BPGT: A Novel Privacy-Preserving K-Means Clustering Framework to Guarantee Local d_χ-privacy

Fan Chen[1], Bizhi Lei[1], Jielu Zhu[1], Xiaoyu Zhu[2], and Shaobo Zhang[1(✉)]

[1] Hunan University of Science and Technology, XiangTan 411201, China
2194457208@qq.com, 2945667409@qq.com, 1927895718@qq.com, shaobozhang@hnust.edu.cn
[2] Changsha University, ChangSha 410022, China
zhuxiaaoyu@ccsu.edu.cn

Abstract. Privacy-preserving data analytics has received much attention in recent years. However, existing local privacy-preserving mechanisms struggle to address data generation bias and distribution shifts, which makes it difficult to achieve the optimal privacy-utility trade-off in downstream data analysis tasks. In this paper, we propose a privacy-preserving K-means clustering framework: Bounded Perturbation Generation Mechanism and TK-means (BPGT), aiming to balance between robust privacy preservation and high data utility. On the one hand, we propose the bounded perturbation generation mechanism, which combines bounded noise sampling with synthetic data generation based on gradient descent. This mechanism mitigates the cumulative error problem inherent in traditional local differential privacy methods while ensuring d_χ-privacy. On the other hand, we propose the TK-means algorithm integrating the T-Mixture Model and the Expectation Maximization algorithm to enhance the robustness of the K-means algorithm to complex data distributions and outliers. To evaluate the performance of our method, we provide theoretical guarantees on privacy preservation and data utility, and verify its effectiveness through extensive experiments. The results show that BPGT improves on average 9.11% and 7.51% in clustering accuracy and 76.35% in error control compared to existing methods.

Keywords: K-means clustering · Data utility · Local differential privacy · d_χ-privacy · T-Mixture Model

1 Introduction

In the era of big data, it is crucial to adopt effective data analysis methods to obtain valuable information from massive user data [1,2]. K-means clustering as a classical data analysis method has been widely used in the fields of market analysis [3], anomaly detection [4] and so on. Improving the quality of K-means

clustering results has become a primary concern. In addition, the process of K-means clustering analysis leaks sensitive information such as users' consumption habits and GPS locations [6,7], According to the Statista 2024 report, around 60% of organizations worldwide face legal risks due to privacy violation issues during data analytics. Therefore, striking a balance between high-quality clustering and privacy protection is a key challenge in a privacy-preserving clustering.

To address this challenge in privacy-preserving K-means clustering, Local Differential Privacy (LDP) [8,9] provides a solution. LDP protects user privacy while guaranteeing data utility by giving some upper bounds on the statistical difference between the distributions of perturbed responses based on any two possible data records in the domain prior to uploading the data. availability, which promotes the research of LDP-based clustering [17,18,19]. However, the uniform noise injection of LDP conflicts with the heterogeneity of real-world data and the different analyzed values recorded. This poses two problems for LDP-based clustering studies: (1) the core data points, which are critical to the cluster centroids and boundaries, receive the same noise as the peripheral or noisy points, reducing the utility of the data; (2) in high-dimensional spaces, the cumulative noise distorts the separability of the clusters, thus hindering effective clustering. Therefore, there is an urgent need for a privacy-preserving framework customized for clustering tasks to address these two issues.

The d_χ-privacy proposed by Chatzikokolakis et al. [13] successfully bridges two shortcomings of LDP clustering studies. d_χ-privacy is a generalization of differential privacy, and unlike CDP (where $d(X, X') = 1$) or LDP (which has a constant constraint on the probability density function), d_χ-privacy increases the constraint linearly with the distance of the data points, and thus different distance metrics can be defined according to specific application scenarios, thus providing stronger privacy protection, such as Geo-Indistinguishability [14] and metric-LDP [15]. Due to the strong applicability and scalability of d_χ-privacy, Yang et al [24] first proposed Bounded Perturbation Mechanism (BPM) to apply d_χ-privacy to K-means, which improves the clustering quality, but the two-step sampling introduces data bias and outlier sensitivity. Alishahi et al. [25] proposed an nD-Laplace clustering method that extends GP to balance privacy and utility, but Laplace noise and grid redrawing distort the data distribution, especially in the case of high-dimensional data or low privacy budget.

Overall, existing d_χ-privacy-based clustering methods still have the following limitations:

(1) **Data generation bias**: Sampling and perturbation methods (e.g., multi-step sampling or random response) can amplify noise and outliers, skewing the generated data from the true distribution and degrading downstream clustering performance.
(2) **Data distribution shifts**: In d_χ-privacy-based K-means clustering, the privacy radius causes distribution bias. Noise perturbation alters data characteristics (e.g., shape, density, cluster centers), creating discrepancies between perturbed and original distributions that impair clustering accuracy.

To address the challenge of balancing privacy and clustering utility amidst data generation biases and distributional shifts, we propose BPGT, a novel framework that integrates two components: a Bounded Perturbation Generation Mechanism (BPGM) on the user side and a TK-means algorithm on the server side.The BPGM reduces bias by perturbing the data holistically instead of dimensionally to minimize minimizes statistical errors caused by noise, and uses gradient descent to generate synthetic data that preserves the features of the original data in a bounded domain, thus improving utility. On the server side, TK-means utilizes a heavy-tailed T-Mixture Model (TMM) and an Expectation Maximization (EM) algorithm to improve the robustness of K-means to outliers and varying distributions. These components work together to optimize privacy and utility, effectively addressing the challenges of data bias and distributional shifts.

The main contributions of this paper are as follows:

- We propose the Bounded Perturbation Generation Mechanism (BPGM), which uses bounded noise sampling and gradient descent to perturb overall user data, generating synthetic data that aligns with original features. This reduces noise-induced variance in LDP clustering while ensuring d_χ-privacy, enhancing data utility.
- We introduce the TK-means algorithm, leveraging the T-Mixture Model and EM algorithm to address traditional K-means sensitivity to heavy-tailed data and outliers, improving clustering stability and generalization.
- We provide rigorous proofs of privacy and utility for BPGT, validated through extensive experiments on real-world datasets. The results demonstrate that BPGT achieves average improvements of 9.11% in Adjusted Rand Index (ARI), 7.51% in Normalized Mutual Information (NMI), and 76.35% in Relative Error (RE) compared to existing methods.

2 Related work

In this section, we outline the evolution of Local Differential Privacy (LDP), with an emphasis on LDP-based K-means clustering methods and the application and recent developments of d_χ-privacy in privacy protection.

2.1 K-means clustering based on LDP

Local Differential Privacy (LDP) [8,9] is a robust privacy framework that perturbs data locally before server upload, eliminating reliance on trusted third parties and offering stronger privacy guarantees than traditional methods.

