applied sciences

MDPI

Article

SSKM_DP: Differential Privacy Data Publishing Method via SFLA-Kohonen Network

Zhiguang Chu [1,2], Jingsha He [1], Juxia Li [2], Qingyang Wang [2], Xing Zhang [2] and Nafei Zhu [1,*]

[1] School of Software Engineering, Beijing University of Technology, Beijing 100124, China
[2] School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China
* Correspondence: eielnut@163.com

Abstract: Data publishing techniques have led to breakthroughs in several areas. These tools provide a promising direction. However, when they are applied to private or sensitive data such as patient medical records, the published data may divulge critical patient information. In order to address this issue, we propose a differential private data publishing method (SSKM_DP) based on the SFLA-Kohonen network, which perturbs sensitive attributes based on the maximum information coefficient to achieve a trade-off between security and usability. Additionally, we introduced a single-population frog jump algorithm (SFLA) to optimize the network. Extensive experiments on benchmark datasets have demonstrated that SSKM_DP outperforms state-of-the-art methods for differentially private data publishing techniques significantly.

Keywords: differential privacy; data publishing; Kohonen network; SFLA; maximum information coefficient

Citation: Chu, Z.; He, J.; Li, J.; Wang, Q.; Zhang, X.; Zhu, N. SSKM_DP: Differential Privacy Data Publishing Method via SFLA-Kohonen Network. *Appl. Sci.* **2023**, *13*, 3823. https://doi.org/10.3390/app13063823

Academic Editors: Konstantinos Rantos, Konstantinos Demertzis and George Drosatos

Received: 15 January 2023
Revised: 14 March 2023
Accepted: 14 March 2023
Published: 16 March 2023

1. Introduction

With the advent of the era of big data and artificial intelligence, massive amounts of data are produced every day with an explosive growth in data scale, such as customer transaction records established by banks, disease information of patient archives by medical institutions, employee salary information recorded by companies, and so on. These data contain a lot of valuable information, and the collection, sharing, mining, and analysis of these data can provide great support for market trend prediction, scientific discovery, and decision-making and the quality of life of the public. However, data are a double-edged sword. While providing a variety of convenient services, they also bring with them the problem of disclosure of users' privacy by releasing data. The released data contain a large amount of sensitive information (such as bank transaction records, patients' medical records, etc.). Although personal identifiers are deleted or encrypted in the process of data release, private information may still be disclosed through mining and analyzing other public information associated with data release. Therefore, protection against users' privacy or sensitive data have become a research hotspot in data release. In order to solve the problem of privacy information disclosure, k-anonymity, l-diversity, t-closeness, and their improved methods [1] are proposed one after another. These methods all effectively prevent attribute link attack, but most of them are difficult to resist background knowledge attack and composite attack. Differential privacy protection methods of privacy are more popular in recent years, as privacy protection technology based on data distortion, without assuming having background knowledge of the attack and attack type, through the strict mathematical model of quantitative intensity of privacy protection, avoids the shortcomings of traditional privacy protection methods and provides stronger protection for privacy information about the data. However, in order to protect the privacy of the original data, most current data publishing methods based on differential privacy introduce a lot of noise, which greatly reduces the availability of published data.

Chen [2] proposed a DP solution based on privacy priority and designed two new indicators, including point confidence and regional average belief, to evaluate its pri-

vacy from a new perspective of privacy preference. However, the dynamic acquisition and release algorithm needs to rely on data distribution and thus faces challenges in the effectiveness and robustness of the algorithm in the face of unknown data distribution. Yan [3] proposed using grid clustering to realize the differential privacy publishing of location-based statistical data to achieve location statistics in the unit of equal size grid, and they designed a bottom-up grid clustering algorithm through the density classification of wavelet transform. However, there are some limitations. The human living environment is mostly based on the distribution of infrastructure, which cannot be well represented by a grid or tree structure and cannot be used to implement an efficient location-based query mode. Zhang [4] proposed a data publishing privacy protection method based on local priority anonymity (LPA), which automatically selects anonymous technology for each anonymous algorithm. Utaliyeva [5] believes that anonymity technology is vulnerable to various attacks and proposed an adaptive differential privacy protection method for structured data. It protects the privacy of sensitive information through machine learning (ML), which solves the privacy–utility trade-off problem. Zhuo [6] proposed an efficient differential privacy spatial information network mechanism that is based on personalized sampling; thus, the network can ensure accurate information privacy while sharing statistical information.

The k-means clustering algorithm is relatively simple and efficient to process the dataset, but it is sensitive to the initial points, the number of clusters k needs to be chosen empirically has a great impact on the clustering effect, and it is extremely sensitive to noise, and the k-means algorithm causes the loss of clustering accuracy. Although the DBSCAN algorithm can find clusters of any shape, and the clustering results are less affected by noise, its clustering effect needs to be improved on high-dimensional data, and it cannot be applied to high-dimensional data. Accordingly, we propose an improved clustering method based on the Kohonen neural network.

There may be complex correlations in the attributes of the data, the correlation between sensitive and non-sensitive attributes can lead to the disclosure of sensitive information, and attackers can infer sensitive information from non-sensitive information. Accordingly, we introduce the maximum information coefficient to measure the relationship between attributes in the data, and according to the correlation between sensitive attributes and non-sensitive attributes, we perturb different degrees of noise to the cluster in which they are located.

Based on the above research, this paper proposes a differential privacy data publishing method (SSKM_DP) based on the SFLA-Kohonen network, which allows the published data to obtain a better privacy protection effect and better availability of the published data. The main contributions of this paper are summarized as follows:

(1) A clustering method based on the SFLA-Kohonen network is proposed, which improves the fitting accuracy of connection.
(2) Weights to training data and the accuracy of clustering results. The validity of the SSKM_DP algorithm is proven theoretically.
(3) Considering that the k-means algorithm is very sensitive to the selection of the initial point, the number of clusters needs to be carefully set empirically, and the DBSCAN algorithm does not work well on high-dimensional data; a clustering method based on the Kohonen neural network was introduced to solve the above problems. In order to initialize the Kohonen network, the single-population frog leaping algorithm (SFLA) was introduced to speed up network convergence.
(4) Considering that there may be complex correlations between attributes in the data, the correlation between non-sensitive attributes and sensitive attributes is bound to lead to the inference of sensitive information from non-sensitive attributes. To solve this problem, we introduced the maximum information coefficient to measure the correlation. An appropriate amount of noise is added to the cluster of non-sensitive attributes to protect non-sensitive attributes and further prevent the private leakage of sensitive data.

(5) In view of the effectiveness of the SSKM_DP algorithm, compared to the algorithms MDAV, IDP_KMENAS, and MDAV_DP on the real datasets NLTCS and UCI Adult, SSKM_DP was carried out in a lot of experiments. Experimental results show that, compared to these similar methods, SSKM_DP not only ensures the privacy of the published data, but it also greatly improves the usability of the published data.

2. Related Works

The privacy data release model based on differential privacy protection is mainly divided into two ways:

(1) Noise is directly added to the original data record, and then the data with noise is released. This method has high privacy protection ability, but it leads to the poor utility of published data.

(2) First, the original data is processed by using compression, transformation, and other technologies, and then noise is added to the processed data. Finally, the data with noise are released. Although this method may lead to a small part of the data information being missing, it greatly improves the effectiveness of published data.

In both methods, the clustering grouping method is used to process the original data, and then the noise is added to each cluster after transformation, which can greatly reduce the noise added to satisfy the differential privacy. At present, there have been some research results from private data publishing methods based on clustering ideas, but these have some problems to some extent.

