

Article

A New Multidimensional Data Anonymization Algorithm for Privacy-Preserving Data Publishing

Burak Cem Kara ^{1,2}, Can Eyüpoğlu ^{3,4} and Oktay Karakuş ^{3,*}

¹ Battle Born AI, Nevada Center for Applied Research, University of Nevada, Reno, NV 89557, USA; burakcemkara@battleborn.ai

² Department of Computer Engineering, Istanbul Esenyurt University, 34510 Istanbul, Türkiye

³ School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK; eyupogluc@cardiff.ac.uk or can.eyupoglu@msu.edu.tr or caneyupoglu@gmail.com

⁴ Department of Computer Engineering, Turkish Air Force Academy, National Defence University, 34149 Istanbul, Türkiye

* Correspondence: karakuso@cardiff.ac.uk

Abstract

Developing a privacy-preserving data publishing algorithm that prevents individuals' identity information from being disclosed without ignoring the utility of the data is still an important goal to be achieved. Finding the optimal balance between data utility and data privacy is an NP-hard problem. In this study, a new k -anonymization-based data anonymization algorithm is proposed. The proposed algorithm partitions the data space with a new efficient partitioning strategy based on the k -dimensional tree (KD-tree), resolves the boundary problem while maintaining the trade-off balance, and introduces a new mechanism to address the problems caused by outliers. Moreover, it can be applied to both numerical and categorical data. The experimental results indicate that the proposed algorithm achieves competitive or improved performance compared with the baseline algorithms across seven commonly used evaluation metrics. Overall, the findings suggest that the proposed framework can improve the utility-privacy trade-off in multidimensional k -anonymization.

Keywords: data anonymization; information loss; privacy-preserving data publishing; k -anonymity; generalization

1. Introduction

In recent years, with the acceleration of technological advancements and the introduction of Industry 4.0, large amounts of data belonging to individuals are being generated and stored from various sources connected to the internet [1]. These data are used for several purposes such as scientific research, statistics, business and research analysis, and so on [2]. However, these data also contain some information that can seriously harm individuals' privacy. The effort to balance the great benefit that organizations derive from processed data with the need to protect individuals' data privacy has made privacy-preserving data publishing an active research area, and many anonymization algorithms have been proposed [3,4]. In addition, the introduction of the General Data Protection Regulation (GDPR) in 2018 has further increased the importance of this field [5]. In conclusion, privacy-preserving data publishing algorithms are critical for both individuals and organizations that generate value from data [6]. In addition, a large part of these data

Academic Editor: Andreas Mauthe

Received: 12 June 2026

Revised: 8 July 2026

Accepted: 14 July 2026

Published: 18 July 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

is collected through sensor-based systems such as wearable devices, connected vehicles, and smart city applications, which makes the anonymization of person-specific sensor data before publishing a practical requirement in many domains [7,8].

Different data de-identification technologies that can be used in privacy-preserving data publishing algorithms are available. The first method introduced among these technologies was anonymization [9]. Due to its primitive nature, stronger privacy protection models such as k -anonymity [10], l -diversity [11], t -closeness [12], and differential privacy [13] have been developed in the literature [14]. Each of these models has vulnerabilities to different types of attacks. In this study, k -anonymity-based approaches have been adopted for privacy-preserving.

k -anonymity based algorithms use operators such as generalization, masking, anonymization, permutation, and perturbation to anonymize datasets [15]. Generalization is the most common and powerful operator used for anonymization. The algorithms based on global full domain [16], local recoding [17], sub-tree schema [18], and multidimensional [19] methods are used to perform the generalization process on datasets [20]. The proposed algorithm in the study is based on a multidimensional approach, which provides more flexibility and higher data resolution and allows for more effective application of real-world data compared to other methods in the literature.

