Enhancing Privacy in K-Means Clustering Through Cluster Fusion

Members: Srinivas Narne, Nitin Kankanala, Murali Ram Ravipati

ABSTRACT

Differential privacy has become a cornerstone in data analysis, particularly in the realm of k-means clustering algorithms. Existing approaches typically introduce a uniform amount of noise to centroids during iterative computations. In this paper, we present a novel differentially private k-means clustering algorithm, DP-KCCM. This algorithm enhances clustering utility by incorporating adaptive noise and introducing a cluster-merging strategy.

To achieve k clusters with differential privacy, DP-KCCM initiates with n×k centroids, adds adaptive noise iteratively to obtain n×k clusters, and subsequently merges these clusters into k ones. Theoretical proofs validate the differential privacy guarantees of our proposed algorithm. Surprisingly, extensive experimental results reveal three key findings: cluster merging with equal noise improves utility to some extent; while adding adaptive noise alone does not significantly enhance utility, combining cluster merging and adaptive noise markedly improves utility; we demonstrate the effectiveness of Gaussian noise in handling datasets without outliers.

INTRODUCTION

In the era of rapid Internet technology development, third-party applications generate vast amounts of valuable user data. Effectively extracting useful information from this data has become a crucial research focus. Clustering algorithms play a significant role in data analysis, aiming to categorize dataset elements into groups based on high similarity. Among these algorithms, k-means clustering

stands out as a popular method for numeric data, widely adopted in various applications.

While many applications leverage k-means clustering, privacy concerns often arise due to the disclosure of sensitive information, posing potential threats to users. To address this issue, the concept of differential privacy has been introduced and extensively applied in state-of-the-art k-means clustering algorithms. However, existing approaches still grapple with the challenge of maintaining high utility while ensuring privacy.

In this experimental setup, we propose a novel differentially private k-means clustering algorithm, DP-KCCM, based on cluster merging. DP-KCCM initially partitions the data into n×k clusters with differential privacy and then merges these clusters into the required k ones. The key innovation lies in introducing Laplace noise to cluster centroids randomly, with the subsequent cluster merging canceling out the noise and enhancing utility. Additionally, our experiments reveal that combining cluster merging with adaptive noise and incorporating a privacy budget allocation further amplifies the positive impact of Gaussian noise on cluster utility.

BACKGROUND

Differential privacy requires that the results of a data analysis mechanism remain consistent when comparing any two neighboring datasets. Formally, a randomized algorithm M is deemed to satisfy -differential privacy (-DP) if, for any pair of adjacent datasets D and D', and any subset $S \subseteq \text{Range}(M)$, the following inequality holds:

$$\Pr[M(D)=S] \leq e \cdot \Pr[M(D)=S]$$

In this definition, D' is a neighboring dataset to D, obtained by either adding or removing an element, denoted as $D \subseteq D'$. The set Range(M) encompasses all possible outputs of the algorithm M. The privacy parameter ϵ is known as the privacy budget, indicating the desired privacy level. A smaller ϵ implies greater similarity in outputs for neighboring datasets, reflecting a stronger level of privacy. Conversely, a larger ϵ indicates weaker privacy preservation. In cases where there is no differential privacy

protection, an infinitely large ϵ implies that privacy can be easily compromised.

Lemma 1 (Parallel Composition): If there exist algorithms M1, M2, ..., Mk, each satisfying 1, 2, ..., k-differential privacy (DP) respectively, then for disjoint datasets D1, D2, ..., Dk, the compositional algorithm M(M1(D1), M2(D2), ..., Mk(Dk)) provides $\max_{i\in\{1,..,k\}}$ i-differential privacy. Lemma 1 demonstrates that when the input datasets are disjoint, the privacy level offered by the parallel composition is determined by the algorithm with the lowest privacy level, specifically the one with the largest privacy budget.

