future internet

MDPI

Article

Differential Privacy Data Publication Based on Scoring Function

Ke Yuan [1,2], Quan Zhang [1], Yinghao Lin [1,3], Yuye Wang [1,*] and Chunfu Jia [1,4]

[1] School of Computer and Information Engineering, Henan University, Kaifeng 475004, China; yuanke@henu.edu.cn (K.Y.); zhangquan@henu.edu.cn (Q.Z.); linyh@henu.edu.cn (Y.L.)
[2] Henan Provincial Engineering Research Center of Spatial Information Processing, Henan University, Kaifeng 475004, China
[3] Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475001, China
[4] College of Cryptology and Cyber Science, Nankai University, Tianjin 300350, China
* Correspondence: wangyuye@henu.edu.cn

Abstract

Existing Bayesian network-based differential privacy algorithms predominantly employ uniform privacy budget allocation. However, since attribute nodes carry heterogeneous information loads, the traditional privacy budget allocation strategy may result in insufficient noise being added to important attributes, while excessive noise is added to less important attributes. To optimize privacy budget utilization, we propose SA-PrivBayes, a scoring-function-driven allocation method. To enhance Bayesian network precision, we introduce a threshold mechanism during network construction that pre-filters low-scoring attribute pairs before applying the exponential mechanism for selection. Subsequently, during parameter learning, privacy budgets are dynamically allocated to low-dimensional attribute sets based on node-specific scoring functions. Under identical privacy budgets, our algorithm demonstrates stronger data protection capabilities compared to the PrivBayes algorithm. Experimental results indicate that, compared to traditional differential privacy methods based on Bayesian networks under identical privacy budgets, our algorithm better meets the privacy protection requirements of high-dimensional data while maintaining higher data utility.

Keywords: Bayesian networks; differential privacy; scoring functions; data publishing

1. Introduction

In recent years, as data-driven technologies have advanced rapidly, global data volumes have grown exponentially, accompanied by a surge in sensitive information, including personal data, transactional records, and behavioral datasets. While these data hold immense commercial value, they pose unprecedented challenges for privacy protection. Ensuring uncompromised user privacy throughout data collection, storage, analysis, and publication has become a pressing concern, making privacy protection a critical issue in information security that demands innovative solutions to balance data utility and safeguards against leakage [1].

Private data leakage has accelerated the development of privacy protection. Unlike traditional data anonymization methods (k-anonymity [2], l-diversity [3], t-closeness [4], (α, k)-anonymity [5]) that fail to defend against all background knowledge attacks, differential privacy [6,7] has gained widespread attention due to its rigorous mathematical foundations. Adding noise to data prevents accurate identification of individual information and has been widely applied in data analysis and publication as an ideal privacy protection solution.

Check for updates

Academic Editors: Carlo Blundo and Gianluigi Ferrari

Received: 7 November 2025
Revised: 8 February 2026
Accepted: 13 February 2026
Published: 15 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Scholarly definitions of high-dimensional data vary: Narayanan et al. [8] define it as record-type data with multiple attributes, Wang et al. [9] require at least two dimensions, and Amaratunga [10] provides a rigorous characterization—dozens to hundreds of features with intricate structural correlations and rich semantic information—forming the conceptual basis of this study. Traditional differential privacy mechanisms face inherent limitations in processing such data: Sweeney's research [2] shows that zip code, birth date, and gender can identify 87% of U.S. citizens, demonstrating the impracticality of rigid privacy/public attribute categorization. Direct application of low-dimensional frameworks to high-dimensional data leads to two core challenges [11,12]: (1) Noise Amplification Paradox: Exponential attribute growth requires geometrically increased noise, inversely correlating privacy protection and data utility; (2) Utility Optimization Impasse: Lack of sophisticated mechanisms to balance information fidelity and privacy risks, especially for feature spaces exceeding hundreds of dimensions.

To address these issues, we propose SA-PrivBayes, an improved differential privacy Bayesian network algorithm for data generation. Firstly, we adopt a mutual information-based structure-learning method to address structural variability arising from random initial node selection during Bayesian network construction. Secondly, we implement a threshold-based strategy to filter out low-scoring attribute pairs, mitigating the exponential mechanism's tendency to select them and thereby improving data quality and network accuracy. Finally, we design a dynamic privacy budget allocation mechanism that ranks attribute node importance using scoring functions to support rational budget distribution. Experimental results show that, compared to traditional Bayesian network-based differential privacy methods, our approach better meets the requirements for publishing high-dimensional data while ensuring adequate privacy protection, reducing computational complexity, and providing new insights for privacy-preserving data publication.

2. Related Works

Existing differential privacy algorithms for high-dimensional data publication include methods based on probabilistic graphical models [13], feature dimensionality reduction [14,15], tree models [16,17], rough sets [18], threshold filtering [19], and projection techniques [20]. These methods achieve dimensionality reduction to mitigate noise's impact on utility, but suffer from limitations: feature dimensionality reduction may fail under non-Gaussian distributions; tree models require substantial resources and are prone to overfitting; rough set methods struggle with continuous data; and threshold filtering and projection often overlook inter-attribute dependencies, leading to suboptimal privacy or utility.

For high-dimensional data publication, researchers primarily use probabilistic graphical models (e.g., Bayesian networks [21], Markov networks [22]) to model attribute correlations, as non-graphical methods [23–25] suffer from computational inefficiency or poor practical performance. Among graphical model-based methods, PrivBayes [13] is representative, but its excessive number of subnetworks leads to uneven model fitting, affecting synthetic data quality and privacy protection. Subsequent improvements address PrivBayes' shortcomings but have their own limitations: DPSynthesizer [26] adds synthesis-phase privacy safeguards but struggles with non-numerical data and incurs high computational overhead; AprivBayes [27] uses multiple initial nodes and FNC-Bayes to enhance privacy and accuracy but is computationally intensive for large-scale datasets; PrivSyn [28] relaxes conditional independence assumptions to capture correlations but may omit attribute dependencies, reducing utility and privacy; Liu et al.'s incremental learning method [29] prunes weak edges to improve accuracy but has high computational complexity; Ni et al.'s normalized information entropy-optimized Bayesian network [30] achieves effective pri-

vacy protection for large-scale data; Li et al.'s method [31] generates high-quality synthetic financial data using traditional differentially private Bayesian network structure learning; ACDP-Tree [32] uses attribute correlation classification trees and leaf node Laplace noise for privacy preservation; PrivBC [33] clusters correlated attributes and uses a relational matrix to reduce candidate space, improving efficiency; PrivSCBN [34] leverages spectral clustering for high-dimensional binary data and allocates distinct sub-algorithm budgets to enhance precision; PrivASG [35] uses attribute sensitivity levels to enhance sensitive data protection but may cause privacy leakage or reduced utility; LoHDP [36] adopts adaptive marginal computation and attribute clustering to resolve marginal calculation issues in local differential privacy; Shi et al.'s method [37] uses rough set theory and frequent itemsets for distributed high-dimensional data publication, protecting privacy via the exponential mechanism and Laplace noise; Chen et al.'s edge-based method [38] combines UMAP and LSTM dynamic time windows for efficient data aggregation with privacy guarantees. Notably, these clustering or probabilistic graphical model-based methods typically adopt uniform privacy budget allocation, failing to distinguish between important and non-important attributes. This leads to insufficient protection for critical data or excessively low utility for non-important data, making rational privacy budget allocation an unresolved challenge in high-dimensional data privacy protection.

3. Preliminaries

This section introduces the fundamental concepts of differential privacy and Bayesian networks.

3.1. Differential Privacy

Differential privacy is a privacy-preserving model with rigorous mathematical proofs. It employs randomized noise on datasets to ensure that changes to individual records cannot be identified by attackers, while preserving the accuracy of aggregate data queries, thereby achieving the goal of privacy protection.

Definition 1. *ε-differential privacy [6]. D1 and D2 are any two adjacent data set; that is, they differ by only one record. For any possible output Ω, there is a random algorithm A that satisfies*

$$\frac{pr(A(D_1) \in \Omega)}{pr(A(D_2) \in \Omega)} \leq \exp(\varepsilon) \tag{1}$$

A randomized algorithm A is said to satisfy ε-differential privacy. Here, ε represents the privacy budget, which is inversely related to the strength of privacy protection. A smaller value of ε indicates a stronger privacy guarantee.

Definition 2. *Laplace mechanism [7]. For any dataset D and query function f,*

$$M(D) = f(D) + Lap\left(\frac{\Delta f}{\varepsilon}\right) \tag{2}$$

The algorithm M satisfies ε-differential privacy. Here, Δf is the global sensitivity of the query function f, and $Lap\left(\frac{\Delta f}{\varepsilon}\right)$ is the Laplace noise added to the dataset. From the formula, the amount of noise added is directly proportional to the global sensitivity of the query function f and inversely proportional to the privacy budget.

https://doi.org/10.3390/fi18020103

Definition 3. *Mechanism definition [39]. For any dataset D and a query function $u(D, r)$, let Ω denote the output range, and let r denote the output space. The mechanism M satisfies if*

$$M(D, u) = \{r :| Pr[r \in \Omega] \propto \exp\left(\frac{\varepsilon u(D, r)}{2\Delta u}\right)\} \tag{3}$$

The algorithm M satisfies ε-differential privacy. Among them, the output value of r is determined by the decision function $u(D, r)$, where higher values of r are chosen to output with greater probability.