Soria-Comas et al. [7] combined *k*-anonymity with differential privacy and realized *k*-anonymity [8] through micro-clustering, adding noise to each cluster, realizing the differentiation of differential privacy from individual to cluster, reducing the amount of noise absorbed to satisfy differential privacy, and improving the availability of published data. A differential privacy protection method based on *k*-means clustering was proposed [9], which uses the clustering center point to replace the privacy in the original records. However, this method is limited by the size of the data, and the availability of clustering results is highly dependent on the size of the privacy budget. David et al. [10] carried out micro-clustering according to the level of attributes to improve the homogeneity within the cluster, to reduce information loss, and to improve the availability of data. However, the algorithm has very high computational requirements, which requires high running time and space complexity to process large data. Monedero et al. [11] proposed an efficient micro-aggregation method to anonymize multidimensional numerical data by reducing the number of attributes through principal component analysis. The algorithm realized data privacy protection and improved the utility of published multidimensional numerical data, but it was not applicable to discrete attributes and compound attributes. Xiao et al. [1] defined three different security levels for different sensitive attribute values proposed an *l*-diversity model for multiple sensitive attributes [1,12], and also proposed three greedy algorithms to achieve *l*-diversity for multiple sensitive attributes. This algorithm can solve the problem that information loss increases greatly with an increase in the number of sensitive attributes. Li Yuxi et al. [13] proposed a mobile social network privacy protection scheme supporting the K-nearest neighbor search for the first time, which reduces the communication cost between users and servers and reduces the location information and search pattern leaked to servers. Sensitivity calculation methods based on different center cross-distance clustering have been proposed [14] and so have published data satisfying differential privacy protection. However, the method does not delve further into the more flexible micro-aggregation method. Gu Zhen et al. [15] studied data publishing based on probabilistic principal component analysis, and Chen Si et al. [16] studied data publishing based on a neural network multi-cluster distributed algorithm. Ye et al. [17] proposed an anonymization method to protect the privacy of micro data with multiple sensitive properties through anatomy and arrangement. In this paper, the naïve multi-sensitive buckets and the nearest multi-sensitive buckets are used to anonymize the data. This approach only works for a single release, rather than focusing on multiple releases. Saraswathi et al. [18]

proposed an enhanced t-closeness algorithm for multiple sensitive attributes. The algorithm applies the t-closeness [19] on the MSB k-anonymous clustering attribute layer (MSB-kaca) algorithm and uses the EMD method to avoid the probabilistic reasoning attack caused by bucking. Acs et al. [20] proposed a new method of differential privacy protection based on neural networks, which combine differential privacy and neural networks to generate high-dimensional data satisfying differential privacy. The DP-OPTIC-BASED differential privacy protection method to balance the privacy protection capability and data utility to improve the availability of data was proposed [21]. However, this method is only applicable to numerical data.

Therefore, using the idea of the machine learning model for reference, this paper proposes a differential privacy data publishing method (SSKM_DP) based on the SFLA-Kohonen network, which meets the requirements of differential privacy protection and improves the availability of published data compared to the algorithm in Table 1. SSKM_DP no longer uses the traditional clustering method to cluster data, but it introduces the clustering method of the Kohonen neural network, which avoids the defect of the traditional method requiring the artificial specified number of clustering, and makes the clustering more reasonable. Aiming at the problem of selecting the initial weight of the Kohonen network, the single-population frog leaping algorithm was introduced to optimize the initial connection weight of the Kohonen network to obtain the best initial weight. Considering that there is a complex correlation between insensitive attributes and sensitive attributes of data, the largest information coefficient is introduced as a measure of the correlation intensity. For non-sensitive attributes with relevance, noise is added to further protect sensitive information from disclosure so that the released data can meet the requirements of privacy protection and improve data utility to a large extent.

Table 1. Contrast algorithms.

Algorithm	Main Idea	Limitation
MDAV [7]	By micro-aggregating all attributes to achieve K anonymization, the amount of noise required can be effectively reduced.	Too much noise, poor utility, limited clustering effect, information loss.
IDP_KMENAS [22]	It uses a canopy to select the initial center point and uses the Laplace mechanism to realize the differential privacy protection.	Poor clustering results, poor utility, slow convergence.
MDAV_DP [23]	Adds noise to the micro-aggregated version ofthe original dataset, with the micro-aggregation dataset as our protection target.	Not suitable for complex data; value attribute utility is not considered.

3. Definitions

3.1. Differential Privacy

Differential privacy protection technology adds noise to the original data itself or its transformation in order to achieve the purpose of privacy protection. This method ensures that a record is inserted or deleted from the dataset without affecting the output of the query.

Definition 1 (Differential privacy [24]). *Given two data D and D′, which are identical or differ by at most one record, given a random algorithm A, range(A) represents the range of A, and S is a subset of Range(A). If A satisfies (1), then Algorithm A satisfies ε-differential privacy,*

$$P_r[A(D) \in S] \leq e^{\varepsilon} \times P_r[A(D') \in S] \quad (1)$$

where probability $P_r[\bullet]$ represents the probability of the algorithm, which is determined by algorithm A; ε is the privacy budget, which represents the degree of privacy protection against algorithm A. The smaller the value of ε, the higher the degree of privacy protection for A.

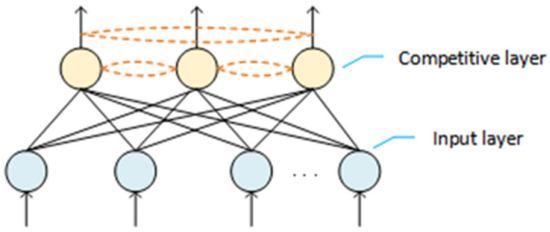
Definition 2 (sensitivity [25]). *Given the query function f: D→Rd, the input data is D, and the output is d-dimensional vector, then the sensitivity is defined as:*

$$\Delta f = \max_{d(D,D')=1} \|f(D) - f(D')\|_1 \quad (2)$$

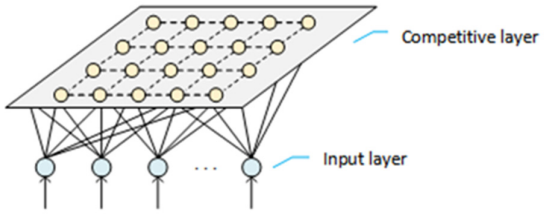
where $\| \cdot \|_1$ denotes the L1 norm.

3.2. Kohonen Network

The Kohonen network, namely Self-Organization Feature Map (SOFM), is a self-organizing competitive neural network proposed by Kohonen et al. in 1981, which is an unsupervised learning model [26]. The Kohonen network is a neural network of an input layer and a competing layer (output layer) that realizes the bidirectional link between two layers through a full connection. Each node in the competing layer represents an aggregated class and connects adjacent nodes through weight. Under the premise of no prior knowledge, the "competitive learning" method is used to identify the rules and relationships between the input samples and realize the clustering of the samples. The topology of the Kohonen network is shown in Figure 1.



(**a**) One-dimensional linear matrix.



(**b**) Two dimensional linear arrays.

Figure 1. Kohonen neural network topology.

The core idea of the Kohonen network is that when the Kohonen network receives the input vector, the input vector is automatically divided into different nodes, and each node of the competitive layer responds to the input in a ”competitive” way, obtains a winning node, and updates the weight of the node’s neighborhood. Through repeated learning and training for input vectors, the distribution of connection weight between nodes in the competitive layer is close to the input value; thus, the input vector with correlation can obtain clustering results from the competitive layer.

The steps of data clustering in the Kohonen network are as follows:

(1) Initializing The Network

The connection weight W_j of each neuron is set in the input layer I and the competition layer. W_j is usually a random number in the range (0, 1). The initial value of the learning rate $\eta(0)$ is determined, and its value range is (0, 1). The maximum learning time T is set.

(2) Looking For Winning Neurons

For the input vector I_i, the most matching neuron in the competition layer should be searched for and the winning neuron determined. The matching degree is measured by Euclidean distance. The smaller the distance, the higher the matching degree. The calculation method is shown in (3):

$$d_j = \|I - W_j\| = \sqrt{\sum_{i=1}^{n} [I_i(t) - W_{ij}(t)]^2} \tag{3}$$

where W_{ij} is the connection weight between the ith neuron in the input layer and the jth neuron in the competition layers.

(3) Adjusting And Updating The Weight

According to the winning neuron and the neighborhood function, the winning neighborhood of the winning neuron is determined, all neurons in the winning neighborhood are found out, and the weight of these neurons is adjusted. The updating method is shown in (4):

$$\begin{aligned} W_{ij}(t+1) &= W_{ij} + \Delta W_{ij} \\ &= W_{ij} + \eta(t) * N_{ij} * [I_i(t) - W_{ij}(t)] \end{aligned} \tag{4}$$

where N_{ij} represents the domain function, and $\eta(t)$ represents the learning rate at time t, which decreases with the increase of t.