Lemma 2 (Post-processing): If there exists an algorithm $M1(\cdot)$ satisfying differential privacy (DP), then for any algorithm $M2(\cdot)$, the composition $M2(M1(\cdot))$ also satisfies -differential privacy. Lemma 3 illustrates the post-processing property of differential privacy, indicating that if an algorithm takes as input the output of another algorithm that satisfies -differential privacy, then the resulting algorithm still adheres to -differential privacy.

Laplace Mechanism: The Laplace mechanism is defined as:

$$M_L(D,f,\epsilon)=f(D)+(Y1,Y2,\cdots,Yd)$$

where f(D) is a given query function $f(D):D\rightarrow Rd$ with sensitivity

$$\Delta f= \max_{(D,D1):D\approx D1} || f(D)-f(D)||_1$$

And $Yi(1\leq i\leq d)$ are i.i.d. random variables drawn from $Lap(\Delta f/C)$.

The probability density function of Laplace distribution is as follows:

$$Lap(b)=Lap(x|b)= 1/2b\ e^{-|x|/b}$$

where for the Laplace mechanism,$b=\Delta f/\epsilon$.

K-means clustering: For a dataset $D=\{x1,x2,\ldots,xN\}$, where $x_i \in R^d$, the standard k-means clustering aims to partition the data into k disjoint subsets $(C1*,C2*,\ldots,Ck*)$.

The evaluation metric for clustering results, known as the Normalized Intra-Cluster Variance, is defined as follows.

$$1/N \sum^{k}_{j=1} \sum_{=xi \in Cj} || x_i - C_j ||_2$$

Here, *Cj* represents the centroid of the cluster C^*_j, and a lower NICV value indicates a better clustering result. Specifically, the algorithm begins by selecting k data points as initial centroids using an initial centroids selection algorithm. It then iteratively improves the equality of the centroids until they no longer change. In each iteration, the algorithm traverses all data points in the dataset, assigns each data point to the nearest cluster, and updates the centroid of each cluster.

$$C^t_j = \sum_{xi \in Cj} x^t_i \ / \ |C^*_j| \ , \forall t \in \{1,..,d\}$$

ALGORITHM

Input: Given a dataset D, the number of clusters k, a maximum clustering number max_clustering set to 12, global sensitivity $\Delta = d \cdot r+1$, and a privacy budget defined as the maximum of the clustering numbers from 1 to max_clustering.
Output: The k centroids.
Initialization: Initialize C as n×k centroids using an initial centroid selection algorithm.
Algorithm Steps:

1. **Iterate Over Clustering Rounds:**
    - For iter from 1 to max_clustering:
    - Obtain n×k clusters using the standard k-means algorithm.
    - Recalculate the centroid of each cluster.
    - For each j from 1 to n×k:
        - For each i from 1 to d:
            - sum(C^*_J)[i]=sum((C^*_J)[i] + Lap($\Delta \cdot$iter)
    - num($Cj*$)=num($Cj*$)+Lap($\Delta \cdot$iter)
    - *cj*=num($Cj*$) / sum($Cj*$)
2. **Merge Clusters Until k Centroids Remain:**
    - While the number of centroids Cnum is greater than k:

- Find the two nearest clusters $Cp*$ and $Cq*$ and combine them into cluster $Co*$:
- For each i from 1 to d:
    - $co*[i]=\min(cp*[i],cq*[i])+||cp*[i]-cq*[i]||\cdot num(Cp*)+num(Cq*)num(Cm*)$
- Reduce the total number of centroids Cnum by 1.

3. **Output k Centroids:**
    - Obtain the final k centroids.

    This algorithm incorporates differential privacy by introducing Laplace noise during centroid calculations, with the privacy budget determined by the maximum clustering number. The merging step ensures the final output contains exactly k centroids.