To prove that the algorithm satisfies differential privacy, we need to use two composition properties.

Property 1. *Sequential composition property [40]. Given a dataset D and a sequence of independent randomized algorithms $A_1(D), A_2(D), \ldots, A_m(D)$ each satisfying ε_i-differential privacy on D, the combined algorithm A satisfies $\sum_{i=1}^{m} \varepsilon_i$-differential privacy.*

Property 2. *Parallel composition property [40]. Given a dataset D, partitioned into m disjoint subsets $D = \{D_1, D_2, \ldots, D_m\}$, and independent randomized algorithms $A_1(D_1), A_2(D_2), \ldots, A_m(D_m)$ each satisfying ε_i-differential privacy on their respective subsets, then the combined algorithm satisfies $\max(\varepsilon_i)$-differential privacy.*

Definition 4. *Mutual information function. In 1948, Shannon introduced the concept of information entropy. The mutual information I between attributes represents the degree of association between them. For high-dimensional dataset D, the mutual information between attribute nodes X and Y is given by Formula (4), calculated as follows:*

$$I(X : Y) = H(X) + H(Y) - H(X, Y) \tag{4}$$

3.2. Bayesian Networks

A Bayesian network N is a directed acyclic graph that can be used to describe the relationships between nodes, thereby better understanding the attributes and interdependencies among them. It mainly consists of three parts: (X, A, Θ), where (X, A) is the set of nodes in the network, A represents the directed edges in the network, and Θ denotes the network parameters. Bayesian networks effectively express independence relationships among attributes. Therefore, the joint probability of all nodes represented by the Bayesian network can be described as the product of the conditional probabilities of nodes $X_1, X_2, \ldots, X_n$. According to the conditional independence assumption, the Bayesian network factorizes the joint probability P in detail, resulting in the following formula:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i)) \tag{5}$$

where n is the number of nodes, X_i is the i-th node, and $Pa(X_i)$ is the parent set of node X_i.

Definition 5. *A Bayesian network can be represented by a set of attribute nodes and their parent node sets, where the attribute nodes are $\{(W_1, \Pi_1), (W_2, \Pi_2), \ldots, (W_i, \Pi_i)\}$.*

From this definition, the following information can be derived:

- *W_i $(i \in d)$ is an attribute node in the attribute set.*
- *Π_i $(i \in d)$ is the set of parent nodes of attribute W_i.*

Definition 6. *Importance. In a Bayesian network N containing d attribute nodes, the scoring function assigns a numerical value to each attribute node to represent its importance in the published dataset. During the construction of the Bayesian network, the score function value of each node is used to measure its importance. The higher the score function value, the greater the node's importance; conversely, the lower the score function value, the lower the node's importance.*

Definition 7. *Correlation matrix. Given a dataset D, $A = \{A_1, A_2, \ldots, A_d\}$ as the set of attributes of dataset D, the mutual information between the attributes is calculated, and the correlation matrix is obtained as in Equation (6). MIM is a fully symmetrical matrix, as the mutual information between a variable and itself is meaningless, so the diagonal elements are set to zero.*

$$MIM(A) = \begin{pmatrix} 0 & I_{A_1,A_2} & I_{A_1,A_3} & \cdots & I_{A_1,A_d} \\ I_{A_2,A_1} & 0 & I_{A_2,A_3} & \cdots & I_{A_2,A_d} \\ I_{A_3,A_1} & I_{A_3,A_2} & 0 & \cdots & I_{A_3,A_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_{A_d,A_1} & I_{A_d,A_2} & I_{A_d,A_3} & \cdots & 0 \end{pmatrix} \tag{6}$$

Definition 8. *Average mutual information. Given a dataset D, $A = \{A_1, A_2, \ldots, A_n\}$ as the set of attributes of dataset D, the average mutual information between one variable and the others is calculated as follows. The metric reflects the average association between a variable and other variables. To ensure dimensional consistency, the denominator is set to $(d-1)$.*

$$AMI(A_i) = \frac{1}{d-1} \sum_{j=1}^{d} (MIM)_{i,j} \tag{7}$$

4. Score Function-Based SA-PrivBayes Method

A critical issue in algorithms for publishing high-dimensional data privacy is the trade-off between dimensionality reduction and differential privacy protection. The PrivBayes method, which leverages Bayesian models, effectively reduces dimensionality for high-dimensional datasets. However, existing research has revealed significant flaws in the Bayesian network construction process, such as random selection of the initial node and insufficient information selection strategies. During Bayesian network construction, the choice of the initial node and the order of node addition are crucial to the quality of the resulting network. To address this, we introduce average mutual information. Meanwhile, existing data publication methods focus on correlations between attributes and uniformly allocate privacy budgets across all attribute nodes, failing to account for differences in information content among attributes. This leads to an irrational allocation of privacy budget during attribute protection. To address these shortcomings, this paper proposes a high-dimensional data publication framework for differential privacy based on a scoring function. The method's flowchart is shown in Figure 1.

To comprehensively evaluate SA-PrivBayes, we compare it with current mainstream privacy protection algorithms. As shown in Table 1.

From the comparison results, it can be seen that SA-PrivBayes has significant advantages in terms of network structure learning and privacy budget allocation, effectively addressing the limitations of existing algorithms.

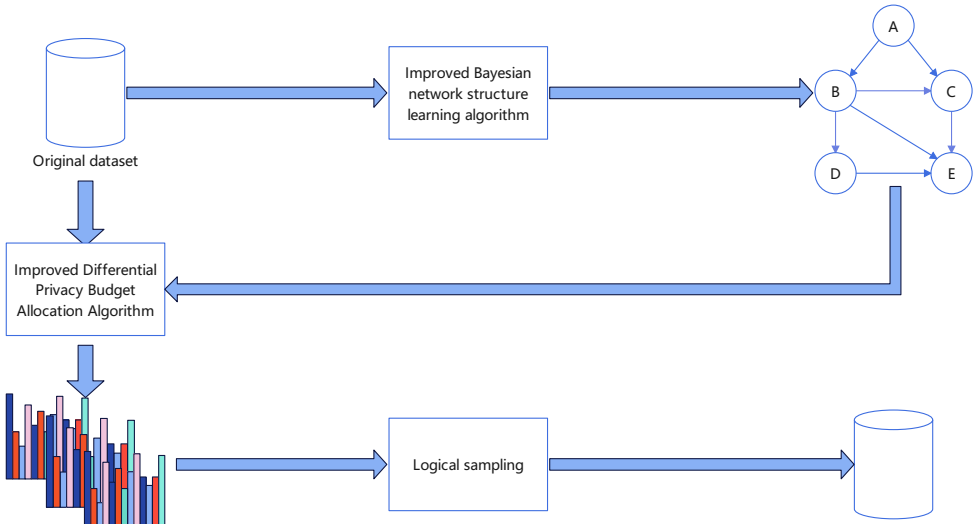


Figure 1. Flowchart of high-dimensional data publication under differential privacy protection

Table 1. Comparison of SA-PrivBayes with existing algorithms.

Algorithm	Key Limitations	Improvements & Advantages of SA-PrivBayes
PrivBayes [13]	Insufficient protection for core attributes; low-value AP pairs interfere with network accuracy	1. AMI-based root node selection to strengthen core correlations; 2. Threshold filtering for low-value AP pairs; 3. Cluster-based budget allocation to enhance core attribute protection
ELPrivBayes [41]	Irrational budget allocation	1. Threshold + probabilistic screening to adapt to noise interference; 2. Collaborative budget allocation to balance privacy and utility
AprivBayes [27]	High computational complexity, incompatible with large-scale data	1. Attribute importance-based allocation superior to sub-network equal division; 2. Supports large-scale datasets (e.g., Adult with 41,292 records)
DPSynthesizer [26]	High computational overhead	1. Full-process collaborative optimization of network structure without additional noise; 2. Time complexity $O(nd^2)$ for higher efficiency
PrivASG [35]	Sensitivity not combined with network structure, poor adaptability	1. Scoring function integrating network structure and attribute correlation strength; 2. Cluster + dynamic q-value adjustment for multi-scenario adaptation; 3. Stronger cross-domain generality
ACDP-Tree [32]	Adopts an "equal division + arithmetic progression adjustment" strategy and fails to dynamically optimize based on attribute correlation	1. Threshold filtering for low-value AP pairs; 2. Captures correlations among all attributes comprehensively based on Bayesian networks

4.1. SA-PrivBayes Algorithm Bayesian Network Construction

In research on Bayesian network structure modeling, most methods rely on the greedy approach used in the GreedyBayes algorithm. In the GreedyBayes algorithm, an attribute node is randomly selected as the initial node of the Bayesian network. Then, by enumerating the mutual information between all attributes, the attribute node with the highest mutual information is sequentially added to the network, eventually completing the construction of the Bayesian network. Previous studies have shown that the initial node in the GreedyBayes algorithm is selected randomly, potentially leading to the selection of a node with no

strong dependencies on other nodes. This can prevent the Bayesian network from fully utilizing the valuable information in the data, reducing the network's ability to capture mutual information and thereby affecting how well the approximate distribution $Pr_N[A]$ approximates $Pr[A]$. In the SA-PrivBayes algorithm, we select the attribute node with the highest average mutual information as the initial node, and determine the order in which nodes enter the Bayesian network based on the size of the average mutual information, thereby enhancing the rationality of the network structure. However, the exponential mechanism still uses small mutual information to sample AP pairs. In this case, the exponential mechanism tends to select candidate AP pairs with very low mutual information. As the privacy budget decreases, this probability increases, leading to very low-quality output. To address this, we set a threshold $\lambda(\lambda < 1)$ when selecting AP pairs through the exponential mechanism. If the mutual information function of an attribute pair is below the λ-th position in the sorted list of all attribute pairs' mutual information, it will be filtered out. The exponential mechanism is then applied to select from the remaining attribute pairs in the candidate space.