(4) Iterating The Process

The learning rate η is updated to determine whether η reaches the preset condition or whether the learning time t reaches the maximum learning time T. If $\eta \leq \eta_{\min}$ or $t = T$, the iteration ends and the clustering is completed. Otherwise, step (2) is returned until the end of the iteration.

3.2.1. Leap Frog Algorithm

The shuffled frog leaping algorithm (SFLA) [27] is a new and effective bionic swarm intelligence optimization algorithm that was proposed by Eusuff et al. to simulate the behavioral interaction of frog groups foraging [28]. The SFLA algorithm combines the advantages of the particle swarm optimization algorithm [29] (PSO) and meme calculus algorithm (MA) and has the characteristics of fewer parameters, a fast computation speed, and strong global optimization ability.

The basic idea of the SFLA algorithm is that there are N frogs living in a wetland, and they find the place with the most food by jumping over different rocks. Each frog is defined as a feasible solution, and N frogs are divided into different subgroups according to specific rules. Each frog has its own decision information, and it evolves from the subgroups by communicating with each other, and the subgroups evolve accordingly (local search). After the evolution to a certain extent, the information about the subgroups is exchanged until the algorithm meets the convergence condition (global search). A schematic diagram of the SFLA algorithm is shown in Figure 2.

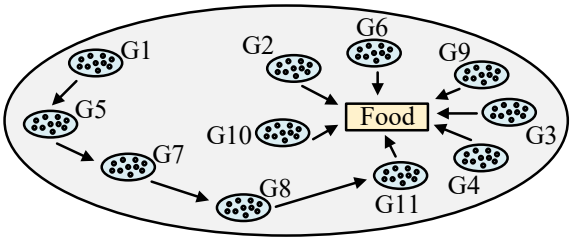


Figure 2. Schematic diagram of the SFLA algorithm.

The workflow of the SFLA algorithm optimization can be divided into four steps:

(1) Population Initialization

An initial population $R = \{X_1, X_2, \ldots, X_N\}$ consisting of N frogs is randomly generated; the ith frog is denoted as $X_i = \{A_1, A_2, \ldots, A_k\}$, and k is the dimension of the frog.

(2) Subgroup Division

After the frog population R is generated, the fitness value *f(i)* of all frogs in R is calculated, and the frog with the highest fitness value is derived as the frog X_g with the optimal population. N frogs should be ranked in descending order of *f(i)* and R divided into P subpopulation: $\{S_1, S_2, \ldots, S_p\}$, each subpopulation containing q frogs, satisfying $N = p \times q$.

(3) Local Search

After dividing the population, the frogs with the worst fitness value and the frogs with the best fitness value in each subpopulation are labeled as X_w and X_b, respectively. Frogs with the worst fitness position in each subpopulation are cyclically updated according to (5) and (6):

$$D = rand() \times (X_b - X_w) \tag{5}$$

$$X'_w = X_w + D,\ D_{\min} \le D \le D_{\max} \tag{6}$$

where $rand()$ represents the random number in the range (0, 1), D represents the leapfrog step, and $D_{\min}$ represents the minimum and $D_{\max}$ represents maximum leapfrog step.

After the frog position is updated, if the updated frog is better than the current frog X_w, then $X\prime_w$ replaces X_w; if the new frog is not better than the current frog, then frog X_g, the optimal frog of the population, replaces frog X_b. If no better than the current fitness value is obtained, a new frog $X\prime_w$ is randomly generated to replace X_w.

When the P subgroup completes the local search, all frogs are remixed and reordered according to the fitness value. The molecular group is reclassified, and the local search is carried out again until the maximum number of iterations or the required convergence condition is reached. The algorithm terminates and the optimal frog X_g of the population is output.

3.2.2. Maximum Information Coefficient

The Pearson coefficient, Spearman coefficient, mutual information (MI), and k-nearest distance (KNN) are often used to measure the degree of correlation between two attributes. However, the Pearson coefficient cannot measure nonlinear and non-functional relations. Although the Spearman coefficient can be applied to simple monotone nonlinear relationships, its statistical efficiency is low. The mutual information has weak computing power for continuous variables, has low accuracy, and cannot compare the calculation results of different data. KNN needs to calculate the distance between each sample and all sample points to obtain its k nearest neighbors, which requires a large amount of calculation. Maximal Information Coefficient (MIC) [30] is a new method to measure the correlation between variables based on mutual information and meshing proposed by Reshef et al. in 2011, which can overcome the shortcomings of the above methods. It captures the linear, nonlinear, and non-functional relations among attributes more accurately and has

the advantages of universality, balance, and low computational complexity. The pairs of common coefficients are shown in Table 2.

Table 2. Comparison of common coefficients.

	Scope of Application	Standardized	Computational Complexity	Robustness
Pearson coefficient	Linear data	Yes	Low	Low
Spearman coefficient	Linear data simple monotone nonlinear data	Yes	Low	Medium
KNN	Linear data nonlinear data	No	High	High
MIC	Linear data nonlinear data	Yes	Low	High

The specific definition of MIC is described as follows:

Definition 3 (Maximum information coefficient)**.** *Given order to the data D,* $X = \{x_i, i = 1, 2, \ldots, n\}$ *and* $Y = \{y_i, i = 1, 2, \ldots, n\}$ *are the two variables in D,* x_i *and* y_i*, respectively, according to the value of a mesh of* a × b*. There are many kinds of* a × b *meshing, respectively, used to calculate the mutual information of each grid under different division* $I(X : Y)$*, selecting different divisions under the maximum mutual information of* $Max(I(X : Y))$*. The largest information coefficient is defined as shown in (7).*

$$MIC(X : Y) = \max_{a \times b \leq B} \frac{Max(I(X : Y))}{\log_2 \min(a, b)} \tag{7}$$

In the formula, B is the upper limit of the a × b *grid, generally* $n^{0.6}$.

In this paper, the maximum information coefficient was used to measure the correlation between sensitive attributes and between sensitive attributes and non-sensitive attributes in the data. The greater the value of MIC, the stronger the correlation between attributes; conversely, the smaller the value of MIC, the weaker the correlation between attributes.

4. The Proposed Data Publishing Method

4.1. Description of Problem

The general data method based on differential privacy protection is the original data by differential privacy protection, releasing a private dataset that users can use to perform any query operation of general data, but this method of the original data for privacy protection adds a lot of noise and greatly reduces the release data utility. By reducing the sensitivity of differential privacy and allocating the privacy budget reasonably, the amount of noise added to satisfy differential privacy can be effectively reduced, and the availability of published data can be improved.

Most existing methods do not consider the complex correlation between attributes in the data. When adding noise to sensitive attributes in the data, the correlation between sensitive attributes and non-sensitive attributes in the data should be considered, and then the non-sensitive attributes with a strong correlation with sensitive attributes should be protected.

Based on the above problems, this paper proposes a differential privacy data publishing method SSKM_DP based on the SFLA-Kohonen network. This method conducts a clustering operation on the original data, reduces the query sensitivity, and reduces the intake of noise while reducing the data dimension, and then it determines the correlation between attributes. The noise required by differential privacy is added to protect the privacy. When the same differential privacy protection effect is achieved for the published

data generated by the SSKM_DP method, less noise is added and the availability of the data is better.

4.2. SSKM_DP Multi-Sensi tive Attribute Data Publishing Mechanism

The operation mechanism of the differential privacy data publishing method based on the SFLA-Kohonen network is shown in Figures 3 and 4 as a detailed flow chart of the proposed method.

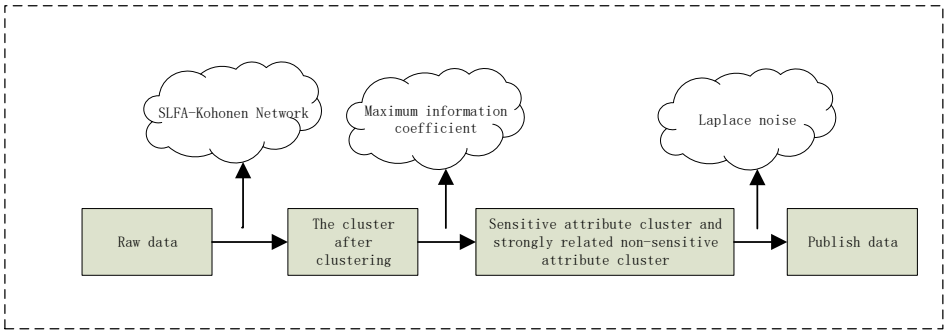


Figure 3. SSKM_DP data publishing framework satisfying differential privacy protection.