LDP-based K-means clustering has gained significant attention. Erlingsson et al.'s RAPPOR [16] pioneered LDP by using randomized responses to protect user data, though it sacrifices accuracy due to cumulative effect of noise. Su et al.'s EUGkM [10,17] blends interactive and non-interactive approaches with RAPPOR to balance privacy and clustering quality, but struggles with

computational overhead in high-dimensional data. Xia et al. [11] addressed this by perturbing binary-transformed feature vectors, yet frequent user-server interactions increase communication and computation costs.Nissim and Stemmer's GoodCenters [12] employs locally sensitive hashing to group nearby points, reducing collisions for distant ones. Stemmer and Kaplan [18] refined this, cutting interaction rounds and multiplicative error, with later work [19] reducing additive error.

At present, the LDP has developed in several directions. Song et al.'s APLDP [21] adapts privacy via RAPPOR or k-RR based on user preference. Feng et al. [20] proposed OC and OP methods, minimizing mean square and maximum errors, respectively, to balance privacy and utility. Fu et al.'s GC-LDP [22] uses multidimensional grids, non-uniform partitioning, and novel perturbation to reduce iterations, noise, and enable multi-density clustering. Liebenow et al.'s DPM [23] recursively partitions data, minimizing sensitive leaks and rivaling non-private K-means results.

Recent advancements in LDP clustering have bolstered its practicality by enhancing the privacy-utility balance, yet challenges such as computational efficiency, noise-induced accuracy loss, and adaptability to diverse data distributions persist, necessitating further innovation in LDP-based K-means clustering.

2.2 d_χ-privacy and its application Advances

Chatzikokolakis et al. [13] proposed d_χ-privacy, extending differential privacy to arbitrary distance metrics, offering a flexible privacy framework that boosts data utility across applications. Miguel E. Andrés et al. [14] built on this with Geo-Indistinguishability (GP), adding controlled noise to location data to ensure privacy without compromising service quality. Liang and Yi [28] proposed Centralized Geographic Privacy (CGP) as an alternative, enhancing composability and reducing noise for high-dimensional data and complex queries, outperforming GP in privacy-utility balance despite its centralized nature. Alvim et al. [15] developed metric-LDP for metric spaces, improving utility at equivalent privacy levels. Fernandes et al. [36] analyzed the privacy-utility trade-off under metric differential privacy using Quantitative Information Flow, exploring diverse loss functions and mechanisms. Athanasiou and Chatzikokolakis [26] enhanced metric-LDP with a shuffler, solved the problem of shuffler failure.

Yang et al. proposed the BPM method [24] first applied d_χ-privacy to K-means clustering, improving quality while ensuring privacy, though its two-step sampling biases data, reducing utility and slowing convergence. Alishahi et al.'s nD-Laplace [25] extends GP with a grid-remapping mechanism, better balancing privacy and utility. Zhou et al.'s mLDP-KDE [27] integrates locally sensitive hashing and generalized random response for accurate Kernel Density Estimation (KDE) under privacy, but high-dimensional hash perturbation can impair KDE accuracy. Oluwaseyi et al. [35] proposed hyperbolic space sampling for d_χ-privacy, optimizing the privacy-utility trade-off in downstream machine learning.

Studies on d_χ-privacy-based K-means clustering offer insights into balancing utility and privacy, but issues such as sampling and perturbation biases, distribu-

tion shifts, and limited algorithm adaptability to diverse data remain, requiring innovative solutions to preserve both privacy and utility.

3 Preliminaries

Table 1. Notations.

Notation	Description	Notation	Description
D	Dateset	ℓ	Noise upper bound
$\hat{d}$	Noise distance sampled	δ_i	Degrees of freedom of TMM
r_i	Raw user data	m_k	The mean vector of the k-th component
$\hat{r}_i$	User perturbation data	S_k	Covariance matrix for the k-th component
ϵ	Privacy budget	$z_{n\,k}$	Indicates whether x_n belong to the k-th component
$d(\cdot,\cdot)$	Distance function	u_n	Missing value

3.1 Local Differential Privacy

Definition 1(ϵ-local differential privacy): Given a randomized algorithm $M{:}D \to \mathbb{R}$, M satisfies ϵ-local differential privacy if and only if for any pair of distinct input values $\nu, \nu' \in D$ and for any possible non-empty output set $S \subseteq M(D)$. have:

$$\Pr[M(v) \in S] \leq e^{\epsilon} \Pr[M(v') \in S]. \tag{1}$$

Then M is said to satisfy ϵ-local differential privacy. where ϵ is the privacy budget, the smaller its value, the stronger the privacy protection and the smaller the data availability. For local differential privacy, each data record performs a randomization mechanism M on the user side.

3.2 d_χ-privacy

Unlike local differential privacy, d_χ-privacy [13,14,15] integrates traditional differential privacy with a distance metric, offering stronger privacy protection for records closer together and weaker protection for those farther apart, rather than uniformly distributing the privacy budget ϵ across identical records.

Definition 2(d_χ–privacy): Given a randomized algorithm $M{:}\chi \to \mathbb{R}$, $M(\nu)$ follows the distribution of the probability density function $f_v : \mathbb{R} \to \mathbb{R}_{\geq 0}$, and the randomized algorithm M satisfies d_χ–privacy if and only if given $\nu, \nu' \in \chi$ and $t \in \mathbb{R}$ have:

$$f_v^{\ell}(t) \leq e^{d_\chi(\nu,\nu')} f_{v'}^{\ell}(t). \tag{2}$$

To incorporate a measure of privacy loss, the privacy budget can be embedded into the distance function, note that we do not use $d_\chi(\nu, \nu')$ as the original

distance function, but rather $d_\chi(\nu, \nu') = \epsilon d(\nu, \nu')$, and thus fixing $\epsilon > 0$, the randomized algorithm M satisfies $\epsilon d(\nu, \nu') - privacy$ if and only if for any $\nu, \nu' \in \chi$, $t \in \mathbb{R}$ has:

$$f_v^\ell(t) \leq e^{\epsilon d(\nu,\nu')} f_{v'}^\ell(t). \tag{3}$$

In this paper, we consider the distance as Euclidean distance because the K-means algorithm divides the data points based on the Euclidean distance. We will ϵd_E–privacy stand for d_χ–privacy with respect to the Euclidean distance, in different missions this distance can of course be the Manhattan distance, the Chebyshev distance and so on.

3.3 K-means Clustering

K-means is a widely used clustering algorithm in data mining, image processing, and machine learning. It assigns each data point to the nearest of K initial cluster centroids, then updates each centroid as the mean of its assigned points. This process iterates until centroids stabilize or a set iteration limit is reached.

For a sample set $D = \{x_1, x_2, \ldots, x_n\}$ with initial centroids $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$, where $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$, K-means minimizes the squared error of the partition $C = \{C_1, C_2, \ldots, C_K\}$ via the objective function:

$$E = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|_2. \tag{4}$$

4 BPGT Framework

The BPGT framework (Fig 1) consists of two components, the user side and the server side. The user side perturbs the data and the server side clusters the perturbed data. Key terms are defined in Table 1, and we assume that the data is located at $D = [0, 1]^d$ and any data can be adjusted by normalization.