The structure learning algorithm of SA-PrivBayes is shown in Algorithm 1.

Algorithm 1	**Construction of bayesian network of SA-PrivBayes**
Input:	Dataset D, maximum number of nodes for bayesian network k.
Output:	Bayes network N.
1.	$N = \emptyset, \mathcal{A} = \emptyset, Scores = \emptyset$;
2.	Calculate the correlation matrix MIM based on Equation (6);
3.	Based on Equation (7), calculate the average mutual information for all attributes in $\mathcal{A}$, and sort the attributes by AMI in descending order;
4.	Add $(A_1, \emptyset)$ to N;
5.	Split total privacy budget ε_1: ε_τ (for threshold screening mechanism), $\varepsilon_{\exp}$ (for exponential mechanism);
6.	For $i \leftarrow 2$ to d do:
7.	$\Omega = \emptyset$;
8.	for each $\Pi \in \left(\binom{\{A_j\}_1^{i-1}}{k}\right)$, add (A_i, Π_i) to set Ω;
9.	Calculate the score of each candidate, and sort; τ = the value corresponding to the first λ positions in the sorted list; $\tau = \tau_0 + \mathrm{Lap}\left(\frac{\Delta_f}{\varepsilon_\tau}\right)$;
10.	For retry in range($\lvert\Omega\rvert$):
11.	lower_bound $= \tau - \frac{\Delta_f \cdot \ln(1/\alpha)}{\varepsilon_\tau}$; upper_bound $= \tau + \frac{\Delta_f \cdot \ln(1/\alpha)}{\varepsilon_\tau}$;
12.	If the score of the selected AP is more than upper_bound:
13.	Skip;
14.	else if the score of the selected AP is less than lower_bound:
15.	Remove (A_i, Π_i) from the set Ω;
16.	else:
17.	If Bernoulli(α) == 1:
18.	Skip;
19.	From MIM, examine the mutual information for each (A_i, Π_i), and use mutual information as a scoring function, the exponential mechanism is employed to select (A_i, Π_i) in Ω;
20.	Add (A_i, Π_i) to N, Add $Score_{(A_i, \Pi_i)}$ to Scores;
21.	End for;
22.	Return N, $Scores$.

The input of Algorithm 1 includes the original dataset D, the maximum number of parent nodes k in the Bayesian network, and the total privacy budget ε. Line 1 performs initialization operations. Line 2 calculates the mutual information matrix MIM based on Formula (6). Line 3 calculates the average mutual information (AMI) of all attributes in $\mathcal{A}$ according to Formula (7), and sorts the attributes in descending order of AMI values. Line 4 adds $(A_1, \emptyset)$ to N, selecting the first attribute A_1 in the sorted list as the starting node of the network, with its parent set being empty. Line 5 completes the fine-grained allocation of the total privacy budget ε. Line 6 initiates an iterative loop that processes the second attribute through the d-th attribute. The iteration order strictly follows the AMI ranking of attributes, ensuring that strongly correlated attributes are processed first, thereby improving the efficiency of network construction. Line 7 initializes the set of candidate attribute–parent pairs Ω for the current attribute to an empty set. Line 8 generates all candidate parent combinations for the current attribute A_i. Line 9 completes the scoring

calculation and privacy-preserving threshold generation for candidate AP pairs: first, using mutual information as the scoring function, it calculates the scores for all candidates in Ω and sorts them in descending order; then, it selects the scores of the top λ candidates as the original threshold τ_0; finally, it adds Laplace noise $\mathrm{Lap}(\Delta_f/\varepsilon_1)$ to τ_0 to obtain the noisy screening threshold τ. This operation strictly guarantees ε_1-differential privacy in the threshold screening phase through the Laplace mechanism. Line 10 begins the screening loop for candidate AP pairs. Line 11 calculates the lower and upper bounds of the threshold: the lower bound lower_bound and upper bound upper_bound are computed from the noisy threshold τ and a correction term $(\Delta_f \cdot \ln(1/\alpha))/\varepsilon_1$, where α is the probability parameter for Bernoulli sampling. The purpose of the correction term is to balance the strictness of screening with the misjudgment rate, avoiding screening distortion caused by noise. Lines 12–13 determine whether the candidate AP pair's score exceeds the upper bound; if so, the candidate is retained. Lines 14–15 determine whether the candidate's score is below the lower bound: if it is, the candidate is removed from Ω. Such candidates represent weak attribute associations and contribute minimally to network construction; removing them significantly reduces the computational load of the subsequent exponential mechanism, improving algorithm efficiency. Lines 16–18 handle candidates whose scores lie between the lower and upper bounds: for these candidates, Bernoulli sampling is performed, randomly skipping some candidates with probability α. This probabilistic operation further obfuscates the distribution of the candidate set, enhancing the differential privacy protection effect without compromising the overall rationality of node selection. Line 19 uses the exponential mechanism to select the optimal AP pair from the screened Ω, allocating the total budget ε_2 of the exponential mechanism equally across $d-1$ iterations, with a single iteration budget of $\varepsilon_2/(d-1)$. This ensures that the total privacy budget for the node selection process does not exceed ε_2, strictly satisfying differential privacy requirements. Line 20 adds the selected AP pair (A_i, Π_i) to the network structure N, while storing its mutual information score in the *Scores* set, completing the parent node selection and network update for the current attribute. Line 21 concludes the current attribute's iterative loop and proceeds to the following attribute. Line 22 returns the final Bayesian network structure N and the score set *Scores*, providing essential input for the conditional probability distribution perturbation phase in the subsequent SA-PrivBayes algorithm.

4.2. Differentially Private Conditional Distribution Generation in the SA-PrivBayes Algorithm

To approximate the original data distribution, when publishing high-dimensional data models using Bayesian networks, sampling and synthesis can be performed based on the joint probability distribution of each attribute. However, directly sampling from the joint probability distribution of attributes may lead to privacy leakage, necessitating perturbation to protect privacy. To enhance the usability of published data while ensuring privacy protection, differences in the information content of attributes are taken into account. In prior studies, privacy budgets are typically evenly allocated under reduced privacy budget constraints to perturb the joint attribute distribution. However, this uniform allocation method may result in insufficient privacy protection for critical attributes, leading to privacy breaches. In contrast, overprotection of less important attributes significantly reduces data usability and ultimately impacts data utility. Thus, current privacy budget allocation methods still suffer from suboptimal data utility. Since the importance of attribute nodes varies in high-dimensional data, their protection levels should also differ. Our algorithm uses the scoring function of each attribute node to partition them into m attribute clusters. The literature [33] proposes an improved attribute clustering algorithm, MACA, which divides attributes into clusters based on their dependency relationships. Experiments on public high-dimensional datasets such as NLTCS [42], Adult [43], and ACS [44] show that

the optimal number of clusters is 3. Additionally, attribute information can generally be categorized as high, moderate, or low. Therefore, this paper sets m, the number of attribute clusters, to 3.

4.2.1. Dynamic Privacy Budget Allocation

In a Bayesian network N, suppose the evaluation function value (mutual information function) for each attribute node has been calculated. All attribute nodes are divided into three clusters C_1, C_2, C_3 according to the evaluation function values, where C_1 contains the attribute nodes with the smallest evaluation function values, C_2 contains the attribute nodes with moderate evaluation function values, and C_3 contains the attribute nodes with the largest evaluation function values. Assuming that the ratio parameter q satisfies $q > 1$, starting from the cluster with the lowest importance C_1, the attribute nodes in this cluster and the next cluster C_2 are assigned privacy budgets proportional to the ratio q. Let the total privacy budget be ε, divided among three privacy budgets for the clusters. Each cluster C_i is assigned a privacy budget ε_i, where $i \in \{1, 2, 3\}$, and $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = \varepsilon$. Within each cluster, the privacy budget is allocated to all attribute nodes. Suppose cluster C_i has n_i attribute nodes, then each attribute node in cluster C_i receives a privacy budget of ε_i / n_i.

Assume that there are six attribute nodes to allocate the privacy budget. The scoring function for the low-dimensional attribute sets of the attribute nodes is sorted in descending order as A, B, C, D, E, F. The total privacy budget is 0.5, and the scaling constants are set to $q = 1$, $q = 1.1$, and $q = 1.2$. According to the dynamic privacy budget allocation method, the privacy budget allocated to each low-dimensional attribute set is shown in the Table 2.

Table 2. ε Allocation table.

q	A	B	C	D	E	F
1	0.083	0.083	0.083	0.083	0.083	0.083
1.1	0.076	0.076	0.083	0.083	0.091	0.091
1.2	0.069	0.069	0.082	0.082	0.099	0.099

The q-value is a proportional parameter used to allocate privacy budgets. Its core role is to achieve the differential privacy protection goal of "enhancing privacy protection for high-importance clusters while balancing data utility for low-importance clusters" by adjusting the privacy budget for allocation across clusters of different importance. The dynamism of the q-value is reflected in two core aspects: adaptability to dataset characteristics and dynamic matching of privacy requirements.