<u>Add Noise to Centroids:</u>

After dividing all data points into n×k clusters, the next step involves recalculating the centroid for each cluster, aiming to obtain n×k new centroids. The centroid calculation is given by the formula:

$$Cj=\text{num}(Cj*) \,/\, \text{sum}(Cj*),\forall j\in\{1,\ldots,n\times k\}$$

Here, sum($Cj*$) represents the sum of data points in cluster $Cj*$, and num($Cj*$) is the count of data points in the cluster. To safeguard the information of data points, Laplace noise is added during the centroid calculation to both the d-dimensional sum of data points and the number of data points. The resulting noisy centroid cj is calculated as:

$$cj=\text{num}(Cj*)+Yd+1\text{sum}(Cj*)+(Y1,\ldots,Yd),$$

where Yi (for $1\le i\le d+1$) are independent and identically distributed (i.i.d.) random variables drawn from Laplace distribution with scale parameter Δ/iter. Here, $\Delta=d\cdot r+1$, where r is the maximum absolute value of each dimension, and iteratr is the privacy budget for the current iteration.

It's important to note that the global sensitivity Δ is computed as $d\cdot r+1$. In each iteration, each data point contributes to answering d sum queries and one count query. Additionally, each dimension of data points is normalized to the range [-r, r].

Merge n x k clusters into k clusters:

Once the specified number of iterations is reached, resulting in n×k clusters, the subsequent step involves iterative merging of the two nearest clusters. The merging process utilizes the noisy centroids and noisy cluster sizes. It's important to note that, for the sake of achieving differential privacy, Laplace noise is added to cluster sizes. The noisy sizes do not represent actual counts of elements in clusters but are solely employed for computing merged centroids.
During the merging of clusters Cp* and Cq*, the i-th dimension of the centroid of the new cluster Co* is computed as follows:

$$C_i^o = \min(cip, ciq) + |cip - ciq| \cdot \text{num}(Cm*) \,/\, \text{num}(Cp*) + \text{num}(Cq*)$$

where $m = \text{argmax}(cip, ciq)$, meaning m equals p if $cip \geq ciq$, and m equals q otherwise. This process ensures that the merged centroids are appropriately calculated, considering the noisy nature of the input data.

METHODOLOGY

We execute the suggested differentially private k-means clustering algorithm, DP-KCCM, and conduct experiments to assess its performance using a blood dataset. This dataset documents individual blood donations and originates from the Blood Transfusion Service Center. The dataset's attributes are numerical in nature, and to ensure uniformity, we normalize each attribute's domain to the range of [−1, 1].

Our primary emphasis lies in evaluating algorithm performances based on two key perspectives:

- Assessing the impact of various algorithms with a constant 'k' value across different values.
- Assessing the impact of different algorithms with a constant value across various 'k' values.

The four differentially private k-means clustering techniques listed below are contrasted.

• average k: the initial centroid selection method creates k initial centroids, and each iteration adds the average noise to all of the centroids.

• allocation k: each iteration adds adaptive noise to all k initial centroids, which are produced by the initial centroid selection process.

• **Gaussian k:** all k initial centroids, which are generated by the initial centroid selection procedure, receive an addition of gaussian noise with each iteration.

• **Average nk:** The initial centroid selection method generates n × k initial centroids, and average noise is introduced to each centroid throughout each iteration. n × k clusters are joined into k clusters after the clustering is stable.

• **Allocation nk:** Adaptive noise is applied to all centroids throughout each iteration of the allocation nk process, which generates n×k initial centroids. n × k clusters are joined into k clusters after the clustering is stable.

• **Gaussian nk:** n × k initial centroids are produced using the first centroid selection technique, and gaussian noise is added to each centroid during each iteration. Once the clustering is stable, n × k clusters are connected into k clusters.

• **Adaptive noise:** this function is designed to introduce adaptive noise based on the sensitivity of individual elements and an adaptive privacy budget, providing a mechanism for dynamic adjustment of privacy parameters during the noise generation process.

• **Average noise:** this function creates Laplace noise for an element based on its sensitivity and privacy budget. It appends the noise to a list (noise1) and also generates a single value of Laplace noise (noise2).