User Side: collects the raw data r and perturbs it to $\hat{r}$ using the BPGM mechanism.BPGM increases bounded noise (limited by ℓ) through noise sampling and gradient descent, ensuring d_χ-privacy while keeping $\hat{r}$ close to r. The perturbed $\hat{r}$ will be securely transmitted to the server.

Server side: the server collects all perturbed $\hat{r}$ from the user and applies the TK-means algorithm K clusters.

The full process is outlined in Algorithm 1.

4.1 User side

We propose the Bounded Perturbation Generation Mechanism (BPGM) to achieve d_χ-privacy. BPGM differentiates true and perturbed data records using Euclidean distance, preserving privacy while maximizing data utility, such as clustering accuracy and distribution consistency. By adding bounded noise and distance thresholds, BPGM limits the gap between perturbed and original data,

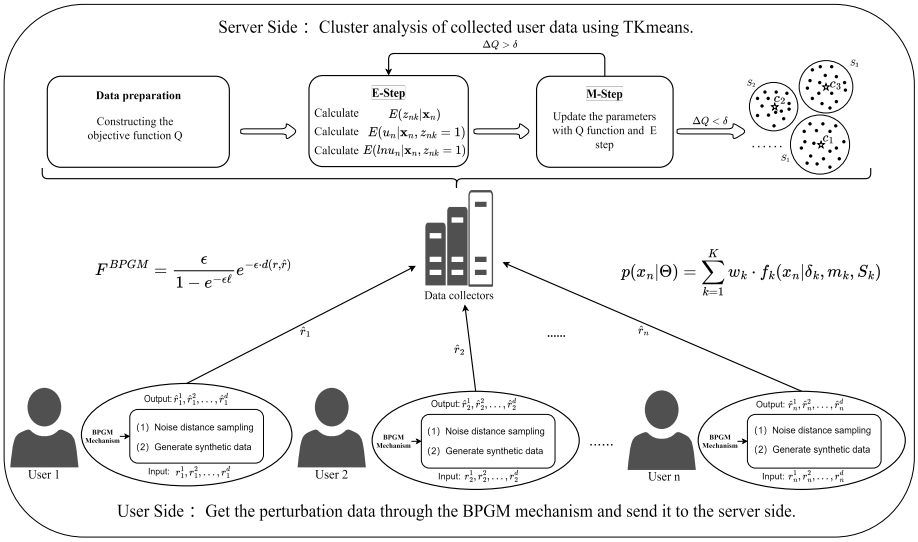

Fig. 1. BPGT Framework

Algorithm 1 BPGT Framework

Input: User's data record r , privacy budget ϵ, noise upper bound ℓ
Output: Clustering results $C_1, C_2, \ldots, C_k$ and $S_1, S_2, \ldots, S_k$

```
[User side]
for each r_i ∈ D do
    Execute r̂_i ← BPGM(r_i);
    Report r̂_i to Server side;
end for
[Server side]
Perform TK-means algorithm to get clustering results C_1, C_2, ..., C_k and S_1, S_2, ..., S_k;
```

preventing adversaries from inferring the original records or their exact distances due to the randomness and constraints, thus ensuring d_χ-privacy. BPGM comprises two steps: bounded noise distance sampling and synthetic data generation.

Boundary noise distance sampling: The BPGM mechanism resembles the Laplace mechanism but perturbs the distance property rather than individual attributes. Given a privacy budget ϵ and true record r, the perturbed record $\hat{r}$ satisfies the probability density function:

$$F^{BPGM} = \frac{\epsilon}{1-e^{-\epsilon\ell}} e^{-\epsilon \cdot d(r,\hat{r})} \tag{5}$$

Where $d(r,\hat{r}) \leq \ell$, and ℓ is a server-defined upper bound on noise distance. The normalization factor $\frac{\epsilon}{1-e^{-\epsilon\ell}}$ ensures sampling within $[0,\ell]$. Varying ℓ adjusts

noise levels for a fixed ϵ, offering flexibility for different privacy needs. The noise distance $\hat{d} = d(r, \hat{r})$ is sampled locally via inverse transform sampling, meeting local differential privacy requirements.

When selecting the noise distance upper bound ℓ, the scale characteristics of the data set should be considered. For data sets normalized to $[0, 1]^d$, ℓ can be set to a certain proportion of the data range (e.g. 0.1 to 1). When the variance of the data set is large or the distance between clusters is small, a smaller ℓ (e.g., $\ell = 1$ can better preserve the cluster structure; on the contrary, for sparsely distributed data, ℓ (e.g., $\ell = 10$) can be increased appropriately to enhance privacy protection. In practice, the optimal ℓ can be selected through cross-validation to balance the privacy budget ϵ and clustering utility.

Synthetic data generation: To generate the perturbed record $\hat{r}$, we model the process as an optimization task using gradient descent. We initialize synthetic data randomly, define the loss function as the difference between the Euclidean distance of real data r and synthetic data $\hat{r}$ and the sampled noise distance $\hat{d}$, then iteratively update $\hat{r}$ via gradient descent until a stopping condition is met:

$$loss = \|r - \hat{r}\|_2 - \hat{d}. \tag{6}$$

Algorithm 2 details this synthetic data generation, illustrated in Fig 2, where r is the original record, $\hat{r}$ is the synthetic record, $\hat{d}$ is the sampled noise distance, and ℓ is the noise sampling upper bound.

The proposed noisy distance sampling and synthetic data generation methods ensure d_χ-privacy, preventing attackers from determining the distance between real and generated data. Synthetic records are generated randomly and updated only based on distance properties. If a synthetic record falls outside the specified data domain, its authenticity may be detectable, but this reveals nothing about the actual value.

4.2 Server side

The perturbed data received by the server retains the distance relationship to the original record, thus preserving structural and clustering properties. Traditional K-means is a variant of the Gaussian Mixture Model (GMM) [29,32], but due to the nature of its fine-tailed distribution [30] it is difficult to deal with outliers, which reduces accuracy.

We propose TK-means, which uses a heavy-tailed T-Mixture Model [31] to enhance robustness to outliers and complex datasets. Unlike the hard clustering of K-means, TK-means employs soft clustering and EM optimization by iterating E-steps and M-steps and optimizing the loss function to improve stability and accuracy.