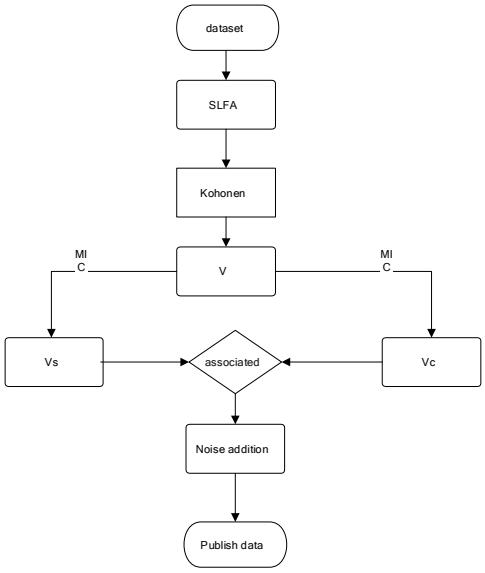


Figure 4. Detailed flow chart of the proposed method.

The steps of the SSKM_DP data publication method are as follows:

(1) Attribute Clustering

The Kohonen network is optimized, the original data are clustered by using the improved SFLA-Kohonen network, and the data are reasonably divided into multiple sub-data to achieve the differentiation of sensitive attributes from individuals to groups to reduce the data and query sensitivity and reduce the noise required to meet the differential privacy.

(2) Attribute Correlation Judgment

Part of the sensitive attribute exists on a strong affinity, by inferring sensitive attributes, introducing the largest information coefficient sensitive to data with the sensitive attribute. The connection between each child data clustering partition cluster with the sensitive property has a strong correlation between the sensitive attributes. Add an appropriate amount of noise to the subdataset cluster to protect such non-sensitive attributes and further prevent the privacy leakage of sensitive data.

(3) Data Noise

The privacy budget satisfying differential privacy is allocated to the subset cluster obtained by SFLA-Kohonen network clustering. Then, the corresponding noise is added to the cluster of sensitive attributes and the cluster of non-sensitive attributes associated with sensitive attributes, to reduce the required noise amount and improve the availability of data.

Algorithm 1 is the process algorithm for proposing the model.

Algorithm 1 SSKM-DP

Input: dataset $U = \{x_1, x_2, \ldots, x_n\}$, the number of neurons in the input layer of Kohnen's network t, the number of frogs N, learning rate η, Maximum learning times T.
Output: published dataset $\widetilde{U} = \{x_1, x_2, \ldots, x_n\}$.
1: $W_{ij} \leftarrow$ $SFLA$ optimizes the initial weight of Kohonen network $(N,\ t)$
2: $FModel \leftarrow SFLA -$ Kohonen network to achieve data clustering data clustering $\left(W_{ij}, \eta, T\right)$
3: $V = v_1, v_2, \ldots, v_m \leftarrow FModel(U)$
4: $V_c = v_{c1}, v_{c2}, \ldots, v_{cq}$ and $V_s = v_{s1}, v_{s2}, \ldots, v_{sp} \leftarrow$ Attribute correlation determination method
5: published dataset $\widetilde{U} = \{x_1, x_2, \ldots, x_n\} \leftarrow Noise(V_c, V_s)$

4.3. SFLA-Kohonen Data Clustering Algorithm

The general data publishing method based on differential privacy protection is to add noise to each record of the data to meet the differential privacy protection and publish universal data where data users can perform any query operation. However, this method introduces a large amount of noise, which greatly reduces the availability of published data. By reducing the sensitivity of differential privacy and allocating the privacy budget reasonably, the amount of noise added to satisfy differential privacy can be effectively reduced, and the availability of published data can be improved. The literature [31] points out that the method of clustering or grouping is used to process the original data, and then noise is added to each cluster after conversion, which can greatly reduce the amount of noise added to satisfy the differential privacy. Based on this, the idea of clustering was introduced in this paper to divide data attributes into clusters, reduce the sensitivity of differential privacy, and reduce the required intake noise.

The traditional and classical clustering methods include k-means [32] and the DBSCAN [33] algorithm, but both of them and their improved methods have some problems. k-means is very sensitive to the selection of the initial point, and the number of clustering k is artificially selected according to experience. This setting method is extremely unreasonable, resulting in different clustering results, which are bound to result in insufficient or excessive privacy protection ability and reduce the availability of released data. Although DBSCAN does not need to set the number of clusters and has high robustness, it is unable to obtain better clustering results from data on many dimensions. Aiming at the limitations of the above methods, a clustering method based on the Kohonen neural network was introduced in this paper, and the neural network model was combined with differential privacy to improve the privacy protection ability of sensitive data and the utility of published data.

However, in the training process of the Kohonen network, the initial connection weight must be specified in advance, which depends on the setting of experience, and the accuracy of clustering results depends very much on the selection of the initial connection weight. Aiming at the shortcomings in the clustering method based on the Kohonen network, the single population frog leaping algorithm (SFLA) was used to optimize the initial connection weight of the Kohonen network, and a clustering method based on the SFLA-Kohonen network was proposed to improve the fitting accuracy of connection weight to training data and the accuracy of clustering results.

Algorithm 2 of the initial optimization process of Kohonen networks using SFLA is as follows:

Algorithm 2 SFLA optimizes the initial weight of Kohomen network

Input: data the number of neurons in the input layer of Kohomen's network; the number of frogs.
Output: the optimal initial weight of the SOM nwtwork.

1: $R = \{X_1, X_2, \ldots X_N\} \leftarrow X(s)\frac{1}{\sqrt{2\pi\sigma}}e^{(-\frac{(s-u)^2}{2\sigma^2})}$
2: $fit(X_i) = \frac{1}{1+E\left[\sum_{a1,a2,} N_x(b_1-a_1,b_2-a_2)x_i-W(a_1,a_2)\right]}$
3: **for** $t = 0 \rightarrow T$ **do**
4: $D = rand() \times (X_b - X_w)$
5: $X'_W = X_W + D$
6: **if** $fit(X'_W) > fit(X_W)$ then
7: $X_W = X'_W$
8: **end if**
9: **end for**
10: return $X_g \rightarrow SOM$ network

Input: data $U = \{x_1, x_2, \ldots, x_n\}$; the number of neurons in the input layer of the Kohonen network; the number of frogs.

Output: The optimal initial weight of the SOM network.

Step 1.The initial population is generated composed of N frog $R = \{X_1, X_2, \ldots, X_N\}$, and the generation method follows the Gaussian distribution formula, as shown in (8):

$$X(s) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{(s-u)^2}{2\sigma^2}) \tag{8}$$

where $\mu = 0, \sigma = 1$.

Step 2. After the frog population R is generated, all frogs are substituted for the Kohonen network model. The input vectors are randomly selected to calculate the fitness value of all frogs in R, *fit*(X_i). The fitness calculation method used in this paper is shown in (9):

$$fit(X_i) = \frac{1}{1 + E[\sum\limits_{a_1,a_2} N_x(b_1 - a_1, b_2 - a_2)\|x_i - W(a_1, a_2)\|]} \tag{9}$$

where E is the mathematical expectation, $N_x(\ )$ is the domain function, $W(a_1, a_2)$ is the weight of the neuron (a_1, a_2), and (b_1, b_2) represents the coordinate of the winning neuron in U.

Step 3. N frogs are ranked in descending order of *fit* (X_i) to obtain frog X_w with the worst fitness value and frog X_b with the best fitness value. Frogs with the worst fitness value of a cycle are ranked according to position update (10) and (11):

$$D = rand() \times (X_b - X_w) \tag{10}$$

$$X'_w = X_w + D \tag{11}$$

where *rand*() represents a random number in the range (0, 1) and D represents the leapfrog steped size.

Step 4. The fitness value is calculated after the frog position is updated. If the updated frog is better than the current frog X_w, X'_w replace X_w and retain the updated frog's parameters. If the updated frog is not better than the current frog, keep the current frog's parameters.