Developing a privacy-preserving data publishing algorithm that aims to find the best balance between the privacy risks for data owners and the benefits of data sharing for stakeholders is a challenging task. Therefore, the main motivation of this study is to develop an algorithm that maximizes data utility while preserving privacy. In this study, a new anonymization algorithm has been proposed that uses a k -dimensional tree (KD-tree)-based partitioning strategy to divide the dataset into subgroups, considers trade-off balance to solve the boundary problem, and addresses issues caused by outliers by introducing a new mechanism. Moreover, the proposed algorithm is applicable to not only numerical data but also categorical data. As a result, this study's contributions and original value to the literature in terms of conceptual and methodological aspects are as follows:

- A new outlier detection mechanism has been developed that manages outlier data, preventing the harm these data can cause to datasets, and enabling the benefit to be obtained from the entirety of outlier data without removing them from the dataset.
- A new mechanism for addressing the boundary problem while maintaining the utility-privacy trade-off, which can be adapted to a wide range of generalization-based anonymization algorithms in the literature, has been proposed.
- A new utility-focused partitioning strategy has been proposed that enables more effective use of data partitioning methods based on divide and conquer strategies such as KD-tree, providing utility-guaranteed equivalent classes for both categorical and numerical data.
- A new data anonymization model based on k -anonymization, which addresses the limitations of existing k -anonymization algorithms by combining the three mechanisms described above under a single framework and can be adapted to partitioning-based algorithms that follow the divide and conquer strategy, regardless of whether the data is categorical or numerical, and can be applied to real-world data, has been developed.

To position the contributions of this study precisely, it is useful to distinguish the genuinely novel components of the proposed framework from the components adapted from the literature. The COF algorithm, the KD-tree partitioning structure, and the normalized certainty penalty (NCP) and hierarchical certainty penalty (HCP) information-loss measures are existing techniques adopted from previous studies. The genuinely novel contributions of this study are (i) the equivalence class certainty penalty (EQCP), a new equivalence-class-level utility score that unifies the NCP and HCP measures into a single normalized indicator applicable to mixed-type (numerical and categorical) data; (ii) the utility-oriented boundary problem solution mechanism, which uses the EQCP score to selectively repartition low-utility equivalence classes instead of merging or suppressing them; and (iii) a new integration strategy that positions outlier handling in the preprocessing stage so that outlier records never participate in the partitioning process, and that reprocesses the outlier set with the same partitioning procedure instead of discarding it. The remaining parts of the framework constitute an engineering integration of these existing techniques. To the best of our knowledge, no previous k -anonymization-based algorithm addresses outlier handling, the boundary problem, and utility-oriented partitioning together within a single framework that is applicable to both numerical and categorical data.

The organization of the rest of the paper is as follows: In Section 2, the definition of existing problems in the literature has been provided, and the shortcomings in previous studies have been explained in detail, along with an explanation of the problems that the proposed algorithm solves. In Section 3, the privacy-preserving data publishing algorithm proposed in this paper is introduced. In Section 4, the dataset and metrics used in this study are introduced. In addition, the experimental results obtained by the proposed algorithm are also presented in this section. Finally, the study is concluded in Section 5.

2. Problem Definition and Related Work

Various recent studies aiming to develop new anonymization models by identifying the existing problems of k -anonymization-based algorithms in the literature are presented in Table 1 by comparing them based on various criteria such as scope, model, test dataset, compared approaches, compared metrics, enhancement target, and enhancement result. The k -anonymization-based methods in the table aimed to develop their models based on four main problems: the partitioning approach, the boundary problem, outliers, and the anonymization technique. In addition, each model has been examined in terms of its contribution to the literature in terms of either data privacy or data utility level.

The proposed algorithm in the study did not result in an improvement in terms of the anonymization technique from the four main headings in the problem space of k -anonymization-based algorithms; however, some necessary improvements and contributions were made for effective results in the other three criteria. The purpose of creating this table is to demonstrate that the proposed algorithm in the study is a new algorithm that focuses on problem areas still existing in the literature that have not yet been solved partially or completely. Additionally, the proposed algorithm has been added to Table 1 to fully understand its capabilities and differences from other existing models.

Table 1. Studies examining the problems related to k -anonymization-based algorithms.