Data prepation: In TK-means, We assume the dataset $D = \{x_n\}_{n=1}^N$, where $x_n \in \mathbb{R}^p$, follows a T-Mixture distribution. The heavy-tailed nature of this distri-

Algorithm 2 Synthetic data generation

Input: User's data record r, noisy distance $\hat{d}$, learning rate α, threshold δ, First momentum decay rate β_1, Second momentum decay rate β_2
Output: synthetic data record $\hat{r}$

Sample the noisy distance $\hat{d}$ following F^{BPGM} with sample space $[0, \ell]$
$\hat{r} \leftarrow$ initialize a synthetic data record randomly, $x \leftarrow$ variate vector $(x_1, x_2, ..., x_d)$
$L \leftarrow \|r - x\|_2, E \leftarrow loss(\hat{r}, \hat{d}), t \leftarrow Iterations$
while $E > \delta$ **do**
  $m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * \nabla \mathbf{loss}(\hat{r}_t, \hat{d})$;
  $v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * \nabla \mathbf{loss}(\hat{r}_t, \hat{d})^2$;
  $\hat{m}_{t+1} = m_t / (1 - \beta_1^t)$;
  $\hat{v}_{t+1} = v_t / (1 - \beta_2^t)$;
  $\alpha_t = \alpha * \frac{\sqrt{1-(\beta_2)^t}}{1-(\beta_1)^t}$;
  $\hat{r}_{t+1} = \hat{r}_t - (\alpha_t * \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + e})$;
  $E = loss(\hat{r}_{t+1}, \hat{d})$;
  if $E > \delta$ **then**
    Break;
  end if
end while
return $\hat{r}$;

bution enables better handling of outliers compared to Gaussian distributions. We define the following parameters:

- Mixing coefficients: w_k, satisfying $w_k = \frac{1}{K}$ and $\sum_{k=1}^{K} w_k = 1$.
- Mean vectors: $m_k \in \mathbb{R}^p$.
- Covariance matrices: $S_k \in \mathbb{R}^{p \times p}$.
- Degrees of freedom: δ_k, controlling the tail behavior of the distribution.

The probability density function of the T-Mixture Model is expressed as:

$$p(x_n|\Theta) = \sum_{k=1}^{K} w_k \cdot f_k(x_n|\delta_k, m_k, S_k). \tag{7}$$

Where $\Theta = \{w_k, \delta_k, m_k, S_k \mid k = 1, \ldots, K\}$, and $f_k(x_n|\delta_k, m_k, S_k)$ is the probability density function of the k-th T-distribution component.

For simplification, we assume $S_k = \beta I$ (where β is a scalar, I is the unit matrix), and $\delta_k = \delta$. We introduce latent variables z_{nk} (indicating whether x_n belongs to the k-th cluster) and missing data u_n. The complete data is defined as:

$$\boldsymbol{y} = (\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}). \tag{8}$$

Where $\boldsymbol{x} = (x_1, \ldots, x_N)$, $\boldsymbol{z} = (z_1, \ldots, z_N)$, and $\boldsymbol{u} = (u_1, \ldots, u_N)$.

The conditional distributions are assumed as:

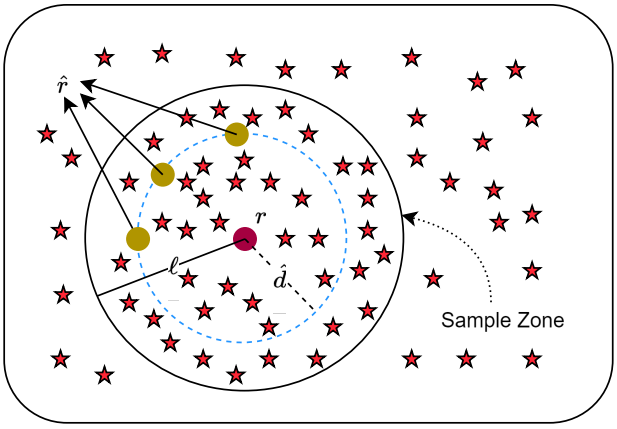

Fig. 2. Schematic diagram of synthetic data generation

$$x_n|u_n, z_{nk} = 1 \sim \mathcal{N}\left(m_k, \frac{\beta I}{u_n}\right), \quad u_n|z_{nk} = 1 \sim \text{Gamma}\left(\frac{\delta}{2}, \frac{\delta}{2}\right). \tag{9}$$

The log-likelihood function for the complete data can be decomposed into two parts:

$$\ln L_{\text{complete}}(\Theta|\boldsymbol{y}) = \ln L_{\text{Gamma}}(\delta|\boldsymbol{u}, \boldsymbol{z}) + \ln L_{\text{Normal}}(m, \beta|\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{z}). \tag{10}$$

Objective Function in EM Algorithm: In the EM algorithm, we aim to maximize the conditional expectation of the complete data log-likelihood:

$$Q(\Theta^\star|\Theta) = \mathbb{E}_q[\ln L_{\text{complete}}(\Theta^\star|\boldsymbol{y})]. \tag{11}$$

Where $\Theta^\star = \{\delta^\star, m^\star, \beta^\star\}$ are the updated parameters. The objective function can be split into:

$$Q(\Theta^\star|\Theta) = Q_1(\delta^\star|\Theta) + Q_2(m^\star, \beta^\star|\Theta). \tag{12}$$

Where:

$$Q_1(\delta^\star|\Theta) = \mathbb{E}_q[\ln L_{\text{Gamma}}(\delta^\star|\boldsymbol{u}, \boldsymbol{z})], \tag{13}$$

$$Q_2(m^\star, \beta^\star|\Theta) = \mathbb{E}_q[\ln L_{\text{Normal}}(m^\star, \beta^\star|\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{z})]. \tag{14}$$

E-Step: Computing Expectations

E-1: Compute Posterior Probability Calculate the posterior probability γ_{nk} that x_n belongs to the k-th cluster:

$$\gamma_{nk} = \mathbb{E}[z_{nk}|x_n] = \frac{w_k f_k(x_n|\delta, m_k, \beta I)}{\sum_{j=1}^{K} w_j f_j(x_n|\delta, m_j, \beta I)}. \tag{15}$$

E-2: Compute Expectation of u_n Given $x_n|u_n, z_{nk} = 1 \sim \mathcal{N}(m_k, \frac{\beta I}{u_n})$ and $u_n|z_{nk} = 1 \sim \text{Gamma}(\frac{\delta}{2}, \frac{\delta}{2})$, the posterior distribution $u_n|x_n, z_{nk} = 1$ follows $u_n|x_n, z_{nk} = 1 \sim \text{Gamma}\left(\frac{\delta+p}{2}, \frac{\delta+\frac{1}{\beta}(x_n-m_k)^\top(x_n-m_k)}{2}\right)$. Thus, the expectation of u_n is:

$$\mathbb{E}[u_n|x_n, z_{nk} = 1] = \frac{\delta + p}{\delta + \frac{1}{\beta}(x_n - m_k)^\top(x_n - m_k)} = w_{nk}. \tag{16}$$

E-3: Compute Expectation of $\ln u_n$ Using the property of the Gamma distribution, for $R \sim \text{Gamma}(a, b)$, the expectation $\mathbb{E}[\ln R] = \psi(a) - \ln b$ [31], where $\psi(a)$ is the Digamma function. Thus:

$$\mathbb{E}[\ln u_n|x_n, z_{nk} = 1] = \psi\left(\frac{\delta + p}{2}\right) - \ln\left(\frac{\delta + \frac{1}{\beta}(x_n - m_k)^\top(x_n - m_k)}{2}\right). \tag{17}$$