Step 5. When the maximum number of iterations or the required convergence condition are reached, the algorithm is terminated and the optimal frog X_g of the population is output. The parameter of the optimal frog is taken as the initial weight of the SOM network.

There is no need to set the number of clustering clusters when the SFLA-Kohonen network is used for clustering, and the clustering results have better accuracy and rationality. Secondly, the adjacent relation is imposed on the center of mass of the cluster, resulting in higher homogeneity within the cluster. At the same time, the SFLA-Kohonen network

has good self-stability and strong anti-noise ability, which makes the cluster sensitivity formed by clustering low to reduce the noise required by differential privacy and improve the availability of data.

The Algorithm 3 for the data clustering process using the SFLA-Kohonen network is as follows:

Algorithm 3 SFLA—Kohonen networks to achieve data clustering

Input: dataset $U = \{x_1, x_2, \ldots x_n\}$; the learning rate is and its value range is (0,1); Maximum learning times T
Output: Clusters formed by clustering $V = \{v_1, v_2, \ldots v_m\}$.
1: $W_{ij} \leftarrow X_g$
2: **for** $\eta < \eta_{max}$ or $t < T$ **do**
3: calculate $d_j = \sqrt{\sum_{i=1}^{n} \left[x^i(t) - W_{ij}(t)\right]^2}$
4: Obtain new winning neurons, update W_{ij}
5: $W_{ij}(t+1) = W_{ij} + \eta(t) * N_{j,c(x)} \left[x_i(t) - W_{ij}(t)\right]$
6: $\eta(t) = \eta(0)e^{-t/T}$
7: **end for**
8: **return** FModel

Input: Dataset $U = \{x_1, x_2, \ldots, x_n\}$; the learning rate is η, and its value range is (0, 1); maximum learning times T.

Output: Clusters formed by clustering $V = \{v_1, v_2, \ldots, v_m\}$.

Step 1. The value of optimal frog X_g obtained in Algorithm 1 is set as the initial connection weight W_{ij} of each neuron in the input layer I and the competition layer of the Kohonen network. In this paper, the Gaussian function was adopted as the domain function, and its definition is shown in (12):

$$N_x = \left\{ \begin{array}{cc} \exp(-\frac{\|d_i - d_j\|^2}{2\delta^2}) & d_i - d_j \le \delta \\ 0 & d_i - d_j > \delta \end{array} \right\} \tag{12}$$

Step 2. The Euclidean distance d_j is calculated from all input neurons x_i and neurons in the competition layer at time t, as shown in (13):

$$d_j = \sqrt{\sum_{i=1}^{n} \left[x_i(t) - W_{ij}(t)\right]^2} \tag{13}$$

The neuron with the smallest Euclidean distance is obtained to determine the winning neuron.

Step 3. The winning neighborhood of the winning neuron is obtained according to the domain function. The weight of all neurons is adjusted in the winning neighborhood according to Equation (14):

$$W_{ij}(t+1) = W_{ij} + \eta(t) * N_{j,c(x)} * [x_i(t) - W_{ij}(t)] \tag{14}$$

where $\eta(t)$ represents the learning rate at time t, which decreases with the increase of t.

The $\eta\ (t)$ function used in this paper is shown in Equation (15):

$$\eta(t) = \eta(0)e^{-t/T} \tag{15}$$

Step 4. The learning rate η is updated and whether η reaches the preset condition or whether the learning times t reaches the maximum learning time T is determined. If $\eta \le \eta_{\min}$ or $t = T$, then the iteration ends and the clustering is completed; Otherwise, Step 2 is repeated until the end of the iteration.

Step 5. The trained model FModel is obtained and input U into FModel, and the cluster set is obtained by clustering $V = \{v_1, v_2, \ldots, v_m\}$.

4.4. Attribute Correlation Determination Method

There may be complex correlations between attributes in the data. Some attributes are correlated to each other, while some attributes are independent of each other. If there is a relationship between a non-sensitive attribute and a sensitive attribute, it is likely that sensitive information can be inferred from the non-sensitive attribute. Therefore, when noise is added to sensitive attributes in the data, it is necessary to consider the correlation between the attributes in the data.

The common methods used to measure the degree of attribute correlation are the Pearson coefficient, mutual information, and k-nearest distance. However, Pearson's coefficient cannot measure nonlinear relationships and non-functional relationships. The mutual information has weak computing power for continuous variables, has low accuracy, and cannot compare the calculation results of different data. KNN needs to calculate the distance between each sample and all sample points, which requires a large amount of calculation. The maximum information coefficient can overcome the shortcomings of the above methods and reflect the correlation degree between attributes more accurately. Therefore, when the SSKM_DP algorithm measures the connection strength between attributes, the maximum information coefficient is adopted as the metric index.

Definition 4 (Connection strength)**.** *Given the properties z_i andz_j, the calculation method to define the connection strength between them is shown in Equation (16):*

$$CS(z_i : z_j) = MIC(z_i : z_j) \tag{16}$$

where MIC is the maximum information coefficient between attribute$z_i$andz_j.

Algorithm 4 for finding non-sensitive attributes linked to sensitive attributes is described below:

Algorithm 4 Attribute correlation determation method

Input: cluster formed by SFLA-SOM network clustering $V = \{V_1, V_2, \ldots\ldots V_m\}$; Connection strength threshold CS_{Tsh}
Output: Clusters with sensitive attributes Vs; there exists cluster V_c with non-sensitive attributes strongly connected to sensitive attributes.
1: Mark all sensitive attributes xs in the data
2: V_s add $V_i(xs_i)$
3: Calculate Connection strength $CS\left(xs_i : xv_j\right)$
4: $CS(xs_i : xv_i) = \max\limits_{a \times b < B} \frac{Ma(I(xs_i : xv_i))}{log_2 \min(a,b)}$
5: **if** $CS\left(xs_i : xv_j\right) \leq CS_{Tsh}$ **then**
6: Vc add $V_i(xv_i)$
7: **end if**
8: **return** $Vs = \{v_{s1}, v_{s2}, \ldots\ldots V_{sp}\}$, $Vc = \{v_{c1}, v_{c2}, \ldots\ldots v_{cpq}\}$

Input: Cluster formed by SFLA-SOM network clustering $V = \{v_1, v_2, \ldots, v_m\}$; connection strength threshold $CS_{T\mathrm{sh}}$.

Output: Clusters with sensitive attributes Vs; there exists cluster V_{c} with non-sensitive attributes strongly connected to sensitive attributes.

Step 1. All sensitive attributes xs are marked in the data.

Step 2. The connection strength between each sensitive attribute xs_j and the non-sensitive attribute xv_i of other subset clusters $CS(xs_i : xv_j)$ is calculated, as shown in Equation (17):

$$\begin{aligned} CS(xs_i : xv_j) &= MIC(xs_i : xv_j) \\ &= \max_{a \times b \leq B} \frac{Max(I(xs_i : xv_j))}{\log_2 \min(a,b)} \end{aligned} \tag{17}$$

Step 3. It is determined $CS(xs_i : xv_j)$ whether the connection strength reaches the threshold of the CS_{Tsh} connection strength. If $CS(xs_i : xv_j) \leq CS_{Tsh}$, it indicates that there is a strong connection between them; otherwise, they are considered to be only weakly connected and are not marked.

Step 4. Clusters with sensitive attributes and clusters $Vs = \{v_{s1}, v_{s2}, \ldots, v_{sp}\}$ with non-sensitive attributes with a strong connection $Vs = \{v_{s1}, v_{s2}, \ldots, v_{sp}\}$ are obtained according to the results of tags in $Vc = \{v_{c1}, v_{c2}, \ldots, v_{cq}\}$.

4.5. Data Noise

Satisfying differential privacy noise added to each cluster after conversion can be greatly reduced compared to adding to each record. The privacy budget satisfying differential privacy is allocated to the clustering center of the subset cluster formed by the SFLA-Kohonen network clustering, and then the corresponding noise is added to the clustering center of the cluster where the sensitive attribute is located and the cluster with the non-sensitive attribute is associated with the sensitive attributes. For the cluster center of each cluster, the calculation method is described as follows:

Given data composed of n records $U = \{x_1, x_2, \ldots, x_n\}$, each record has q attributes, U forms, and m clusters through SFLA-Kohonen network clustering. Assuming A_i^q that the attribute completes the clustering, there are m_j records in the clustering $v_j (j = 1, 2, \ldots, m)$. The calculation of the clustering center is shown in Equation (18):

$$Center(v_j(A_i^q)) = \frac{\sum_{p=1}^{m_j} v_{jp}(A_i^q)}{m_j} \tag{18}$$

where $v_{jp}(A_i^q)$ is the A_i^q value of p records in v_j, and m_j represents the number of records in v_j.