Cross-dataset dynamic adaptation: The selection of the q-value is personalized based on the inherent characteristics of the dataset, with core reference dimensions including attribute correlation strength, sensitive attribute distribution, and data dimensionality scale. For datasets with dense attribute correlations, the dependency relationships between attribute pairs are more significant, and the privacy protection requirements for core clusters are higher; thus, a larger q-value is chosen to enhance privacy protection for core attributes. For datasets with relatively sparse attribute distributions, the dependencies between attributes are more dispersed, so a smaller q-value can be chosen to improve data utility while ensuring privacy.

Dynamic matching of privacy requirements for the same dataset: For the same dataset, the q-value can be automatically optimized based on user-preset privacy protection levels, achieving demand adaptation for different scenarios:

- High protection level: corresponds to q = 1.2–1.4, suitable for scenarios with higher privacy risks such as data sharing and public release;
- Medium protection level: corresponds to q = 1.0–1.2, suitable for regular scenarios such as internal data analysis and model training;

- Low protection level: corresponds to q = 0.8–1.0, suitable for exploratory analysis of non-sensitive data and utility-priority scenarios.

This design enables the algorithm to adapt to varying privacy requirements by dynamically matching the q-value without manual parameter adjustment, thereby enhancing the method's practicality and flexibility. Its design logic ensures a balance between privacy and utility in different scenarios, guaranteeing the stability and generalizability of the algorithm.

The essence of differential privacy lies in achieving privacy protection by adding noise to data: a smaller ϵ leads to more noise added and a higher level of privacy protection, while a larger ϵ results in less noise and better preservation of the original data utility. Attribute nodes with large scoring function values are the core of information correlation and key carriers of data distribution characteristics in high-dimensional data. They contain more concentrated user privacy information, making them significantly more vulnerable to being exploited by attackers for privacy inference and thus classified as high-privacy-risk nodes. In contrast, attribute nodes with smaller scoring function values have lower correlation with other attributes and exert a weaker impact on the overall data distribution, leading to relatively lower privacy leakage risks and being categorized as low-privacy-risk nodes. We allocate smaller privacy budgets to high-privacy-risk nodes to enhance protection and larger privacy budgets to low-privacy-risk nodes to reduce noise interference. This strategy precisely aligns with the principle that "the higher the privacy risk, the stronger the protection; the lower the privacy risk, the more sufficient the utility preservation," and perfectly conforms to the core design logic of differential privacy mechanisms—"allocating protection resources on demand."

4.2.2. Algorithm Implementation

The specific implementation process of our algorithm is described in Algorithm 2.

Algorithm 2	**Noise addition method for conditional distributions in SA-PrivBayes**
Input:	Dataset D, Bayesian network N, the degree of the Bayesian network k, node evaluation function Scores, Parameter q.
Output:	Differential privacy distribution P^*.
1.	$P^* = \emptyset$;
2.	For $i \leftarrow k+1$ to d do:
3.	Calculate the joint distribution $Pr(X_i, \Pi_i)$;
4.	According to the node scoring function, calculate the privacy budget ϵ_i for the node's distribution, using the updated privacy budget ϵ_i to perturb the distribution: $P^*[X_i, \Pi_i] \leftarrow Pr[X_i, \|\Pi_i] + Lap(2/(n \times \epsilon_i))$;
5.	Set the negative values in $Pr^*[X_i, \Pi_i]$ to 0, and update $P^*[X_i, \Pi_i]$ accordingly;
6.	From $Pr^*[X_i, \Pi_i]$, derive $Pr^*[X_i \| \Pi_i]$, and add the results to P^*;
7.	End for
8.	For $i \leftarrow 1$ to k do:
9.	From $Pr^*[X_{k+1}, \Pi_k]$, derive $Pr^*[X_i, \|\Pi_i]$, and add the results to P^*;
10.	End for
11.	Return P^*;

The input of Algorithm 2 includes the dataset D, the Bayesian network N, the degree k of the Bayesian network (which is a hyperparameter controlling network complexity), the node scoring function Scores, and the parameter q, which is an adjustment coefficient for budget allocation used to optimize the efficiency of privacy budget utilization. The output of the algorithm is the noisy distribution P^*, which includes the conditional distributions $Pr^*[X_i|\Pi_i]$ of all attributes, serving as the core basis for subsequently generating privacy-preserving synthetic data. Line 1 performs initialization by setting the noisy distribution set P^* to empty. Line 2 initiates the first iteration loop, iterating from the $k+1$-th attribute to the d-th attribute. Line 3 computes the joint distribution $Pr(X_i, \Pi_i)$ of the current attribute X_i and its parent node set Π_i. Line 4 is the core step of Algorithm 2, implementing

differential privacy budget allocation and Laplace noise addition: first, the privacy budget ϵ_i allocated to the current attribute node is calculated via the node scoring function Scores—higher-scoring attributes are allocated less budget to ensure the privacy of core attributes; then, based on the adjusted ϵ_i, Laplace noise is added to the joint distribution $Pr(X_i, \Pi_i)$. Line 5 post-processes the noisy joint distribution $Pr^*[X_i, \Pi_i]$: first, negative values in the distribution are set to 0, then the distribution is normalized to ensure that the sum of probabilities for all value combinations equals 1. Line 6 derives the conditional distribution based on the noisy joint distribution, completing the construction of the noisy distribution for the current attribute. Line 7 concludes the first iteration loop, completing the construction of noisy conditional distributions for the $d-k$ attributes. Line 8 initiates the second iteration loop, iterating from the 1st attribute to the k-th attribute, to process the core nodes of the Bayesian network. Line 9 deduces the conditional distribution for the core nodes. Line 10 concludes the second iteration loop, completing the construction of noisy conditional distributions for all attributes. Line 11 returns the final noisy distribution P^*.

4.3. Privacy Security Analysis

Theorem 1. *SA-PrivBayes differential privacy guarantee: Let $\varepsilon_1, \varepsilon_2 > 0$, Algorithm 1 satisfies ε_1-differential privacy, Algorithm 2 satisfies ε_2-differential privacy, and the input of Algorithm 2 depends on the output of Algorithm 1. Then the SA-PrivBayes algorithm satisfies ε-differential privacy, where $\varepsilon = \varepsilon_1 + \varepsilon_2$.*

Proof. We prove the theorem through the following steps: Step 1: Privacy Guarantee of Algorithm 1. The privacy consumption of Algorithm 1 mainly stems from two sub-processes: Laplace noise addition for the threshold τ, and $d-1$ invocations of the exponential mechanism to select Attribute-Parent (AP) pairs. These two sub-processes are denoted as $\mathcal{M}_\tau$ and $\mathcal{M}_{\text{exp},i}$ ($i = 2, 3, \ldots, d$), respectively. Substep 1.1: Privacy Guarantee of the Threshold Filtering Sub-process $\mathcal{M}_\tau$ According to Definition 2, the threshold filtering sub-process $\mathcal{M}_\tau$ satisfies ε_τ-differential privacy. Substep 1.2: Privacy Guarantee of the AP Pair Selection Sub-process via the Exponential Mechanism Algorithm 1 selects AP pairs through $d-1$ invocations of the exponential mechanism. The total privacy budget for the exponential mechanism, ε_{exp}, is evenly allocated among the $d-1$ invocations. According to Definition 3, each invocation of the exponential mechanism satisfies $\varepsilon_{\text{exp}}/(d-1)$-differential privacy. Substep 1.3:Overall Privacy Guarantee of Algorithm 1 The complete execution process of Algorithm 1 can be expressed as: $\mathcal{A}_1(D) = \left(\mathcal{M}_\tau(D), \mathcal{M}_{\text{exp},2}(D), \ldots, \mathcal{M}_{\text{exp},d}(D)\right)$ First, $\mathcal{M}_\tau$ is executed sequentially with the $d-1$ invocations of $\mathcal{M}_{\text{exp},i}$. Second, the $d-1$ invocations of $\mathcal{M}_{\text{exp},i}$ are combined sequentiall; thus, their joint execution satisfies ε_{exp}-differential privacy. Combining with the ε_τ-differential privacy of $\mathcal{M}_\tau$, Algorithm 1 satisfies ε_1-differential privacy according to Property 1, where $\varepsilon_1 = \varepsilon_{\text{exp}} + \varepsilon_\tau$. Step 2: Privacy Guarantee of Algorithm 2. Algorithm 2 adds Laplace noise perturbation to the joint distribution of $d-k$ attributes. Let the noise addition mechanism for the i-th attribute be $\mathcal{N}_i : \mathcal{D} \to \mathcal{P}_i$, where $\mathcal{P}_i$ denotes the perturbed distribution space of the i-th attribute. Each mechanism $\mathcal{N}_i$ adopts the Laplace mechanism, satisfying ε_i-differential privacy, and the total budget satisfies: $\sum_{i=1}^{d-k} \varepsilon_i = \varepsilon_2$ Consider the noise addition process of Algorithm 2: $\mathcal{A}_2(D) = (\mathcal{N}_1(D), \mathcal{N}_2(D), \ldots, \mathcal{N}_{d-k}(D))$ Since independent noise addition operations are performed on disjoint attribute subsets, Algorithm 2 satisfies ε_2-differential privacy according to Property 2. Step 3: Privacy Guarantee of the Overall Algorithm. Since Algorithm 1 satisfies ε_1-differential privacy and Algorithm 2 satisfies ε_2-differential privacy, the SA-PrivBayes algorithm satisfies ε-differential privacy, where $\varepsilon = \varepsilon_1 + \varepsilon_2$. □