M-step: Parameter Updates

M-1: Update $m_k^\star$ and $\beta^\star$ Take partial derivatives of $Q_2(m^\star, \beta^\star|\Theta)$ and set them to zero:

$$\frac{\partial Q_2}{\partial m_k^\star} = 0 \implies m_k^\star = \frac{\sum_{n=1}^{N} \gamma_{nk} w_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk} w_{nk}}, \tag{18}$$

$$\frac{\partial Q_2}{\partial \beta^\star} = 0 \implies \beta^\star = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N} \gamma_{nk} w_{nk} (x_n - m_k)^\top (x_n - m_k)}{p \sum_{k=1}^{K}\sum_{n=1}^{N} \gamma_{nk}}. \tag{19}$$

M-2: Update $\delta^\star$ Solve the equation derived from $Q_1(\delta^\star|\Theta)$ using numerical methods:

$$-\psi\left(\frac{\delta^\star}{2}\right) + \ln\left(\frac{\delta^\star}{2}\right) + \kappa = 0. \tag{20}$$

Where κ is a constant computed from the E-step results γ_{nk} and w_{nk}:

$$\kappa = 1 + \frac{1}{K}\sum_{k=1}^{K} \frac{1}{\sum_{n=1}^{N} \gamma_{nk}} \sum_{n=1}^{N} \gamma_{nk}(\ln w_{nk} - w_{nk}) + \psi\left(\frac{\delta + p}{2}\right) - \ln\left(\frac{\delta + p}{2}\right). \tag{21}$$

Using the Digamma function approximation $\psi(s) \approx \ln s - \frac{1}{2s} - \sum_{i=2}^{\infty} \frac{B_i}{is^i}$ [33] (where B_i are Bernoulli numbers), the solution can be further refined.

Iterate E-step and M-step until the Q-function converges or the maximum iterations are reached. The TK-means algorithm are detailed in Algorithm 3.

Algorithm 3 TK-means

Input: User's data record $\hat{r}$, num of cluster K
Output: centroids of cluster μ, cluster label idx, iterations iter

initialization variable:$\mu \rightarrow kclusteringcentres \quad \nu \rightarrow FreedomParameters$
while $iter < maxiter$ **do**
  Calculate the posterior probability $\gamma_{nk} = \mathbb{E}[z_{nk}|x_n] = \frac{w_k f_k(x_n|\delta, m_k, \beta I)}{\sum_{j=1}^{K} w_j f_j(x_n|\delta, m_j, \beta I)}$;
  Calculate the expectation $\mathbb{E}[u_n|x_n, z_{nk}=1] = \frac{\delta+p}{\delta+\frac{1}{\beta}(x_n-m_k)^\top(x_n-m_k)} = w_{nk}$;
  Calculate $\mathbb{E}[\ln u_n|x_n, z_{nk}=1] = \psi\left(\frac{\delta+p}{2}\right) - \ln\left(\frac{\delta+\frac{1}{\beta}(x_n-m_k)^\top(x_n-m_k)}{2}\right)$;
  Updating the Q function value;
  if $\Delta Q < \delta$ **then**
    Break;
  end if
  Update the $m_k^\star = \frac{\sum_{n=1}^{N} \gamma_{nk} w_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk} w_{nk}}$;
  Update the $\beta^\star = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N} \gamma_{nk} w_{nk} (x_n-m_k)^\top (x_n-m_k)}{p \sum_{k=1}^{K}\sum_{n=1}^{N} \gamma_{nk}}$;
  Update the $\delta^\star$ by solving the $-\psi\left(\frac{\delta^\star}{2}\right) + \ln\left(\frac{\delta^\star}{2}\right) + \kappa = 0$;
end while
return μ,idx,iter;

5 Privacy and Utility Analysis

Theorem 1. *BPGM satisfies d_χ-privacy.*

Proof. Recall that d_χ-privacy requires $\frac{f_r^\ell(\hat{r})}{f_{r'}^\ell(\hat{r})} \leq e^{\epsilon d_E(r,r')}$, where $r, r' \in D$, $d_E(r,r') = \|r - r'\|_2 > 0$ is the Euclidean distance between real user data, $\hat{r} \in \mathbb{R}^d$ is the perturbed record, and $\epsilon > 0$ ensures $e^{\epsilon d_E(r,r')} > 1$. Then, We have:

$$\frac{f_r^\ell(\hat{r})}{f_{r'}^\ell(\hat{r})} = e^{-\epsilon \cdot \left(d(r',\hat{r}) - d(r,\hat{r})\right)} \leq e^{\epsilon \cdot d(r,r')} \tag{22}$$

So, BPGM mechanism satisfies d_χ-privacy.

Theorem 2. *The synthetic data $\hat{r}$ generated by BPGM is an unbiased estimate of the original data r.*

Proof. For $i = 1, \ldots, d$, the expected value of $\hat{r}_i$ is $E(\hat{r}_i) = \int \cdots \int \hat{r}_i F^{BPGM}(\hat{r}) d\hat{r}$, and the bias is $\Delta_i = E(\hat{r}_i) - r_i = \int \cdots \int_{\hat{r} \in R} (\hat{r}_i - r_i) \frac{\epsilon}{(1-e^{-\epsilon\ell})} e^{-\epsilon \|\hat{r}-r\|_2} d\hat{r}$, where $\|\hat{r} - r\|_2 \leq \ell$. Our goal is to demonstrate $\Delta_i = 0$.

For $i \leq d-2$, using spherical coordinates [34] where $\hat{r}_i - r_i = r \sin\theta_1 \cdots \sin\theta_{i-1} \cos\theta_i$:

$$0 \leq \Delta i = \frac{\epsilon}{(1-e^{-\epsilon\ell})} \left(\int_0^{\ell} \hat{d}^d e^{-\epsilon\hat{d}} d\hat{d} \right) \cdot \prod_{j=1}^{t-1} \left(\int_0^{\pi} sin^{d-j}\theta_j \right) \cdot \int_0^{\pi} sin^{d-1-i}\theta_i cos\theta_i d\theta_i \cdot \prod_{j=i+1}^{d-2} \left(\int_0^{\pi} sin^{d-1-j}\theta_j d\theta_j \right) \int_0^{2\pi} d\theta_{d-1}. \tag{23}$$

Since $\int_0^{\pi} \sin^{d-1-i}\theta_i \cos\theta_i, d\theta_i = \left[\frac{1}{d-i} \sin^{d-i}\theta_i \right]_0^{\pi} = 0$, $\Delta_i = 0$.

