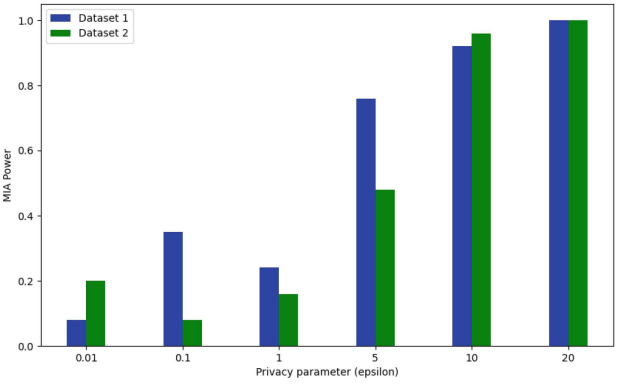


Fig. 4. Analysis of Membership Inference Attack Risks: This figure illustrates the increasing likelihood of data identification in two datasets as privacy parameters (ϵ) increase. The blue bars represent dataset #1, and the green bars represent dataset #2, highlighting the direct correlation between reduced noise levels and heightened data vulnerability. (Color figure online)

these risks, necessitating a strategic approach to balance data utility and privacy in collaborative clustering.

8 Conclusion

This study aims to identify optimal input parameters for four clustering algorithms to facilitate collaborative clustering among multiple data owners. Introducing a semi-trusted third party improves clustering reliability and accuracy by recommending optimal algorithms and parameters. Results show that neither the amount of perturbed data shared nor the privacy budget (ϵ) significantly impacts the server's recommendations.

Furthermore, this study conducts an analysis of membership inference attacks to evaluate the vulnerability of the system. As the privacy budget (ϵ) increases, the power of membership inference attacks also increases. This indicates that higher levels of privacy budget compromise the effectiveness of privacy protection, making it easier for attackers to infer whether an individual's data is part of the shared dataset.

These findings emphasize the need for careful consideration of privacy-preserving mechanisms and the importance of maintaining an appropriate balance between privacy protection and utility. While the server's suggestions for input parameters remain consistent regardless of the amount of perturbed data or the privacy budget, the potential risks associated with membership inference attacks highlight the need to adopt appropriate safeguards and mitigation strategies. Protecting the privacy of individuals and ensuring the security of collaborative clustering processes should be key priorities in future research and system design.

Acknowledgement. The work was partly supported by the National Library of Medicine of the National Institutes of Health under Award Number R01LM013429, the National Science Foundation (NSF) under grant numbers 2141622, 2050410, 2200255, and OAC-2112606, and Cisco Research.

References

1. Extended iris dataset. https://www.kaggle.com/datasets/samybaladram/iris-dataset-extended
2. Estimation of obesity levels based on eating habits and physical condition. UCI Machine Learning Repository (2019). https://doi.org/10.24432/C5H31Z
3. Anikin, I.V., Gazimov, R.M.: Privacy preserving DBSCAN clustering algorithm for vertically partitioned data in distributed systems. In: Proceedings International Siberian Conference Control Communication (SIBCON) (2017)
4. Baby, V., Chandra, N.S.: Distributed threshold k-means clustering for privacy preserving data mining. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2016)