4.4. Algorithm Time Complexity Analysis

To demonstrate the algorithm's feasibility, this section analyzes its time complexity. The algorithm's running efficiency mainly depends on two phases: the Bayesian network construction phase and the generation of the noisy joint probability distribution. In generating the noisy joint probability distribution, since our algorithm reallocates privacy budgets only to attribute nodes to ensure reasonable privacy protection for each node, the overhead of privacy budget calculation has a negligible impact on the overall algorithm's efficiency and can be ignored. Therefore, the algorithm's computational cost primarily lies in constructing the Bayesian network. The computational cost of buliding the Bayesian network structure mainly arises from calculating attribute mutual information. The literature [45] proves the time complexity of estimating mutual information once is $O(n)$, and the time complexity of calculating the mutual information of an AP pair is $O(nk)$. The time complexity of GreedyBayes is $O(nkC_{d+1}^{k+2})$, and the literature [41] has already proven in the text that using a mutual information matrix to store node mutual information during the Bayesian network construction phase results in a time complexity of $O(nd^2)$, which is smaller than the time complexity of the GreedyBayes algorithm. Our algorithm uses a matrix to store mutual information between attribute nodes, thereby improving GreedyBayes's runtime efficiency. Compared with the ELPrivBayes algorithm, our algorithm reduces the exponential mechanism's candidate space. When selecting the optimal item, the exponential mechanism typically uses an algorithm to choose the best candidate based on the distribution of candidate scores. By shrinking the candidate space, the complexity of this step is reduced, thereby speeding up the selection of the optimal item. In theory, our algorithm's time efficiency is faster than ELPrivBayes's when handling large-scale datasets.

5. Experimental Evaluation and Result Analysis

This section evaluates the performance of the SA-PrivBayes method in terms of data utility, data security, and its ability to achieve a personalized balance between these requirements. The comparative methods include ELPrivBayes [41], PrivBayes [13], and ACDP-Tree [32]—a differential privacy medical data publishing algorithm based on attribute correlation and classification tree. Experimental comparisons are conducted to assess the data utility and privacy protection capabilities of these algorithms under varying privacy budgets: for SA-PrivBayes, ELPrivBayes, and PrivBayes, comprehensive evaluations are performed covering multiple metrics; for ACDP-Tree, we focus on two core utility metrics aligned with the experimental framework—classification accuracy (to measure the effectiveness of synthetic/published data in supporting predictive tasks) and average variation distance (to quantify the fitting degree of 2D marginal distributions between the generated data and the original data), ensuring fair and targeted comparison.

5.1. Dataset Overview

Three datasets are used in the experiments: the Adult dataset [43], the BR2000 dataset [44], and the NLTCS dataset [42], which are widely used for publishing high-dimensional data. The Adult dataset originates from the 1994 U.S. Census Bureau and contains 41,292 personal records. BR2000 is derived from 38,000 demographic census records collected in Brazil in 2000. The NLTCS dataset comes from the U.S. National Long-Term Care Survey and includes 21,574 records of disability care surveys. Detailed specifications of the experimental datasets are provided in Table 3.

Table 3. Datasets used in the experiments.

Dataset	Data Type	Dataset Size	Dimension
BR2000	Non-Binary	38,000	14
Adult	Non-Binary	41,292	13
NLTCS	Binary	21,574	16

Dataset Preprocessing Pipeline

To ensure the consistency and validity of experimental data, as well as to meet the input requirements of the SA-PrivBayes algorithm and subsequent utility evaluation models, this study implements a standardized preprocessing pipeline for all experimental datasets. This section details the key steps of dataset preprocessing, focusing on encoding methods for different types of categorical variables, since all datasets used in this study are discrete categorical data and no binning operations are involved. The specific processing logic is as follows:

- Binary categorical attributes (NLTCS dataset): 0-1 encoding is adopted to directly map attributes to binary values.
- Multi-class unordered attributes: One-Hot Encoding is used to generate binary vectors whose dimension is equal to the number of categories.
- Multi-class ordered attributes : Label Encoding is applied to map attributes to continuous integers $(1, 2, \ldots, n)$ according to their ordered levels.

5.2. Experimental Setup

The experimental environment in this paper consists of an Intel(R) Core(TM) i5-7500 CPU @ 3.40 GHz, 16.0 GB RAM, Windows 10, and Python 3.9 as the programming language. The Bayesian network structure learning procedure in all experiments adheres to unified parameter settings: the maximum number of parent nodes k is uniformly set to 2; the candidate parent set for each attribute consists of all pre-existing attributes in the network, with no upper bound on the candidate set size other than the constraint imposed by k; the scoring function adopts the mutual information score consistent with the mutual information function defined in Definition 4, where the score of a candidate parent combination equals the total mutual information between the target attribute and the corresponding parent set; a greedy search strategy is employed in the search process, where attributes are added sequentially in descending order of average mutual information, all feasible parent combinations are traversed for each attribute, and the combination achieving the highest score is selected as the final parent set.

This paper primarily evaluates the performance of SA-PrivBayes on three datasets from the following two aspects: (1) the evaluation of the Bayesian network quality during the structure learning phase, with the evaluation metric being the sum of mutual information; (2) the evaluation of the utility of the generated dataset, with the evaluation metric being the performance of the SVM classifier built on the generated dataset. Low-degree Bayesian networks are a core solution for publishing high-dimensional data. Therefore, in the experiments, we selected $k = 2$.

In the experiment, the specific splitting rules of the privacy budget are as follows: the total privacy budget ε is divided into two parts, where ε_1 is used for Bayesian network structure learning in Algorithm 1, and ε_2 is used for conditional probability distribution perturbation and noise addition in Algorithm 2, with a fixed allocation ratio of ε_1:ε_2 = 3:7. The budget is further subdivided within Algorithm 1: the threshold screening stage is allocated $\varepsilon_\tau = 0.3\varepsilon_1$, and the AP pair selection stage of the exponential mechanism is allocated $\varepsilon_{\text{exp}} = 0.7\varepsilon_1$, and ε_{exp} is evenly distributed to the d-1 node iteration processes. The budget allocation within Algorithm 2 follows the following rules: first, the attribute nodes are divided into 3 clusters according to the scoring function, then the total budget

ε_2 is allocated among the clusters through the proportional parameter q, and the budget within each cluster is evenly distributed to each attribute node. The specific allocation logic is completely consistent with the dynamic budget allocation logic described in Section 4.2.1 of this paper. Meanwhile, the number of repeated runs for all experiments is uniformly specified. To eliminate the impact of random fluctuations on the experimental results, all comparative experiments under each privacy budget ε are independently repeated 50 times, and the various experimental indicators presented in the figures and tables, such as classification accuracy and average variation distance, are all the mean values of the 50 repeated runs.

To ensure fairness in comparison, the experimental parameters of ACDP-Tree are kept consistent with the core conditions of SA-PrivBayes, with specific configurations as follows: the Adult, NLTCS and BR2000 datasets are reused, and the attribute division is unified with that of SA-PrivBayes—the "sensitive attributes" of SA-PrivBayes (e.g., income for Adult, eating for NLTCS) are designated as the sensitive attributes of ACDP-Tree, while the remaining attributes serve as quasi-identifiers, with unique identifiers removed; The total privacy budget ε is exactly the same as that of SA-PrivBayes, and split according to ACDP-Tree's rules: 50% allocated to classification tree construction, and 50% to leaf node noise addition; Published data with the same scale as the original dataset is generated, retaining the same attribute set as SA-PrivBayes to ensure comparability in subsequent metric calculations.

5.3. Computational Cost Evaluation

Computational cost is a core metric for evaluating the practicality of high-dimensional data publication algorithms for differential privacy, with the key to optimization lying in balancing the expressiveness of Bayesian networks and computational complexity. Based on the time-complexity analysis and experimental validation results of the SA-PrivBayes algorithm, the degree k of the Bayesian network, a core hyperparameter that controls the network's complexity, plays a decisive role in balancing computational cost and data utility. From the theoretical derivation of time complexity, the computational cost of the SA-PrivBayes algorithm is primarily concentrated in the Bayesian network construction phase, with the core sources of cost being the computation of mutual information between attributes and the screening of attribute-parent pairs. Experimental validation further quantifies the impact of k on computational cost and data utility. In the experiments, different k values were set, and comparative tests were conducted on three public datasets: Adult, BR2000, and NLTCS. The results show that as k increases, the time complexity of the PrivBayes algorithm grows rapidly during Bayesian network construction. In contrast, the ELPrivBayes and SA-PrivBayes algorithms are almost unaffected by this increase. This is because both algorithms construct a mutual information matrix, making the computational cost independent of k. Additionally, our algorithm exhibits slightly lower time complexity than ELPrivBayes, as our approach reduces the candidate space of the exponential mechanism. As shown in Figure 2, our algorithm achieves lower time overhead.