For $i = d-1$ or d:

$$\Delta_i = \frac{\epsilon}{(1-e^{-\epsilon\ell})} \left(\int_0^{\ell} \hat{d}^d e^{-\epsilon\hat{d}} d\hat{d} \right) \prod_{j=1}^{d-2} \left(\int_0^{\pi} \sin^{d-j}\theta_j d\theta_j \right) \int_0^{2\pi} g(\theta_{d-1}) d\theta_{d-1}. \tag{24}$$

Where $g(\theta_{d-1}) = \cos\theta_{d-1}$ (for $i = d-1$) or $g(\theta_{d-1}) = \sin\theta_{d-1}$ (for $i = d$), and $\int_0^{2\pi} g(\theta_{d-1}), d\theta_{d-1} = 0$, so $\Delta_i = 0$.

Thus, $\Delta_i = 0$ for all i, proving $\hat{r}$ is an unbiased estimate of r. This ensures statistical consistency, maximizing data utility and privacy, making BPGM suitable for diverse data analysis tasks.

Theorem 3. *Robustness Analysis of TK-means.*

Proof. The log-likelihood of TK-means is $\ln L(\Theta \mid x) = \ln \prod_{n=1}^{N} \prod_{k=1}^{K} [f_k(x_n | \delta, m_k, S)]^{z_{nk}}$. Focusing on x-related terms, the loss simplifies to $loss_{TK-means} \propto \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \ln \left(1 + \frac{1}{\delta\beta} (x_n - m_k)^{\top} (x_n - m_k) \right)$, a $\ln L_2$ function. Unlike the L_2 loss of traditional K-means, this form reduces sensitivity to outliers.

In the M-1 step, the centroid update $m_k^{\star} = \frac{\sum_{n=1}^{N} \gamma_{nk} w_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk} w_{nk}}$ leverages all sample information, mitigating the impact of random initial centroids and enhancing stability compared to traditional K-means.

6 Experiments

We compared the proposed BPGT method against the BPM method [24] by Yang et al. and the nD-Laplace method [25] by Alishahi et al, using a Windows 10 system with an AMD Ryzen 7 5800H 3.2GHz CPU. Privacy budgets were set at $\epsilon = 0.1, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$. Standard K-means without privacy protection served as the baseline. Experiments were conducted on four real-world datasets, with each repeated 20 times and results averaged.

6.1 Experimental Setup

The BPGT framework for privacy-preserving K-means clustering was evaluated using two settings: (1) Datasets: Four real-world datasets with varied distributions and privacy challenges; (2) Metrics: Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Relative Error (RE) to measure clustering utility and distortion across privacy budgets.

Datasets

(1) **Iris**: Classic dataset with 150 instances, 4 features each.
(2) **Seeds**: Geometric properties of wheat kernels, 210 instances, 7 features each.
(3) **Wine**: Chemical analysis of wines, 178 instances, 13 features each.
(4) **Wdbc**: Diagnostic Wisconsin Breast Cancer Database, 569 instances, 30 features each.

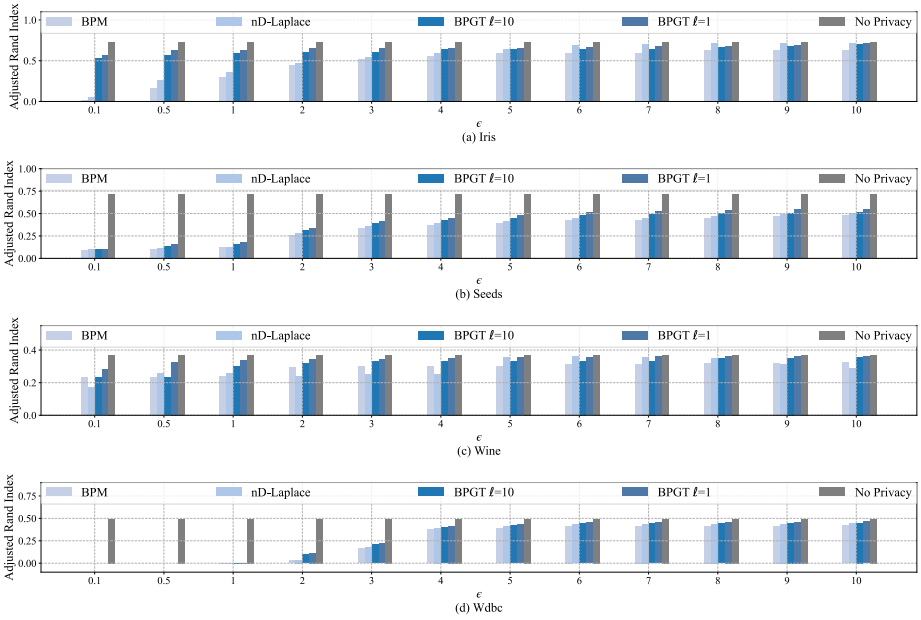

Fig. 3. Comparison of the performance of $BPGT\ \ell = \star$ with the traditional method on different datasets by histograms, with ARI values as y-axis and ϵ as x-axis. Where (a) Iris, (b) Seeds, (c) Wine, (d) Wdbc.

Metrics

(1) **ARI** (Adjusted Rand Index): Measures agreement between clustering results and true labels, ranging from -1 (worse than random) to 1 (perfect consistency), with 0 indicating random assignment.
(2) **NMI** (Normalized Mutual Information): Quantifies shared information between clustering and true labels, ranging from 0 (independent) to 1 (identical).
(3) **RE** (Relative Error): Assesses clustering accuracy via the relative difference between perturbed and original cluster centers.

6.2 Experimental Results

Performance in terms of ARI: To assess how well predicted labels match true labels, we used the Adjusted Rand Index (ARI) to compare our proposed method with traditional approaches. Fig 3 shows the results, with Fig 3a displaying ARI values for the Iris dataset. ARI increases with ϵ: at $\epsilon = 0.1$, all methods perform poorly, with BPM and nD-Laplace being the worst; as ϵ rises, $BPGT\ \ell = 1$ achieves an ARI of 0.66, and at $\epsilon = 9, 10$, both $BPGT\ \ell = 1$ and $BPGT\ \ell = 10$ nearly match K-means without privacy protection. These patterns hold in Figs 3b, 3c, and 3d.

Performance in terms of NMI: We also used Normalized Mutual Information (NMI) for further comparison, with results mirroring ARI trends, as shown in Fig 4. Analysis reveals that $BPGT\ \ell = 10$ adds significant noise at low ℓ, widening the gap between generated and original data, while $BPGT\ \ell = 1$ limits its noise to 0–1, yielding better clustering but restricting the generated data range. Still, attackers cannot easily extract specific original values or analyze user behavior.

Performance in terms of RE: Next, we evaluated clustering center errors across privacy-preserving algorithms. Fig 5 shows Relative Errors(RE) for the Iris dataset, with Fig 5a highlighting $BPGT\ \ell = 1$ as the best performer: error drops from 0.54 at $\epsilon = 0.1$ to 0.18, 0.16 and 0.16 at $\epsilon = 5, 9, 10$. $BPGT\ \ell = 10$ also performs well, while BPM and nD-Laplace lag, with errors of 0.62 and 0.20 at $\epsilon = 10$, 3.88 and 1.25 times higher, respectively. These trends persist in Figs 5b, 5c, and 5d.