5. Bi, R., Guo, D., Zhang, Y., Huang, R., Lin, L., Xiong, J.: Outsourced and privacy-preserving collaborative k-prototype clustering for mixed data via additive secret sharing. IEEE Internet Things J. **10**(18), 15810–15821 (2023). https://doi.org/10.1109/JIOT.2023.3266028
6. Bozdemir, B., Canard, S., Ermis, O., Mollering, H., Onen, M., Schneider, T.: Privacy-preserving density-based clustering. In: Proceedings ACM Asia Conference Computer and Communications Security (ASIACCS) (2021)
7. Cornuéjols, A., Wemmert, C., Gançarski, P., Bennani, Y.: Collaborative clustering: why, when, what and how. Inf. Fus. **39**, 81–95 (2018). https://doi.org/10.1016/j.inffus.2017.04.008
8. Costello, C., et al.: Geppetto: versatile verifiable computation. In: Proceedings IEEE Symposium on Security and Privacy (2015)
9. Davidson, S., Khanna, S., Milo, T., Panigrahi, D., Roy, S.: Provenance views for module privacy. In: Proceedings 30th ACM SIGMOD-SIGACT-SIGART Symposium Principles Database Systems (PODS) (2011)
10. Davidson, S., Khanna, S., Roy, S., Stoyanovich, J., Tannen, V., Chen, Y.: On provenance and privacy. In: Proceedings 14th International Conference Database Theory (ICDT) (2011)
11. De, I., Tripathy, A.: A secure two party hierarchical clustering approach for vertically partitioned data set with accuracy measure. In: Thampi, S.M., Abraham, A., Pal, S.K., Rodriguez, J.M.C. (eds.) Recent Advances in Intelligent Informatics, pp. 153–162. Springer, Cham (2014)
12. Dey, S., Zinn, D., Ludäscher, B.: ProPub: towards a declarative approach for publishing customized, policy-aware provenance. In: Proceedings 23rd International Conference Science Statistics Database Management (SSDBM) (2011)
13. Dwork, C., Smith, A., Steinke, T., Ullman, J.: Exposed! a survey of attacks on private data. Annu. Rev. Stat. Appl. **4**, 61–84 (2017)
14. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings International Conference Knowledge Discovery Data Mining (KDD) (1996)
15. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis. Wiley (2011)
16. Fisher, R.A.: Iris. UCI Machine Learning Repository (1988). https://doi.org/10.24432/C56C76
17. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Inf. Theory (1975)
18. Gymrek, M., McGuire, A.L., Golan, D., Halperin, E., Erlich, Y.: Identifying personal genomes by surname inference. Science **339** (2013)
19. Hamidi, M., Sheikhalishahi, M., Martinelli, F.: Privacy preserving expectation maximization (EM) clustering construction. In: Distributed Computing and Artificial Intelligence, 15th International Conference 15, pp. 255–263 (2019)
20. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data (2005)
21. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
22. Lin, X., Clifton, C., Zhu, M.: Privacy-preserving clustering with distributed EM mixture modeling. Knowl. Inf. Syst. (2005)
23. Liu, J., Xiong, L., Luo, J., Huang, J.Z.: Privacy preserving distributed DBSCAN clustering. Trans. Data Privacy (2013)
24. Meng, X., Papadopoulos, D., Oprea, A., Triandopoulos, N.: Private two-party cluster analysis made formal and scalable. arXiv preprint arXiv:1904.04475v2 (2019)

25. Mohassel, P., Rosulek, M., Trieu, N.: Practical privacy-preserving k-means clustering (2020)
26. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings IEEE Symposium Security Privacy (SP) (2008). https://doi.org/10.1109/SP.2008.33
27. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Proceedings 30th IEEE Symposium Security Privacy (SP), pp. 173–187 (2009). https://doi.org/10.1109/SP.2009.22
28. Nergiz, M.E., Atzori, M., Clifton, C.: Hiding the presence of individuals from shared databases. In: Proceedings ACM SIGMOD International Conference Management Data (SIGMOD) (2007). https://doi.org/10.1145/1247480.1247554
29. Power, J., Beresford, A.: SoK: managing risks of linkage attacks on data privacy. In: Proceedings Privacy Enhancement Technologies (PoPETS) (2023). https://doi.org/10.56553/popets-2023-0043
30. Pyrgelis, A., Troncoso, C., De Cristofaro, E.: Measuring membership privacy on aggregate location time-series. Proc. ACM Meas. Anal. Comput. Syst. **4**(2) (2020). https://doi.org/10.1145/3392154
31. Rahman, M.S., Basu, A., Kiyomoto, S.: Towards outsourced privacy-preserving multiparty DBSCAN. In: 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 225–226 (2017)
32. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. Data Min. Knowl. Discov. (1998)
33. Wang, C., Xu, J., Tan, S., Yin, L.: Privacy-preserving cloud-edge collaborative k-means clustering model in IoT. In: Yang, H., Lu, R. (eds.) Front. Cyber Secur., pp. 655–669. Springer, Singapore (2024)
34. Wu, W., Liu, J., Wang, H., Hao, J., Xian, M.: Secure and efficient outsourced k-means clustering using fully homomorphic encryption with ciphertext packing technique. IEEE Trans. Knowl. Data Eng. **33**(10), 3424–3437 (2021)
35. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005). https://doi.org/10.1109/TNN.2005.845141
36. Xu, X., Ester, M., Kriegel, H.P., Sander, J.: A distribution-based clustering algorithm for mining in large spatial databases. In: Proceedings International Conference Data Engineering (ICDE) (1998)
37. Yuan, C., Yang, H.: Research on k-value selection method of k-means clustering algorithm. J. **2**(2), 226–235 (2019)