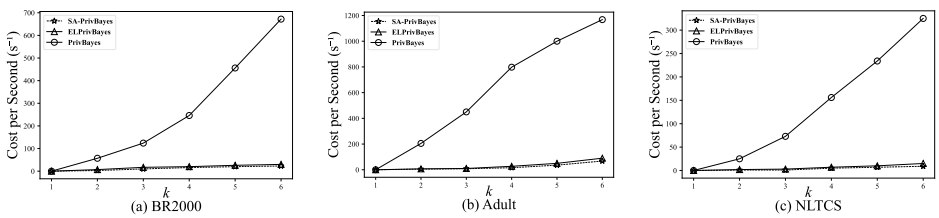


Figure 2. Comparison of computational cost in the Bayesian network construction stage under different k values.

5.4. Bayesian Network Quality Evaluation

For the three datasets mentioned above, this experiment compared the information-capturing capabilities of PrivBayes, ELPrivBayes, and our algorithm, SA-PrivBayes, by constructing Bayesian networks with varying k values. A higher mutual information sum indicates better Bayesian network quality. The goodness-of-fit between the Bayesian network and the original data directly impacts the usability of the published dataset. In this experiment, the sum of mutual information contributed by all attribute-parent pairs in the Bayesian network served as the scoring function, with higher scores indicating a stronger alignment between the Bayesian network and the original data. The experiment selects privacy budget values of 0.02, 0.04, 0.08, 0.2, 0.4, 0.8, 1.6, 3.2, and 6.4. In the Bayesian network construction phase, the ratio of the privacy budget components ε_τ and ε_{exp} is set to 0.3:0.7. The threshold λ of the SA-PrivBayes algorithm is set to 3/4, 2/3, and 1/2, respectively, to compare the accuracy of the Bayesian networks generated by three algorithms. The results are shown in Figure 3.

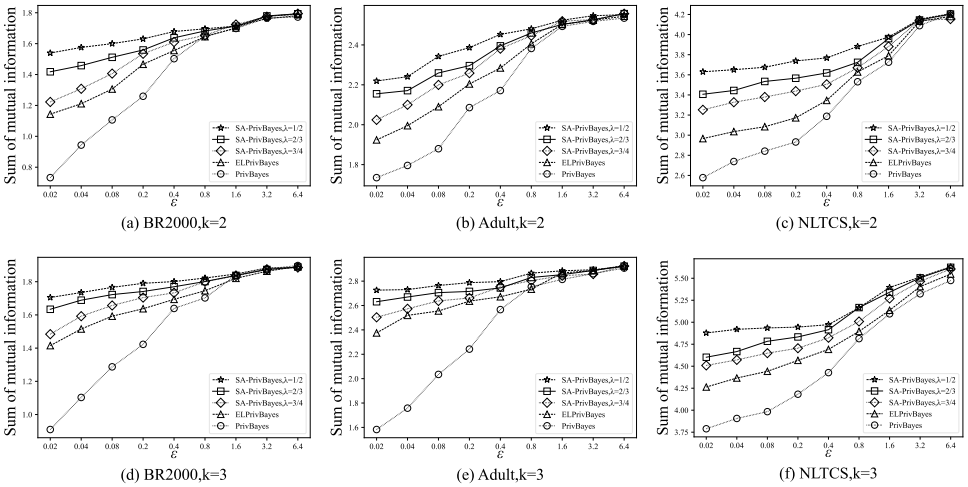


Figure 3. Mutual information captured by structure learning with differential privacy when k is 2 and 3.

From Figure 3, it can be observed that as the privacy budget increases, the accuracy of the Bayesian networks generated by our algorithm consistently outperforms those of ELPrivBayes and PrivBayes. Compared to ELPrivBayes, our algorithm restricts the exponential mechanism to selecting attribute-parent pairs with higher mutual information, thereby improving network precision. Compared to PrivBayes, our algorithm prioritizes the root nodes with the highest average mutual information. These root nodes exhibit stronger dependency relationships, and other nodes in the network are more closely connected to them, thereby enhancing network structure and accuracy. Under small privacy budgets, the exponential mechanism is susceptible to noise. Consequently, our algorithm captures substantially more mutual information than the other two algorithms. As the privacy budget increases, the intensity of privacy protection weakens, and the impact of noise diminishes. In this scenario, the performance gap between all algorithms gradually narrows. Specifically, SA-PrivBayes, ELPrivBayes, and PrivBayes all perform better at capturing the original data information under larger privacy budgets. However, as the privacy budget grows, the exponential mechanism's selection of mutual information becomes less influenced by noise, causing the results of the three algorithms to converge gradually.

At the same time, we observe that, for the ELPrivBayes algorithm, as the threshold λ increases, the Bayesian network's accuracy improves. From the six plots in Figure 3, it can be seen that when the threshold λ is set to 3/4, the amount of captured mutual

information does not show a significant improvement over the ELPrivBayes algorithm. This is because the threshold is still too low to completely filter out attribute pairs with low-scoring functions. When the threshold λ is set to $1/2$, as shown in subplots a, b, d, and e of Figure 3, the total captured mutual information does not change significantly as the privacy budget increases, and it gradually approaches the value in the non-differential privacy case. The reason is that when the threshold λ is sufficiently large, attribute nodes with relatively large mutual information values are also filtered out, so that almost only the attribute pairs with the largest mutual information are selected by the differential privacy exponential mechanism in each iteration. Even with the exponential mechanism, due to the limited candidate space for attribute pairs, the privacy protection effect is lost. Therefore, in this experiment, the threshold λ for the ELPrivBayes algorithm is set to $2/3$.

5.5. Data Utility Evaluation

The accuracy of the published dataset is one of the core indicators to measure the quality of privacy-preserving methods. In the experiments of this section, we select four algorithms—PrivBayes, ELPrivBayes, ACDP-Tree, and the proposed SA-PrivBayes—to compare data utility using two metrics: classification accuracy and average variance distance. The original dataset is split into a training set and a test set at a ratio of 0.8:0.2. Given the training set, each algorithm generates a synthetic dataset of the same size as the training set. On one hand, the synthetic dataset is used to train an SVM multi-classifier, where the classification labels are the income attribute in the Adult dataset, the religion attribute in the BR2000 dataset, and the eating attribute in the NLTCS dataset respectively; the classification accuracy of the original test set on this classifier is used to measure the classification utility of the data. On the other hand, the average variance distance is obtained by calculating the variance distance between the synthetic dataset and the original training set across all attributes, then averaging the results. A smaller value of this metric indicates a stronger consistency between the statistical distribution of the synthetic dataset and the original data, thereby comprehensively evaluating the data utility of different algorithms. Let the total privacy budget be denoted as ε. The allocation strategy follows $\varepsilon_1 = 0.3\ \varepsilon$, $\varepsilon_2 = \varepsilon - \varepsilon_1$, where ε_1 is used to construct the noisy Bayesian network, and the remaining budget ε_2 is allocated to generating the noisy distribution. During Bayesian network construction, the network's degree is set to 2 by default. To ensure reliability and minimize random errors, each algorithm is executed 50 times, and the average result is used as the final metric. The following experiments are conducted with the SA-PrivBayes algorithm, where the value of k is set to 2, the threshold λ is set to $2/3$, and the privacy budget allocation ratios between each cluster are set to $q = 1.0$, $q = 1.1$, and $q = 1.2$. In the probabilistic screening step of the algorithm, the fuzzy coefficient α is set to 0.2, which balances the sensitivity-smoothing effect and data utility loss. Higher-information attribute clusters are assigned smaller privacy budgets, indicating stronger privacy protection, while lower-information clusters are assigned larger budgets, indicating weaker protection. This achieves rational privacy budget allocation. The experimental results are shown in Figure 4. As observed in Figure 4, the classification accuracy of all three algorithms on sensitive attributes across the datasets increases with a larger privacy budget ε. This is because higher εvalues reduce the amount of added noise, thereby decreasing data distortion and weakening privacy protection.

From subplots a, b, and c of Figure 4, it can be seen that when $q = 1$, the SA-PrivBayes algorithm allocates equal privacy budgets, achieving the highest accuracy with the SVM classifier. This is because our algorithm reduces the exponential mechanism's candidate space during the Bayesian network construction phase, filtering out attribute pairs with low mutual information, resulting in a synthetic Bayesian network with higher accuracy

than the other three algorithms. From Figure 4, it can be observed that as the scaling constant q increases, the algorithm's classification accuracy gradually decreases. This is because as q increases, the privacy budget allocated to important attribute nodes becomes smaller, which significantly affects the accuracy of the synthetic data. When $q = 1.2$, as shown in subplots a and c of Figure 4, the accuracy of our algorithm approaches that of the ELPrivBayes algorithm. As q increases further, our algorithm's accuracy may fall below that of ELPrivBayes.

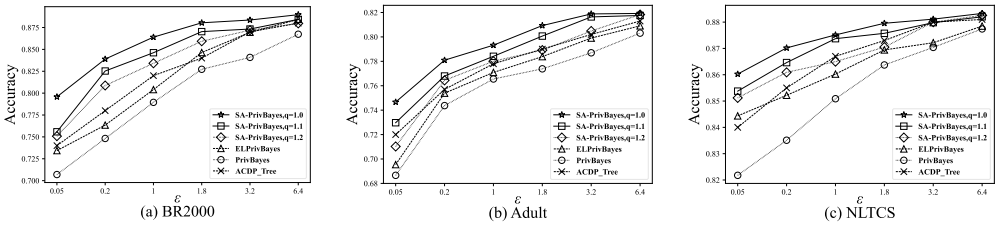


Figure 4. Under different settings of privacy budget ε and proportional parameter q, the classification accuracy of synthetic data generated by the target algorithm and benchmark methods.