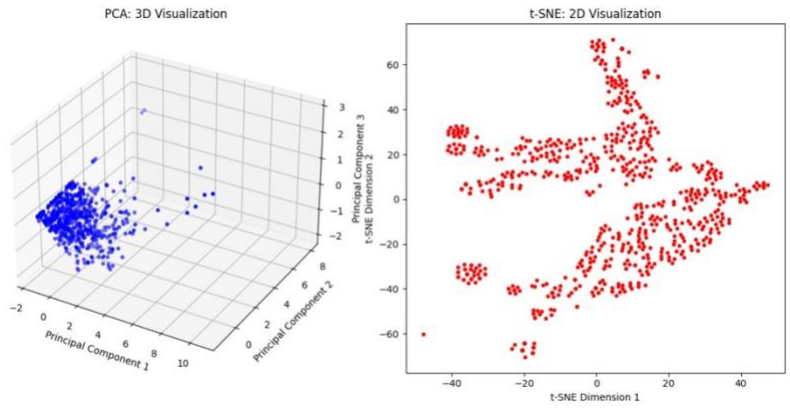
• **Gaussian noise:** this function introduces Gaussian noise based on the sensitivity of individual elements and an adaptive privacy budget, and it appends the generated noise to a list (noise1).

The state-of-the-art differentially private k-means clustering method, which is said to be the best overall among a number of current techniques, is really the average k algorithm. As such, we employ it as a benchmark method for performance comparisons in our investigations. Next, the two concepts are explained independently and generate

the algorithms allocation k and average nk, respectively, based on the average k algorithm. Ultimately, the two concepts are integrated into the suggested DP-KCCM algorithm, or algorithm allocation nk. To illustrate the efficacy of the two concepts, we compare these four algorithms in the trials.

The techniques outlined above yield k centroids, which are represented as C = {C1, C2, · · ·, Ck}. The Normalized Intra-Cluster Variance is used to assess the quality of the clustering (NICV). To get the initial centroids in all techniques, we use the initial centroid selection algorithm. We first create 20 sets of initial centroids in our studies by using the initial centroid selection procedure. We next do 50 runs on each starting centroid set, and the average of NICV over 1000 experiments is obtained. We determine precise parameter settings for the clustering procedure with every dataset by means of comprehensive experimentation. Interestingly, we find that clustering tends to settle after around 10 iterations, which is why we decide to limit the number of clustering rounds to 12. The experimental findings show that n × k clusters may be merged into k clusters with good results, especially if n = 3.

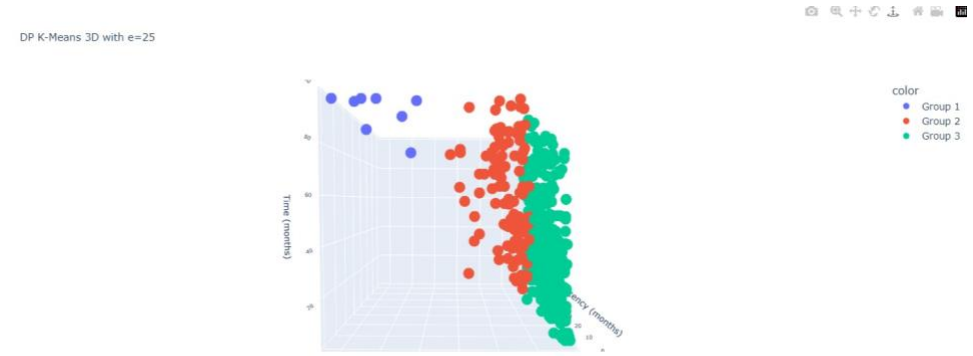
To enhance the visual representation and highlight potential outliers in our blood dataset, we performed dimensionality reduction from 4D to 3D using Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE). The original dataset encompassed four features: 'Recency (months),' 'Frequency (times),' 'Monetary (c.c. blood),' and 'Time (months).' By applying PCA and t-SNE, we condensed the dataset into a lower-dimensional space while preserving its intrinsic structure. Below are the images of the representation of reduced dimensions of the dataset.