The Laplace mechanism is used to add noise to each cluster center to make it meet differential privacy protection, and the *Ue* of differential privacy data is generated. The method of adding noise is shown in Equation (19):

$$Noise(v_j(A_i^q)) = Center(v_j(A_i^q)) + Y \tag{19}$$

where $Y \sim Lap(\Delta f/\varepsilon)$ is the random noise, obeying the Laplace distribution of the scale parameter $\Delta f/\varepsilon$.

5. Analysis of Privacy Protection Effect of the Algorithm

Theorem 1. *SSKM_DP algorithm satisfies the ε-differential privacy.*

Proof of Theorem 1. Given two adjacent data U_1 and U_2, the output of the SSKM_DP algorithm is *A* (U_1) and *A* (U_2), respectively, and $\widetilde{U}$ is the differential privacy data. According to the definition of differential privacy, the following equation is proven to be true:

$$\frac{P_r(A(U_1) \in S)}{P_r(A(U_2) \in S)} \leq \exp(\varepsilon) \tag{20}$$

Assume that the query results of U_1 and U_2 are $f(U_1)$ and $f(U_2)$, respectively, and $f(\widetilde{U})$ is the query results of $\widetilde{U}$.

$$P_r(A(U_1) \in S) \propto \exp\left(\frac{\varepsilon\left|f(\widetilde{U}) - f(U_1)\right|}{\Delta f}\right)$$

then

$$\begin{aligned}\frac{P_r(A(U_1)\in S)}{P_r(A(U_2)\in S)} &= \frac{\exp\left(\frac{\varepsilon|f(\widetilde{U})-f(U_1)|}{\Delta f}\right)}{\exp\left(\frac{\varepsilon|f(\widetilde{U})-f(U_2)|}{\Delta f}\right)} \\ &\leq \exp\left(\frac{\varepsilon|f(U_1)-f(U_2)|}{\Delta f}\right) \\ &\leq \exp\left(\frac{\varepsilon\|f(U_1)-f(U_2)\|_1}{\Delta f}\right) \\ &\leq \exp(\varepsilon)\end{aligned} \tag{21}$$

In the SSKM_DP algorithm, there is no intersection among the m clusters generated. According to the parallel combinatorial property of differential privacy, the privacy budget ε_i allocated by the SSKM_DP algorithm for each cluster is the overall privacy budget ε of SSKM_DP.

The conclusion is that the SSKM_DP algorithm satisfies ε-differential privacy. □

6. Experimental Evaluation

Three advanced methods, namely MDAV [7], IDP_KMENAS [22], and MDAV_DP [23], were compared by designing experiments to measure the effectiveness and availability of the SSKM_DP algorithm.

6.1. Experimental Environment

In this experiment, Python programming language is used to implement the proposed method and the comparison method. The specific setting of the experimental environment is shown in Table 3.

Table 3. Experimental environment information.

Hardware and Software Information	Specific Configuration
CPU	Intel(R) Core(TM) i5-9400F CPU(2.90 GHz)
Memory	16 GB
The operating system	Win10 64-bit
The development environment	PyCharm-professional-2021
Programming language	Python 3

6.2. Experimental Data

Two data that are widely used in the research field of privacy data release, namely NLTCS and UCI Adult, were used in the experiment. NLTCS is data from the Nursing Center Nursing Survey of the United States, which records information about the daily care of 21,574 patients. Adult is census data from the US Census Center, recording 48,842 pieces of personal information. Specific information about data type, number, and size of attributes of the two experimental data is shown in Table 4.

Table 4. Data information.

Datasets	Type	Number of Attributes	Date Size
NLTCS	Binary	16	21,574
Adult	Non-binary	14	48,842

6.3. Experimental Evaluation Indexes

This experiment used mean square error (MSE) and record linkages (RL) to evaluate the performance of the SSKM_DP algorithm. In the SSKM_DP algorithm, the data utility is measured by the information loss caused by the noise added to the original data to satisfy differential privacy. Information loss is generally quantified by the mean square error.

The mean square error (MSE) is defined as the mean sum of the squares of the attribute distance errors between the published data Ue, satisfying differential privacy and the original data U. The calculation method is shown in Equation (22):

$$MSE = \frac{\sum_{u_j} \sum_{a_j^i \in u_j} \left[d_j(a_j^i, (a_j^i)_e) \right]^2}{n} \tag{22}$$

In the formula, $d_j()$ represents the Euclidean distance defined by Equation (3); u_j is an attribute of dataset U, and the a_j^i and $(a_j^i)_e$ distribution represents the ith attribute value of the jth record and its corresponding record to be published. The larger the MSE, the more serious the information loss and the lower the availability of published data.

In the SSKM_DP algorithm, the privacy protection ability is measured by information disclosure. Disclosure is defined as the percentage of the original record that correctly matches the record in the published dataset. Information disclosure is usually represented by the recorded association. The smaller the RL, the lower the degree of information disclosure and the higher the ability of privacy protection. The calculation method is shown in Equation (23):

$$RL = \frac{\sum_{u \in U} P_r(u_e)}{n} \times 100\,\% \tag{23}$$

In the formula, $P_r(u_e)$ represents the probability of association of published record u_e, and the formula is as follows:

$$P_r(u_e) = \begin{cases} \frac{1}{|U_e|}, u \in U_e \\ 0, u \notin U_e \end{cases} \tag{24}$$

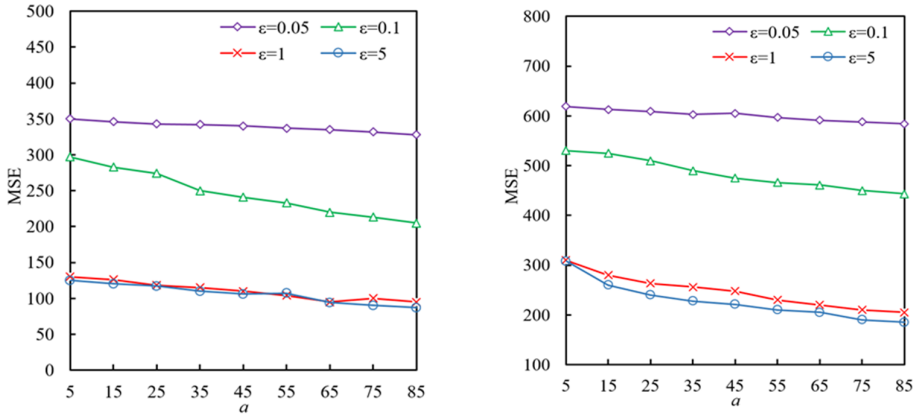
6.4. Analysis of Experimental Results

In order to verify the availability of SSKM_DP, MDAV [16], IDP_KMENAS [17], MDAV_DP [18], and SSKM_DP algorithms were compared on two data, respectively.

In the experiment, the value of the privacy budget ε was {0.05, 0.1, 1, 5}, and the number of data attributes m was {5, 10}. In order to decrease the error caused by the experiment, 20 experiments were carried out for the four algorithms on two data, respectively, and the average value of the 20 experiments was taken as the final experimental result.

For the UCI Adult data, the number of different attributes of m was set to evaluate the data utility through the SSKM_DP algorithm. The experimental results of data utility are shown in Figure 4.