6.3 Discussion

Our analysis indicates that a low privacy budget increases perturbation noise, causing significant data distortion. However, as shown in Figs 6 and 7, BPGT ($\ell = 1$ and $\ell = 10$) mitigates this by projecting data into a one-dimensional space at the user end and injecting noise based on distance attributes. In contrast, nD-Laplace's mesh redrawing alters the data distribution, and BPM's two-step sampling introduces bias. BPGT effectively addresses both issues. Additionally, the T-Mixture Model in TK-means enhances robustness to outliers and varying

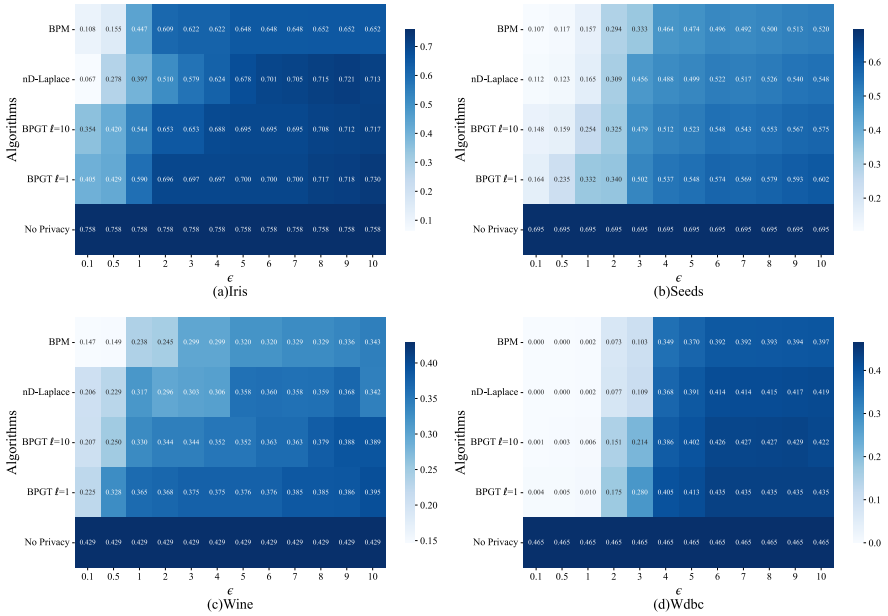

Fig. 4. Comparison of the performance of $BPGT\ \ell = \star$ with the traditional method on different datasets by heat map, with different algorithms as y-axis and ϵ as x-axis. Where (a) Iris, (b) Seeds, (c) Wine, (d) Wdbc.

distributions, maintaining clustering accuracy under strict privacy constraints, making BPGT highly effective in such scenarios.

7 Conclusion

In this paper, we propose BPGT, a d_χ-privacy-based K-means framework that enhances clustering by balancing privacy protection and data utility. Unlike traditional methods, BPGT minimizes noise impact by perturbing distances between user-side data records rather than individual attributes, reducing clustering distortion. It employs TK-means on the server side to handle outliers and heavy-tailed distributions, boosting robustness. We provide rigorous proofs of privacy and utility for the BPGM mechanism. Experiments on real-world datasets show BPGT outperforms conventional methods across multiple metrics, confirming its practical superiority. Future research could explore optimizing the privacy budget ϵ and noise sampling upper bound ℓ, developing adaptive noise injection strategies for diverse data distributions, and improving computational efficiency through distributed computing and dimensionality reduction techniques. These advancements would enhance its applicability and performance in privacy-preserving clustering.

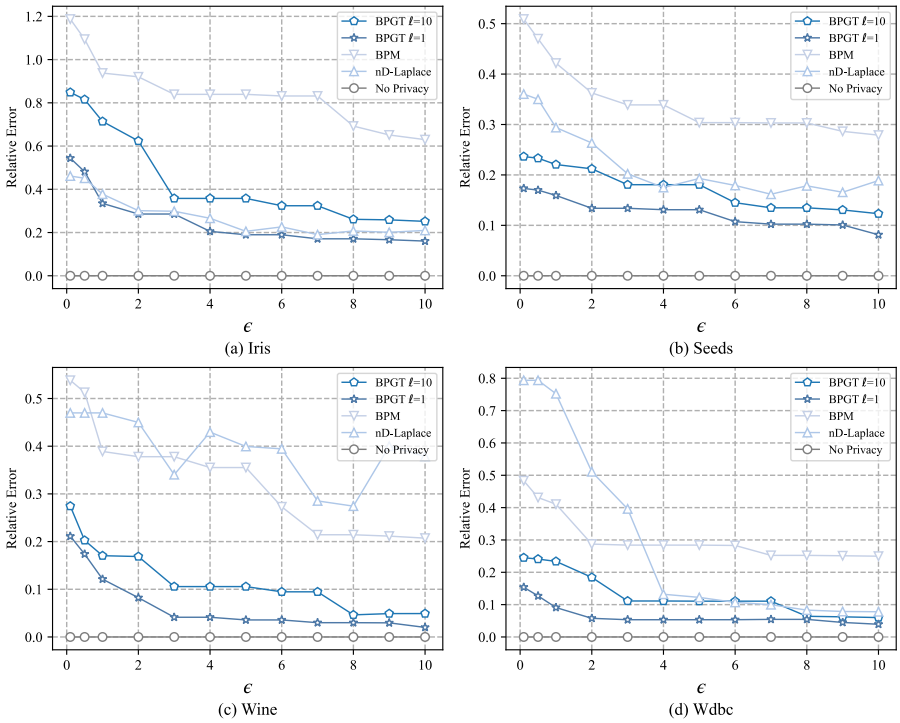

Fig. 5. Comparison of the performance of *BPGT* $\ell = \star$ with the traditional method on different datasets by line graph, with RE as y-axis and ϵ as x-axis. Where (a) Iris, (b) Seeds, (c) Wine, (d) Wdbc.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant number 62272162 and 62302062, the Hunan Provincial Natural Science Foundation of China under Grant number 2025JJ50398 and 2023JJ40081, and the project of Hunan Provincial Social Science Achievement Review Committee of China under Grant number XSP25YBZ104.

References

1. Oyewole, G. J., Thopil, G. A.: Data clustering: Application and trends. *Artificial Intelligence Review* **56** (7), 6439–6475 (2023)
2. Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., Akinyelu, A. A.: A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence* **110**, 104743 (2022)

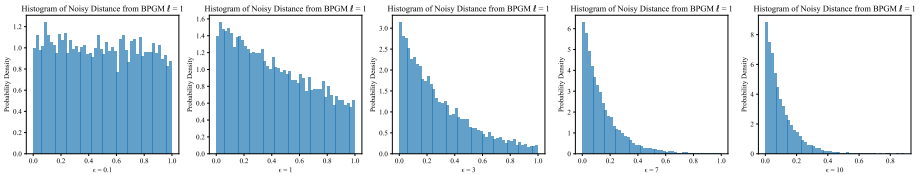

Fig. 6. Noise distance probability distributions for different ϵ on $\ell = 1$ for the BPGM mechanism.