Although the accuracy of our algorithm gradually decreases as q increases, it still provides stronger privacy protection for the data under the same privacy budget. In other words, our algorithm can achieve the same level of privacy protection as PrivBayes with a smaller privacy budget when q is larger. The larger the value of q, the stronger the data protection capability of our algorithm. From subplots a and b of Figure 3, it can be seen that when $q = 1.1$, the SVM classification accuracy of the SA-PrivBayes algorithm is not much lower than when $q = 1$ with equal privacy budget allocation, and the results are very close. Therefore, we can set $q = 1.1$ for privacy budget allocation between attribute pairs, as in this case, our algorithm improves data security while ensuring data utility. In subplot c, we observe a larger accuracy gap between the two algorithms, indicating that the scaling constant q should be dynamically adjusted to meet the different security and utility requirements of the private data pairs. Compared to other methods, SA-PrivBayes achieves relatively higher classification accuracy on sensitive attributes across all three datasets. This indicates that the noisy datasets generated by SA-PrivBayes maintain a reasonable level of data security while offering superior data utility. By dynamically allocating the privacy budget based on attribute information content, SA-PrivBayes effectively enhances the usability of synthetic data without compromising the total privacy budget.

In Figure 5, we compare the SA-PrivBayes algorithm with a threshold $\lambda = 2/3$ and a scaling constant $q = 1.1$ with the other three algorithms, in terms of the average variation distance of the 2D marginal joint probability distribution between the generated data and the original data. The average variation distance measures the degree of fit between the generated data and the original data. A smaller average variation distance indicates greater similarity between the generated and original data. From the figure, it can be seen that our algorithm consistently has a lower average variation distance than the other three algorithms. This is because SA-PrivBayes captures the global dependencies among attributes based on Bayesian networks, enables precise protection of high-importance attributes through dynamic privacy budget allocation, and preserves the global utility of data at the same time. In contrast, ACDP-Tree is built on classification trees combined with attribute generalization, a process that inevitably incurs partial information loss and is particularly deficient in capturing correlations among high-dimensional attributes compared with Bayesian networks. As the privacy budget increases, the average variation distance generally decreases. This is because a larger privacy budget results in less noise added during the Bayesian network construction and joint probability distribution generation

phases, leading to smaller data perturbations. Consequently, the fit between the generated data and the original data improves.

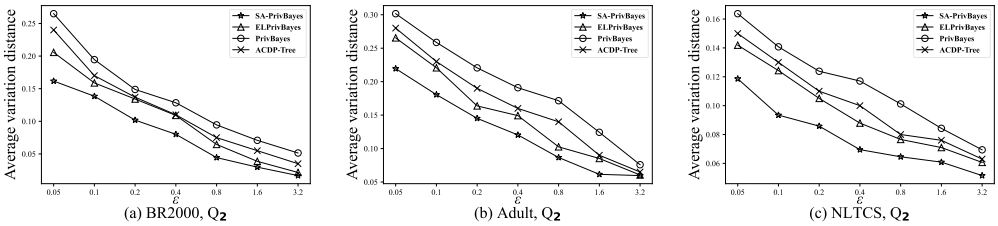


Figure 5. Average variance distance of the 2D marginal distribution between generated and original data at $q = 1.1$, $\lambda = 2/3$.

5.6. Data Robustness Analysis

Data robustness is a key factor supporting the practical application of high-dimensional differential privacy publication algorithms, directly influencing the reliability and generalization ability of the SA-PrivBayes algorithm in specific scenarios. To precisely verify the algorithm's robustness in high-dimensional datasets, this section designs a specialized validation scheme based on paired t-tests. Focusing on the BR2000 dataset with the core parameter combination of total privacy budget $\varepsilon = 1$, proportionality constant $q = 1.1$, and threshold $\lambda = 2/3$, a systematic analysis of data utility is conducted. Specifically, in five independent repeated experiments under identical conditions, SA-PrivBayes is compared with baseline algorithms (PrivBayes, ELPrivBayes) using paired-sample datasets. Using paired t-tests, the statistical significance of performance differences between algorithms is quantified, with a focus on verifying whether SA-PrivBayes consistently maintains data utility advantages under this parameter configuration and dataset. This provides rigorous statistical support for the algorithm's application in scenario-based settings. The specific experimental data are shown in Table 4.

Table 4. Significance Test Results of Classification Accuracy for SA-PrivBayes and Baseline Algorithms.

Experiment	ELPrivBayes Accuracy (%)	PrivBayes Accuracy (%)	SA-PrivBayes Accuracy (%)
1	79.3	78.5	81.2
2	79.8	79.0	80.8
3	79.5	78.8	81.5
4	79.7	79.2	80.9
5	79.2	78.6	81.1
Significance test (SA vs. EL)	$p = 0.0009$ ($t = 8.317$, $df = 4$), Conclusion: Highly significant ($p < 0.001$)		
Significance test (SA vs. Pr)	$p = 0.0005$ ($t = 10.324$, $df = 4$), Conclusion: Highly significant ($p < 0.001$)		

The results of the paired t-test comparing the classification accuracy of SA-PrivBayes with PrivBayes and ELPrivBayes show that:

- SA-PrivBayes has a mean accuracy of 81.10 ($SD = 0.29$), while PrivBayes has a mean accuracy of 78.82 ($SD = 0.26$). The mean difference between them is 2.28 ($SD = 0.493$), with $t(4) = 10.324$, $p = 0.0005$ ($p < 0.001$).
- ELPrivBayes has a mean accuracy of 79.50 ($SD = 0.26$), and the mean difference with SA-PrivBayes is 1.60 ($SD = 0.436$), with $t(4) = 8.317$, $p = 0.0009$ ($p < 0.001$).

The results indicate that the classification accuracy of SA-PrivBayes is significantly better than that of the two benchmark algorithms.

6. Conclusions

This paper addresses the limitation of existing differential privacy methods based on Bayesian networks, which uniformly allocate privacy budgets to all attribute nodes during parameter learning, resulting in suboptimal data utility and an inaccurate reflection of dataset quality. To resolve this, we propose a scoring function-based privacy budget allocation method. This method dynamically assigns privacy budgets to attribute nodes based on their importance, quantified by their scoring functions. To enhance data security in traditional algorithms, one must reduce the global privacy budget; however, in our approach, merely increasing the proportionality constant q achieves the same level of security strengthening. Furthermore, during the structure learning phase, the high sensitivity of the mutual information function often leads to the selection of attribute-parent pairs based on their mutual information scores. To mitigate this, we introduce a loop mechanism that iteratively ranks the sum of mutual information for AP pairs selected by the exponential mechanism. If an AP pair with a low-scoring function is chosen, the mechanism re-selects candidates, thereby improving the Bayesian network's construction. Ultimately, the proposed algorithm enhances data privacy while improving data utility. Compared with the PrivBayes algorithm, it achieves stronger privacy protection and higher data utility under the same privacy budget. Moreover, comparative experiments with the ACDP-Tree algorithm have verified its advantages across both medical and general high-dimensional data scenarios. However, the current research still has certain limitations. First, the algorithm has insufficient capacity to process ultra-high-dimensional, massive datasets; as the attribute dimensions and data scale increase continuously, the time complexity of Bayesian network structure learning rises significantly, and the efficiency of dynamic privacy budget allocation decreases accordingly. Second, the selection of the threshold and the proportional constant q currently relies on empirical settings and lacks an adaptive optimization mechanism, resulting in limited adaptability across datasets with heterogeneous features. To address the above limitations, future research will focus on two key aspects: first, optimizing the structure learning strategy of Bayesian networks by introducing the ideas of dimensionality reduction and divide-and-conquer to reduce the computational complexity of ultra-high-dimensional data and improve the processing efficiency of the algorithm; second, designing an adaptive parameter adjustment model based on data characteristics to realize the intelligent optimization of the threshold and the proportional constant q, thereby enhancing the algorithm's generality on heterogeneous datasets and further expanding its practical application scenarios and engineering value.

Author Contributions: Data curation, Q.Z., Y.L. and C.J.; formal analysis, K.Y., Q.Z. and Y.W.; funding acquisition, Y.L. and C.J.; investigation, Q.Z. and Y.W.; methodology, Q.Z. and Y.W.; project administration, K.Y., Y.L. and C.J.; resources, Y.L., Y.W. and C.J.; software, K.Y. and Q.Z.; validation, K.Y. and Y.W.; visualization, Y.L. and C.J.; writing—original draft, Q.Z.; writing—review and editing, K.Y., Y.L. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program under Grant 2018YFA0704703, the National Natural Science Foundation of China under Grants 61972073, 61972215 and 62172238, the Fundamental Research Funds for the Central Universities of China, the Key Specialized Research and Development Program of Henan Province under Grant 252102210172 and 232102210071.

Data Availability Statement: The data presented in this study were derived from the following publicly available third-party datasets: the Adult dataset is available at http://archive.ics.uci.edu/ml (accessed on 23 May 2025), the BR2000 dataset is available at https://international.ipums.org (accessed on 23 May 2025), and the NLTCS dataset is available at https://www.icpsr.umich.edu/web/NACDA/studies/9681/publications (accessed on 23 May 2025).