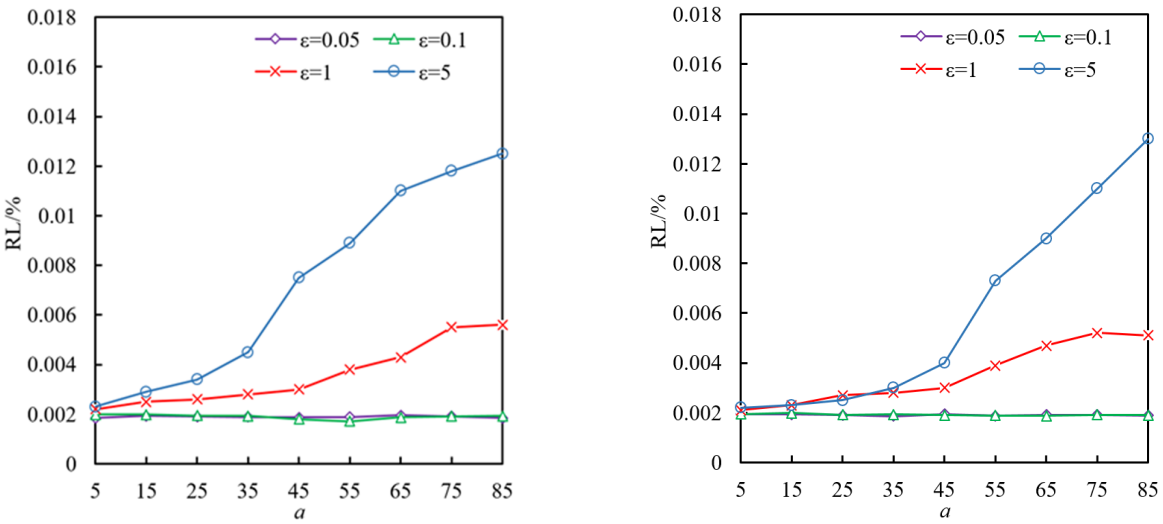
As can be observed in Figure 5, for Adult data, when the value of ε increased from 0.05 to 5, the value of information loss MSE decreased gradually. When ε was 0.05, the MSE value changed slightly with the increase of the number of clustering a, but the MSE value was still very low, indicating that the availability of data released through the SSKM_DP algorithm was also very low when the intake noise was very high. When the value of ε was {1, 5}, the value of MSE was large and the availability of data was greatly improved. Since the clustering scale of the SFLA-Kohonen network is not subject to artificial constraints, it is completely dependent on the network topology mapping relationships. As the number of clusters a increased, the MSE value did not change significantly. Therefore, the SSKM_DP algorithm has a good anti-noise performance.



(**a**) m = 5 (**b**) m = 10

Figure 5. Change trend of SSE for the Adult dataset.

For the UCI Adult data, the number of different attributes of m was set to evaluate the privacy protection ability of the SSKM_DP algorithm. The experimental results of the privacy protection ability are shown in Figure 6.



(**a**) m = 5 (**b**) m = 10

Figure 6. Change trend of RL for the Adult data.

As can be seen from Figure 6, for the Adult data, when the value of ε was {0.05, 0.1}, even if the value of attribute number m and cluster number a changed, the value of the record correlation RL did not change significantly. When the value of ε was 5, the value of RL also increased with the increase of a, the risk of information disclosure gradually increased, and the ability of privacy protection decreased. As can be seen from Figures 4 and 5, when the privacy budget ε was 1 and 5, the SSKM_DP algorithm had good data utility. When ε was 1, the privacy protection ability of SSKM_DP algorithm was much better than that of ε as 5. Therefore, this paper took ε = 1 as the optimal privacy budget value of the SSKM_DP algorithm.

For NLTCS and UCI Adult data, the value of the privacy budget ε was set as 1, and the number of different attributes of m was taken. The SSKM_DP algorithm compares with MDAV [25], IDP_KMENAS [26], and MDAV_DP [27]. Experimental results on information loss are shown in Figure 7.

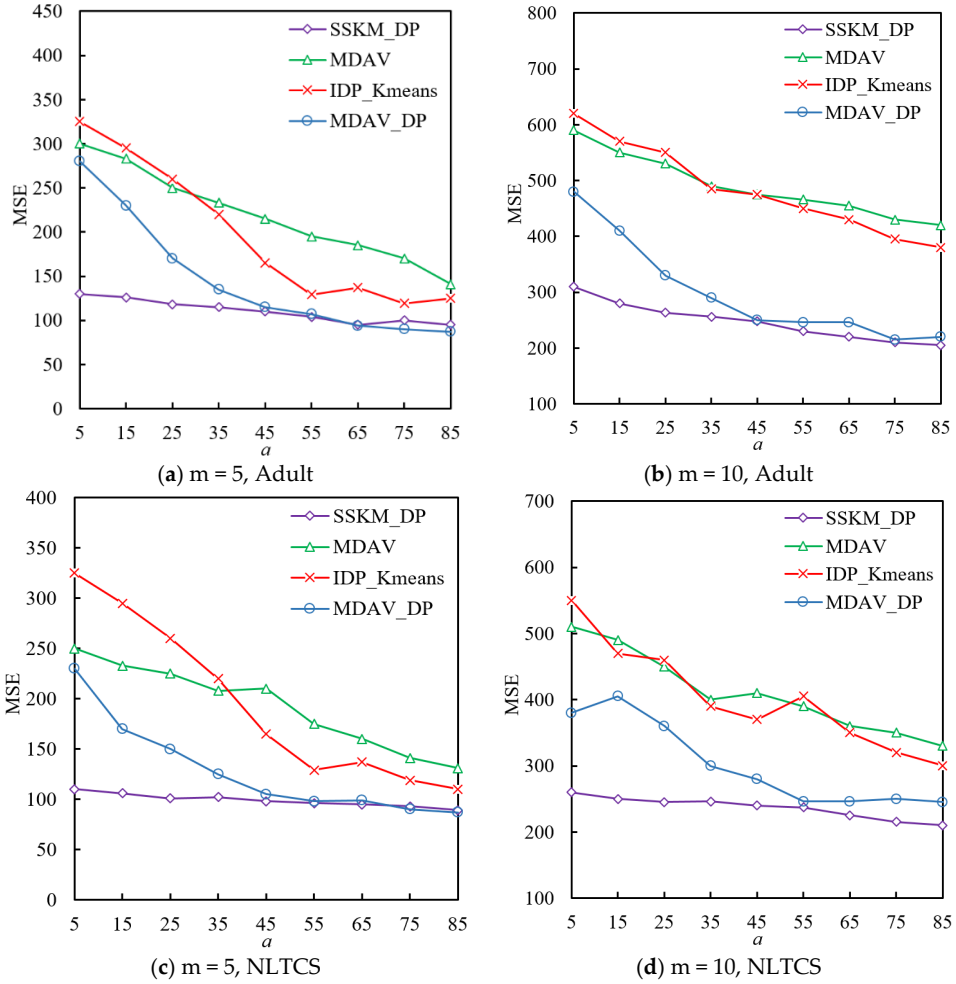


Figure 7. SSE trends of the four algorithms for Adult and NLTCS datasets.

As can be seen from Figure 7, for Adult and NLTCS data, when ε was 1 and the number of attributes m was 5 and 10, respectively, the MSE of MDAV, IDP_KMENAS, and MDAV_DP gradually decreased with the increase of the number of clustering a, while the MSE of the SSKM_DP algorithm remained stable all the time. The MSE of SSKM_DP was always smaller than the MSE of the other three algorithms. This is because MDAV, IDP_KMENAS, and MDAV_DP are very sensitive to the number of clustering a, resulting in uneven clustering results and considerable information loss, while SSKM_DP is not affected by the number of clustering a and has a good anti-noise ability.

It can be clearly concluded from the experimental results that the published data generated by the SSKM_DP algorithm are obviously better than MDAV, IDP_KMENAS, and MDAV_DP in terms of data utility when the privacy protection degree is certain.

7. Conclusions

In this paper, the balance between data utility and privacy protection of multi-sensitive attribute data was studied, and a differential privacy data publishing method based on the SFLA-Kohonen network was proposed. Our proposed model noisily processes the dataset so that it satisfies the differential privacy of privacy protection, which inevitably affects the availability of data, but as the privacy budget is set, with the better the data availability, there is a corresponding decrease in security. A common approach to differential privacy data publishing adds noise to each piece of data, introducing excessive noise and reducing availability. Most previous clustering algorithms need to be improved to achieve better clustering results. For example, k-means clustering, while effective, is limited by the need to artificially set initial k-values. Although the DBSCAN algorithm does not need to set the

number of clusters and it is highly robust, it is not suitable for high-dimensional data. To solve this problem, we introduced the SFLA algorithm to improve the Kohonen network, obtain the insensitive attributes associated with the sensitive attributes through MIC, and add the noise required to satisfy differential privacy to ensure that the data privacy was not leaked. We theoretically proved that the SSKM_DP algorithm improves the availability of published data while satisfying the differential privacy. Finally, the experimental results on real data proved that the performance of the SSKM_DP algorithm is significantly better than other similar methods. Under the premise of meeting the same privacy requirements, the availability of the data to be published by the SSKM_DP algorithm was better. With different sensitivity degrees of attributes, adding noise is not the same. Directly adding the same size of noise is bound to lead to part of the release of data privacy protection as insufficient. Part of the release of data privacy protection is excessive, resulting in the waste of privacy resources and the lack of data information, reducing the utility of the problem of data. In the next research work, we must not only design a more reasonable privacy budget allocation strategy and further improve privacy protection capabilities and data utility, but we must also consider the future in the distributed environment and the security and usability of the algorithm in this paper.