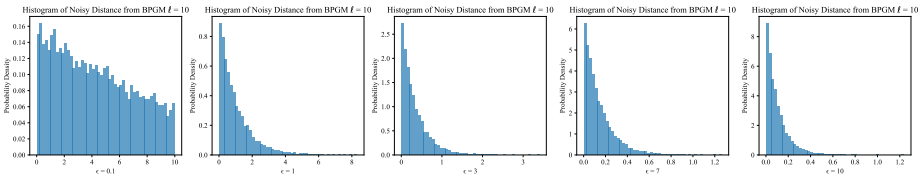

Fig. 7. Noise distance probability distributions for different ϵ on $\ell = 10$ for the BPGM mechanism.

3. Guo, G., Altrjman, C.: E-Commerce customer segmentation method under improved K-Means algorithm. In: International Conference on Multi-modal Information Analytics, pp. 1083–1089. Springer, Cham (2022)
4. Zubair, M., Iqbal, M. D. A., Shil, A., Haque, E., Hoque, M. M., Sarker, I. H.: An efficient k-means clustering algorithm for analysing COVID-19. In: Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020, pp. 422–432. Springer, Cham (2021)
5. Deng, J., Guo, J., Wang, Y.: A novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering. *Knowledge-Based Systems* **175**, 96–106 (2019)
6. Zhang, S., Li, O., Tan, Z., Peng, T., Wang, G.: A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. *Future Generation Computer Systems* **94**, 40–50 (2019)
7. Zhang, S., Guo, T., Liu, Q., Luo, E., Choo, K.-K. R., Wang, G.: ALPS: Achieving accuracy-aware location privacy service via assisted regions. *Future Generation Computer Systems* **145**, 189–199 (2023)
8. Yang, M., Guo, T., Zhu, T., Tjuawinata, I., Zhao, J., Lam, K.-Y.: Local differential privacy and its applications: A comprehensive survey. *Computer Standards & Interfaces* **89**, 103827 (2024)
9. Wang, T., Zhang, X., Feng, J., Yang, X.: A comprehensive survey on local differential privacy toward data statistics and analysis. *Sensors* **20** (24), 7030 (2020)
10. Su, D., Cao, J., Li, N., Bertino, E., Jin, H.: Differentially private k-means clustering. In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, pp. 26–37. ACM, New York (2016)
11. Xia, C., Hua, J., Tong, W., Zhong, S.: Distributed K-Means clustering guaranteeing local differential privacy. *Computers & Security* **90**, 101699 (2020)
12. Nissim, K., Stemmer, U.: Clustering algorithms for the centralized and local models. In: Algorithmic Learning Theory, pp. 619–653. PMLR (2018)

13. Chatzikokolakis, K., Andrés, M. E., Bordenabe, N. E., Palamidessi, C.: Broadening the scope of differential privacy using metrics. In: Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013, Proceedings, pp. 82–102. Springer, Heidelberg (2013)
14. Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: Differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 901–914. ACM, New York (2013)
15. Alvim, M., Chatzikokolakis, K., Palamidessi, C., Pazii, A.: Local differential privacy on metric spaces: Optimizing the trade-off with utility. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pp. 262–267. IEEE, Piscataway, NJ (2018)
16. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067. ACM, New York (2014)
17. Su, D., Cao, J., Li, N., Bertino, E., Lyu, M., Jin, H.: Differentially private k-means clustering and a hybrid approach to private optimization. *ACM Transactions on Privacy and Security (TOPS)* **20** (4), 1–33 (2017)
18. Stemmer, U., Kaplan, H.: Differentially private k-means with constant multiplicative error. In: Advances in Neural Information Processing Systems, vol. **31** (2018)
19. Stemmer, U.: Locally private k-means clustering. *Journal of Machine Learning Research* **22** (176), 1–30 (2021)
20. Feng, X., Zhang, C.: MPLDP: Multi-level personalized local differential privacy method. *IEEE Access* (2024)
21. Song, H., Shen, H., Zhao, N., He, Z., Wu, M., Xiong, W., Zhang, M.: APLDP: Adaptive personalized local differential privacy data collection in mobile crowdsensing. *Computers & Security* **136**, 103517 (2024)
22. Fu, N., Ni, W., Hu, H., Zhang, S.: Multidimensional grid-based clustering with local differential privacy. *Information Sciences* **623**, 402–420 (2023)
23. Liebenow, J., Schütt, Y., Braun, T., Gehrke, M., Thaeter, F., Mohammadi, E.: DPM: Clustering sensitive data through separation. In: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, pp. 273–287. ACM, New York (2024)
24. Yang, M., Tjuawinata, I., Lam, K.-Y.: K-Means clustering with local d_χ-privacy for privacy-preserving data analysis. *IEEE Transactions on Information Forensics and Security* **17**, 2524–2537 (2022)
25. Alishahi, M., Van Der Ende, T., Maathuis, C.: Generalized differential privacy for clustering. *Authorea Preprints* (2024)
26. Athanasiou, A., Chatzikokolakis, K., Palamidessi, C.: Enhancing metric privacy with a shuffler. In: PETS 2025-25th Privacy Enhancing Technologies Symposium (2025)
27. Zhou, Y., Wang, Y., Teng, L., Huang, Q., Chen, C.: Approximate kernel density estimation under metric-based local differential privacy. In: The 40th Conference on Uncertainty in Artificial Intelligence (2024)
28. Liang, Y., Yi, K.: Concentrated geo-privacy. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, pp. 1934–1948. ACM, New York (2023)
29. McLachlan, G. J., Basford, K. E.: *Mixture models: Inference and applications to clustering*, vol. **38**. Marcel Dekker, New York (1988)

30. Peel, D., McLachlan, G. J.: Robust mixture modelling using the t distribution. *Statistics and Computing* **10**, 339–348 (2000)
31. Liu, C., Rubin, D. B.: ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, 19–39 (1995)
32. Langley, P.: *Elements of machine learning*. Morgan Kaufmann, San Francisco, CA (1996)
33. Abramowitz, M., Stegun, I. A. (eds.): *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. **55**. US Government Printing Office, Washington, DC (1948)
34. Muller, M. E.: A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM* **2** (4), 19–20 (1959)
35. Feyisetan, O., Diethe, T., Drake, T.: Leveraging hierarchical representations for preserving privacy and utility in text. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 210–219. IEEE, Piscataway, NJ (2019)
36. Fernandes, N., McIver, A., Palamidessi, C., Ding, M.: Universal optimality and robust utility bounds for metric differential privacy. *Journal of Computer Security* **31** (5), 539–580 (2023)