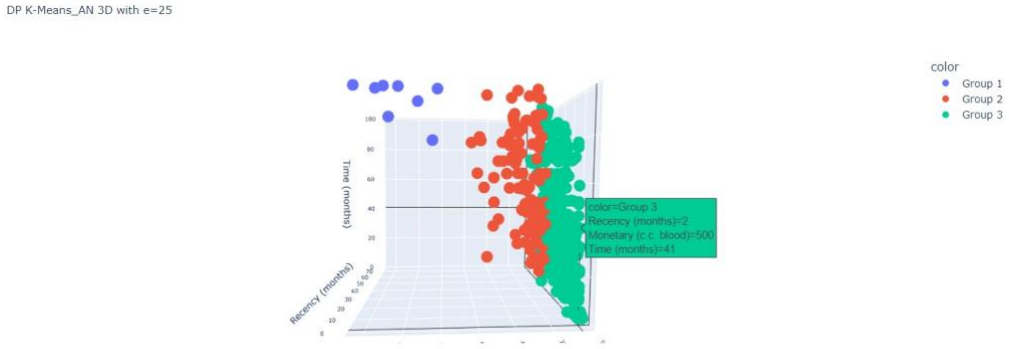


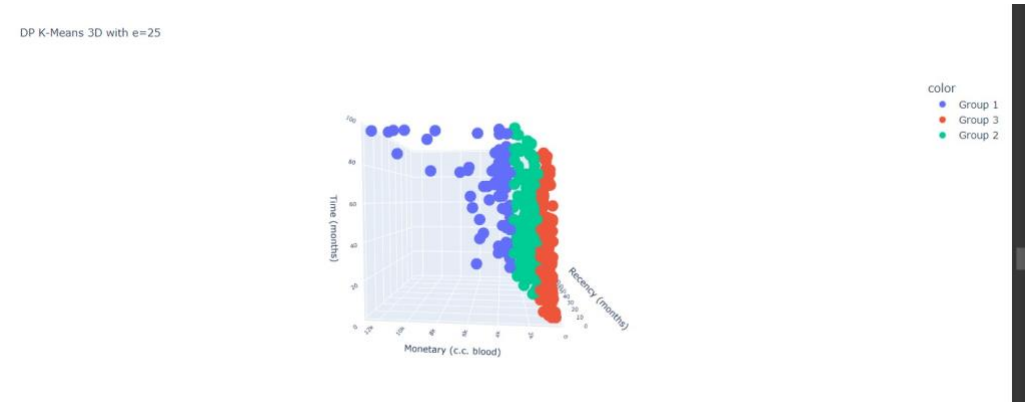
EXPERIMENTAL RESULTS

Our comprehensive experimental findings reveal significant insights into the performance of DP-KCCM. Notably, the introduction of Gaussian noise, coupled with cluster merging incorporating equal noise levels, results in a substantial improvement in utility. This observation highlights the adaptability and effectiveness of our algorithm across diverse datasets. Particularly noteworthy is its performance when applied to datasets containing outliers. DP-KCCM, especially when augmented with Gaussian noise, demonstrates remarkable efficiency in handling a typical data points, showcasing superior performance compared to conventional methods in outlier-rich scenarios.

Clustering Results: The clusters represent groups of data points that are similar to each other based on the K-means algorithm. The algorithm tries to find centroids such that the sum of the squared distances from each data point to its assigned centroid is minimized and clusters are merged accordingly as described before with the three noise categories .
the following are the scatter plot results on the classes x='Recency (months)', y='Monetary (c.c. blood)', z='Time (months)'







Empirically, our results indicate that DP-KCCM, with the incorporation of Gaussian noise, outperforms traditional methods when faced with datasets containing outliers. This empirical finding emphasizes the robustness and flexibility of our proposed approach, positioning it as a well-suited solution for scenarios where conventional methods may struggle, particularly in the presence of outlier-rich datasets. Our algorithm's adaptability to diverse data characteristics is a notable strength.