Author Contributions: Methodology, J.H.; Validation, Z.C.; Formal analysis, X.Z.; Investigation, Q.W.; Resources, N.Z.; Writing—original draft, Z.C.; Visualization, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported in part by Applied Basic Research Project of Liaoning Province under Grant 2022JH2/101300280, Scientific Research Fund Project of Education Department of Liaoning Province under Grant LJKZ0625.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiao, Y.; Li, H. Privacy Preserving Data Publishing for Multiple Sensitive Attributes Based on Security Level. *Information* **2020**, *11*, 166. [CrossRef]
2. Chen, Y.; Xu, Z.; Chen, J.; Jia, S. B-DP: Dynamic Collection and Publishing of Continuous Check-In Data with Best-Effort Differential Privacy. *Entropy* **2022**, *24*, 404. [CrossRef]
3. Yan, Y.; Sun, Z.; Mahmood, A.; Xu, F.; Dong, Z.; Sheng, Q.Z. Achieving Differential Privacy Publishing of Location-Based Statistical Data Using Grid Clustering. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 404. [CrossRef]
4. Zhang, X.; Luo, Y.; Yu, Q.; Xu, L.; Lu, Z. Privacy-Preserving Method for Trajectory Data Publication Based on Local Preferential Anonymity. *Information* **2023**, *14*, 157. [CrossRef]
5. Utaliyeva, A.; Shin, J.; Choi, Y.-H. Task-Specific Adaptive Differential Privacy Method for Structured Data. *Sensors* **2023**, *23*, 1980. [CrossRef]
6. Zhuo, M.; Huang, W.; Liu, L.; Zhou, S.; Tian, Z. A High-Utility Differentially Private Mechanism for Space Information Networks. *Remote Sens.* **2022**, *14*, 5844. [CrossRef]
7. Soria-Comas, J.; Domingo-Ferrer, J.; Sanchez, D.; Martínez, S. Enhancing Data Utility in Differential Privacy via Microaggregation-based K-anonymity. *VLDB J.* **2014**, *23*, 771–794. [CrossRef]
8. Sweeney, L. k-ANONYMITY: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
9. Zhao, X.W.; Liang, J.Y. An Attribute Weighted Clustering Algorithm for Mixed Data Based on Information Entropy. *J. Comput. Res. Dev.* **2016**, *53*, 1018–1028.
10. Sanchez, D.; Domingo-Ferrer, J.; Martinez, S.; Soria-Comas, J. Utility-Preserving Differentially Private Data Releases via Individual Ranking Micro Aggregation. *Inf. Fusion* **2016**, *30*, 1–14. [CrossRef]
11. Monedero, D.R.; Mezher, A.M.; Colome, X.C.; Forné, J.; Soriano, M. Efficient K-anonymous Micro Aggregation of Multivariate Numerical Data via Principal Component Analysis. *Inf. Sci.* **2019**, *503*, 417–443. [CrossRef]
12. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. L-diversity: Privacy beyond K-anonymity. *ACM Trans. Knowl. Discov. Data* **2006**, *1*, 3–5. [CrossRef]
13. Li, Y.; Zhou, F.; Xu, Z. Privacy protection scheme for mobile social networks supporting k-nearest neighbor search. *J. Comput. Sci.* **2021**, *44*, 1481–1500.

14. Parra-Arnau, J.; Domingo-Ferrer, J.; Soria-Comas, J. Differentially private data publishing via cross-moment microaggregation. *Inf. Fusion* **2020**, *53*, 269–288. [CrossRef]
15. Gu, Z.; Zhang, G.; Ma, C.; Song, L. Differential privacy data publishing method based on probabilistic principal component analysis. *J. Harbin Eng. Univ.* **2021**, 1–8. Available online: https://kns-cnki-net.wvpn.lnut.edu.cn/kcms/detail/23.1390.U.20210609.1219.004.html (accessed on 10 August 2021).
16. Chen, S.; Fu, A.; Ke, H.; Su, C.; Sun, H. MCDP: Multi cluster distributed differential privacy data publishing method based on neural network. *Acta Electron. Sin.* **2020**, *48*, 2297–2303.
17. Ye, Y.; Wang, L.; Han, J.; Qiu, S.; Luo, F. An Anonymization Method Combining Anatomy and Permutation for Protecting Pprivacy in Microdata with Multiple Sensitive Attributes. In Proceedings of the 2017 International Conference on Machine Learning and Cybernetics, Ningbo, China, 9–12 July 2017; pp. 404–411.
18. Saraswathi, S.; Thirukumar, K. Enhancing Utility and Privacy Using T-closeness for Multiple Sensitive Attributes. *Adv. Nat. Appl. Sci.* **2016**, *10*, 6–14.
19. Li, N.; Li, T.; Venkatasubramanian, S. t-Closeness: Privacy beyond k-Anonymity and l-Diversit. In Proceedings of the IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115.
20. Acs, G.; Melis, L.; Castelluccia, C.; De Cristofaro, E. Differentially Private Mixture of Generative Neural Networks. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 1109–1121. [CrossRef]
21. Wang, H.; Ge, L.N.; Wang, S.Q.; Wang, L.; Zhang, Y.; Liang, J. Improvement of Differential Privacy Protection Algorithm Based on Optics Clustering. *J. Comput. Appl.* **2018**, *38*, 73–78. (In Chinese) [CrossRef]
22. Yao, S. An Improved Differential Privacy K-Means Algorithm Based on MapReduce. In Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design, Hangzhou, China, 8–9 December 2018; pp. 141–145.
23. Soria-Comas, J.; Domingo-Ferrer, J. Differentially Private Data Publishing via Optimal Univariate Micro-aggregation and Record perturbation. *Knowl.-Based Syst.* **2018**, *153*, 78–90. [CrossRef]
24. Dwork, C. Differential Privacy. In Proceedings of the 33rd International Colloquium on Automata Languages and Programming, Venice, Italy, 10–14 July 2006; pp. 1–12.
25. Ji, Z.; Lipton, Z.C.; Elkan, C. Differential Privacy and Machine Learning: A Survey and Review. *arXiv* **2014**, arXiv:1412.7584.
26. Onishi, A. Landmark Map: An Extension of the Self-organizing Map for a User-intended Nonlinear Projection. *Neurocomputing* **2020**, *388*, 228–245. [CrossRef]
27. Eusuff, M.M.; Lansey, K.E. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. *J. Water Resour. Plan. Manag.* **2003**, *129*, 210–225. [CrossRef]
28. Eusuff, M.; Lanmy, K.; Pasha, F. Shuffled Frog-leaping Algorithm: A Memetic Meta-heuristic for Discrete Optimization. *Eng. Optim.* **2006**, *38*, 129–154. [CrossRef]
29. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
30. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; McVean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting Novel Associations in Large datas. *Science* **2011**, *334*, 1518–1524. [CrossRef]
31. Ye, Q.Q.; Meng, X.F.; Zhu, M.J.; Huo, Z. Survey on Local Differential Privacy. *J. Softw.* **2018**, *29*, 1981–2005. (In Chinese)
32. Bai, L.; Liang, J.; Cao, F. A Multiple K-means Clustering Ensemble Algorithm to Find Nonlinearly Separable Clusters. *Inf. Fusion* **2020**, *61*, 36–47. [CrossRef]
33. Scitovski, R.; Sabo, K. DBSCAN-like Clustering Method for Various Data Densities. *Pattern Anal. Appl.* **2019**, *23*, 541–554. [CrossRef]