Acknowledgments: Special thanks are due to Ke Yuan, Yuye Wang, Chunfu Jia and Yinghao Lin for their expert mentorship and substantive contributions to this investigation.

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

References

1. Wang, H.; Wang, H. Correlated tuple data release via differential privacy. *Inf. Sci.* **2021**, *560*, 347–369. [CrossRef]
2. SWEENEY, L. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
3. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 3-es. [CrossRef]
4. Li, N.; Li, T.; Venkatasubramanian, S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115.
5. Wong, R.C.W.; Li, J.; Fu, A.W.C.; Wang, K. (α, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 20–23 August 2006; KDD '06, pp. 754–759. [CrossRef]
6. Dwork, C. Differential Privacy. In *Proceedings of the Automata, Languages and Programming*; Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–12.
7. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the Theory of Cryptography*; Halevi, S., Rabin, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
8. Narayanan, A.; Shmatikov, V. Robust De-anonymization of Large Sparse Datasets. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 18–22 May 2008; pp. 111–125.
9. Wang, Z.; Scott, D.W. Nonparametric density estimation for high-dimensional data—Algorithms and applications. *WIREs Comput. Stat.* **2019**, *11*, e1461. [CrossRef]
10. Amaratunga, D.; Cabrera, J.; Shkedy, Z. *Exploration and Analysis of DNA Microarray and Other High-Dimensional Data*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
11. Ren, X.; Yu, C.M.; Yu, W.; Yang, S.; Yang, X.; McCann, J.A.; Yu, P.S. LoPub: High-Dimensional Crowdsourced Data Publication with Local Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2151–2166. [CrossRef]
12. Cheng, X.; Tang, P.; Su, S.; Chen, R.; Wu, Z.; Zhu, B. Multi-Party High-Dimensional Data Publishing Under Differential Privacy. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1557–1571. [CrossRef]
13. Zhang, J.; Cormode, G.; Procopiuc, C.M.; Srivastava, D.; Xiao, X. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.* **2017**, *42*, 25. [CrossRef]
14. Hu, J. Survey on feature dimension reduction for high-dimensional data. *Appl. Res. Comput.* **2008**, *25*, 2601–2606.
15. Shi, Q.; Cong, S.; Tang, X. LSI_LDA: Mixture method for feature dimensionality reduction. *Appl. Res. Comput.* **2017**, *34*, 2269–2273.
16. Hao, Z.; Wang, R.; Cai, R.; Wen, W. Privacy data publishing method based on Bayesian network and semantic tree. *Comput. Eng.* **2019**, *45*, 124–129.
17. Zhang, X.; Chen, L.; Jin, K.; Meng, X. Private High-Dimensional Data Publication with Junction Tree. *J. Comput. Res. Dev.* **2018**, *55*, 2794–2809.
18. Li, X.; Luo, C.; Liu, P.; Wang, L.e. Information Entropy Differential Privacy: A Differential Privacy Protection Data Method Based on Rough Set Theory. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019; pp. 918–923.
19. Day, W.Y.; Li, N. Differentially Private Publishing of High-dimensional Data Using Sensitivity Control. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 17 March–14 April 2015; ASIA CCS '15, pp. 451–462. [CrossRef]
20. Xu, C.; Ren, J.; Zhang, Y.; Qin, Z.; Ren, K. DPPro: Differentially Private High-Dimensional Data Release via Random Projection. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 3081–3093. [CrossRef]
21. Li, M.; Ma, X. Bayesian Networks-Based Data Publishing Method Using Smooth Sensitivity. In Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, VIC, Australia, 11–13 December 2018; pp. 795–800.
22. Zhang, W.; Zhao, J.; Wei, F.; Chen, Y. Differentially Private High-Dimensional Data Publication via Markov Network. *EAI Endorsed Trans. Secur. Saf.* **2019**, *6*, 1. [CrossRef]

23. Abay, N.C.; Zhou, Y.; Kantarcioglu, M.; Thuraisingham, B.; Sweeney, L. Privacy Preserving Synthetic Data Release Using Deep Learning. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*; Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G., Eds.; Springer: Cham, Switzerland, 2019; pp. 510–526.
24. Beaulieu-Jones, B.K.; Wu, Z.S.; Williams, C.; Lee, R.; Bhavani, S.P.; Byrd, J.B.; Greene, C.S. Privacy-preserving generative deep neural networks support clinical data sharing. *Circ. Cardiovasc. Qual. Outcomes* **2019**, *12*, e005122. [CrossRef]
25. Frigerio, L.; de Oliveira, A.S.; Gomez, L.; Duverger, P. Differentially Private Generative Adversarial Networks for Time Series, Continuous, and Discrete Open Data. In *Proceedings of the ICT Systems Security and Privacy Protection*; Dhillon, G., Karlsson, F., Hedström, K., Zúquete, A., Eds.; Springer: Cham, Switzerland, 2019; pp. 151–164.
26. Li, H.; Xiong, L.; Zhang, L.; Jiang, X. DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing. *Proc. VLDB Endow.* **2014**, *7*, 1677–1680. [CrossRef]
27. Qi, X.; Ma, X.; Bai, X.; Li, W. Differential Privacy Preserving Data Publishing Based on Bayesian Network. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; pp. 1718–1726.
28. Zhang, Z.; Wang, T.; Li, N.; Honorio, J.; Backes, M.; He, S.; Chen, J.; Zhang, Y. PrivSyn: Differentially Private Data Synthesis. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Vancouver, BC, Canada, 11–13 August 2021; pp. 929–946.
29. Liu, G.; Tang, P.; Hu, C.; Jin, C.; Guo, S.; Stoyanovich, J.; Teubner, J.; Mamoulis, N.; Pitoura, E.; Mühlig, J. Multi-Dimensional Data Publishing With Local Differential Privacy. In Proceedings of the EDBT, Ioannina, Greece, 28–31 March, 2023; pp. 183–194.
30. Ni, G.; Sun, J. Differential privacy protection algorithm for large data sources based on normalized information entropy Bayesian network. *J. Phys. Conf. Ser.* **2024**, *2813*, 012012. [CrossRef]
31. Li, N.; Yang, J.; Ren, X.; Ma, X.; Li, X. Bayesian networks based on differential privacy for financial data privacy. *In Proceedings of the Third International Conference on Electrical, Electronics, and Information Engineering (EEIE 2024)*; SPIE: Wuhan, China, 2025; Volume 13512, pp. 276–281.
32. Zhang, S.; Li, X. Differential privacy medical data publishing method based on attribute correlation. *Sci. Rep.* **2022**, *12*, 15725. [CrossRef]
33. Chen, H.; Ni, Z.; Zhu, X.; Jin, Y.; Chen, Q.; Differential privacy high dimensional data publishing method based on cluster analysis *J. Comput. Appl.* **2021**, *41*, 2578–2585.
34. Lan, S.; Hong, J.; Chen, J.; Cai, J.; Wang, Y. Equation Chapter 1 Section 1 Differentially Private High-Dimensional Binary Data Publication via Adaptive Bayesian Network. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8693978. [CrossRef]
35. Liu, P.; Duan, L.; Shen, Z.; Wang, H. Differential privacy publishing of high-dimensional data based on attribute classification. *Appl. Res. Comput.* **2023**, *40*, 1–8.
36. Shen, G.; Cai, M.; Huang, Z.; Yang, Y.; Guo, F.; Wei, L. LoHDP: Adaptive local differential privacy for high-dimensional data publishing. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8039. [CrossRef]
37. Shi, W.; Zhang, X.; Chen, H.; Zhang, X. High dimensional data differential privacy protection publishing method based on association analysis. *Electronics* **2023**, *12*, 2779. [CrossRef]
38. Chen, Q.; Ni, Z.; Zhu, X.; Lyu, M.; Liu, W.; Xia, P. Dynamic Edge-Based High-Dimensional Data Aggregation with Differential Privacy. *Electronics* **2024**, *13*, 3346. [CrossRef]
39. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; pp. 94–103.
40. Kong, Y.; Tan, F.; Zhao, X.; Zhang, Z.; Bai, L.; Qian, Y. Review of K-means Algorithm Optimization Based on Differential Privacy. *J. Comput. Sci.* **2021**, *49*, 162–173.
41. Lu, X.; Piao, C.; Yang, X.; Bai, Y. Research on Differential Privacy High Dimensional Data Publishing Technology Based on Bayesian Networks. *Comput. Eng.* **2024**, *50*, 167–181.
42. Manton, K.G. National Long-Term Care Survey: 1982, 1984, 1989, 1994, and 2004. 2010. Available online: https://www.icpsr.umich.\-edu/web/NACDA/studies/9681/publications (accessed on 23 May 2025).
43. Bache, K.; Lichman, M. UCI Machine Learning Repository, 2013. Available online: http://archive.ics.uci.edu/ml (accessed on 23 May 2025).
44. Steven, R.; Katie, G.; Ronald, G.; Josiah, G.; Matthew, S. Integrated Public Use Microdata Series: Version 6.0.(2015). 2015. Available online: https://international.ipums.org (accessed on 23 May 2025).
45. Hong, J.; Wu, Y.; Cai, J.; Sun, L. Differentially Private High-Dimensional Binary Data Publication via Attribute Segmentation. *J. Comput. Res. Dev.* **2022**, *59*, 182–196.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.