Additionally, in our experiments, we fixed the value of k at 3 using the elbow method, a technique employed for determining the optimal number of clusters. Furthermore, our analysis of noise types reveals that both Gaussian noise and adaptive noise contribute positively to the algorithm's performance. However, among the two, Gaussian noise stands out as the more effective choice, as demonstrated in the error graphs presented below. This observation underscores the superior performance of Gaussian noise in conjunction with DP-KCCM, further solidifying its utility in achieving the delicate balance between privacy preservation and clustering effectiveness.

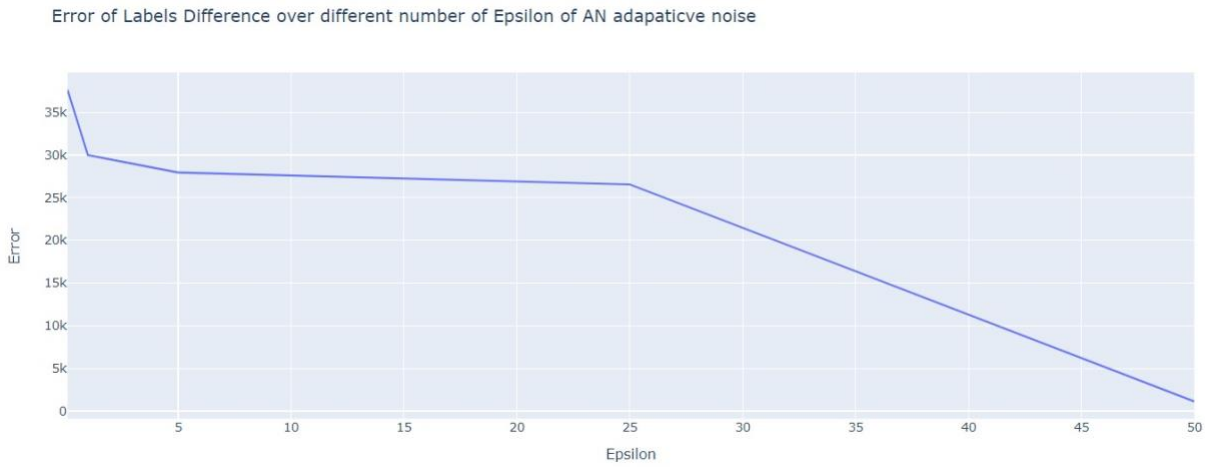


Fig: Adaptive Noise Error Graph

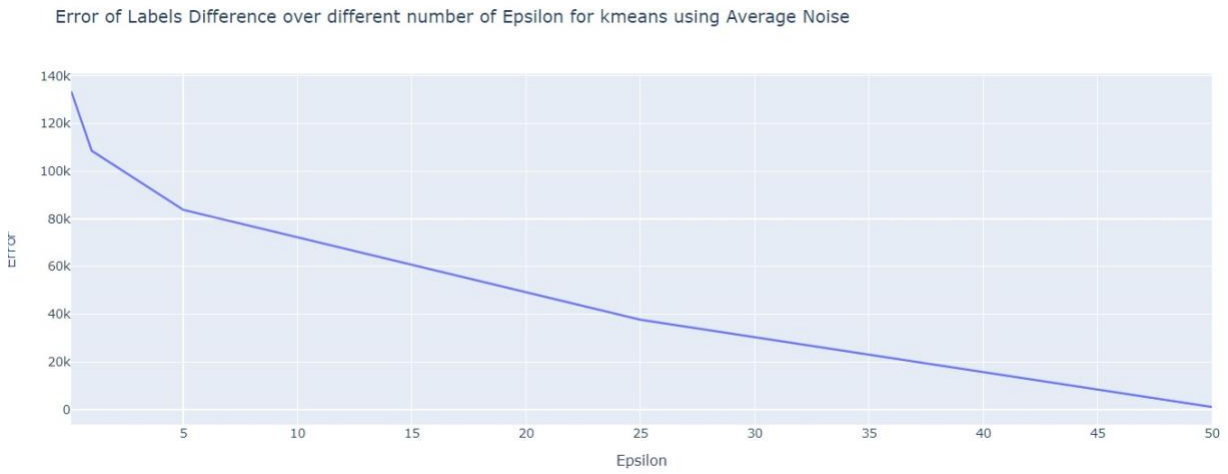


Fig: Average Noise Error Graph

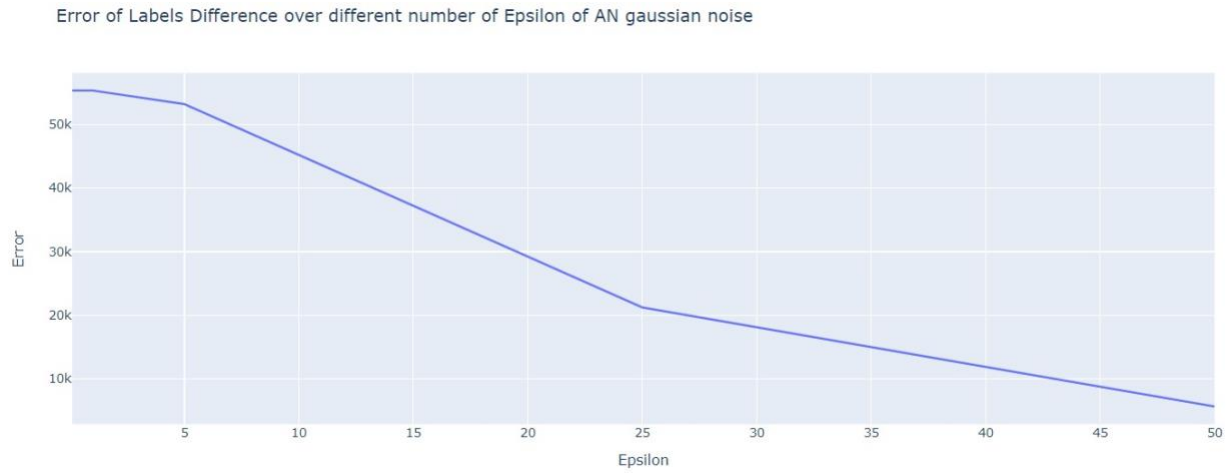


Fig: Gaussian Noise Error Graph

CONCLUSION

This empirical observation underscores the resilience and adaptability of our proposed methodology, making it particularly well-suited for scenarios where traditional methods may encounter challenges, especially in the presence of outlier datasets.

Our research introduces a novel differentially private k-means clustering technique based on the concept of cluster merging. The approach involves initially splitting the data into an abundance of clusters and subsequently merging them to attain the desired number of clusters. This strategy significantly enhances the performance of k-means clustering. The results indicate a substantial improvement

in clustering effectiveness with the incorporation of this cluster merging technique. Moreover, the utility is further enhanced when coupled with budget allocation for privacy.

Through extensive experimentation, our approach consistently outperforms current state-of-the-art algorithms. It's crucial to note that our analysis focuses exclusively on numerical data, and non-numerical or mixed data are not considered in this context. Looking ahead, we envision the development of differentially private k-means clustering algorithms designed to accommodate a broader range of data types.

REFERENCES

1) https://archive.ics.uci.edu/dataset/176/blood+transfusion+service+center
2) C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Foundations and Trends® in Theoretical Computer Science.
3) D. Su, J. Cao, N. Li, E. Bertino, H. Jin, Differentially private k-means clustering, in: Proceedings of the sixth ACM conference on data and application security and privacy.
4) N. Li, L. Min, S. Dong, W. Yang, Differential privacy: From theory to practice, Synthesis Lectures on Information Security Privacy & Trust.
5) Q. Yu, Y. Luo, C. Chen, X. Ding, Outlier-eliminated k-means clustering algorithm based on differential privacy preservation, Applied Intelligence.