Paper	Scope	Model	Test Dataset	Compared Approaches	Compared Metrics	Enhancement Target			Enhancement Result		
						Partitioning Approach	Boundary Problem	Outliers	Anonymization Techniques	Data Privacy	Data Utility
[19]	PPDP	Multidimensional k -anonymity (Mondrian)	- Adult (UCI) - Synthetic data	- Full-domain generalization [16,21] - Single-dimensional partitioning [22,23]	DM	Yes	No	No	No	Yes	Yes
[24]	PPDP	k -anonymity based on ball-tree and projection area density partition	- Adult (UCI) - Gotrack (UCI)	k-anonymity a model for protecting privacy [10] k-anonymity location privacy [25]	- Equivalence class size and number - Information loss	Yes	No	No	No	No	Yes
[26]	PPDP	Probabilistic optimal projection partition KD-tree k -anonymity	Mfeat-fac (UCI)	- Flexible partitioning algorithm [10,27] - An optimal projection partition k-anonymity of KD-tree [28]	- Information loss - Data quality - Run time	Yes	No	No	No	No	Yes
[29]	PPDP	Similarity-based clustering (SBCA)	- Adult (UCI) - Internet service provider (ISP)	- Mondrian [19] - Datafly [30] - Incognito [16]	Total normalized certainty penalty (Total NCP)	Yes	No	No	No	No	Yes
[31]	PPDP	KAB-based k -anonymization	Adult (UCI)	- Mondrian [19] - BHA-based k-anonymization [32]	- GenTotal-IL - GCP - CM - Cavg - Run time	Yes	No	No	No	No	Yes
[33]	PPDP	Hybrid k -anonymity	- Sal - Ucc	- Mondrian [19] - Incognito [16]	- LM - Query error rate	No	No	No	Yes	No	Yes

[34]	PPDP	A new approach to privacy-preserving multiple independent data publishing	•	Adult (UCI)	• ϵ -DP [35] • Hybrid algorithm [36] • Probabilistic (d, α) -linkable model [37] • (ρ, α) -anonymization model [38] • Mondrian [19]	• Distortion ratio • Query error rate • Privacy risk • Run time	No	No	No	Yes	Yes	Yes
[39]	PPDP	k -anonymity without prior value of the threshold k	•	Synthetic data	• No comparison	• No metric applied	No	No	No	Yes	Yes	No
[40]	PPDP	Optimized multidimensional k -anonymity model	•	Adult (UCI)	• [41]	• Information loss	Yes	No	No	Yes	No	Yes
[42]	PPDP	Clustering based k -anonymity	•	Adult (UCI)	• KACA [43] • Incognito [16]	• Information loss • Run time	Yes	No	No	No	No	Yes
[44]	PPDP	Improved k -anonymity algorithm (IKA)	•	Adult (UCI)	• Mondrian [19] • k -Members [45] • OKA [46]	• GCP	Yes	No	No	No	No	Yes
[47]	PPDP	An extended Mondrian (Xmondrian)	•	Adult (UCI)	• Mondrian [19]	• Cavg • Run time	Yes	No	No	No	No	Yes
[48]	PPDP	u -Mondrian	• •	Adult (UCI) Diabetes (UCI)	• Mondrian [19]	• GCP • DM • NES	No	Yes	Yes	No	No	Yes
[49]	PPDP	Outlier-oriented Mondrian	•	Adult (UCI)	• No comparison	• DM • NES	No	No	Yes	No	No	Yes
[50]	PPDP	Utility-based k -anonymization	•	Adult (UCI)	• Mondrian [19] • Bottom-up algorithm [51]	• NCP • Query error rate • DM	No	Yes	No	No	No	Yes
[52]	PPDP	ρ -Gain model	•	Adult (UCI)	• 5-anonymity and 2-diversity models	• Information loss • NES	No	No	Yes	No	No	Yes
[7]	PPDP	KAPP: k -anonymity for multidimensional data against skewness and similarity attacks	•	Real-world medical dataset	• CPPA • SKAM • PAMS	• Clustering accuracy • Diversity • Anonymity	Yes	No	No	Yes	Yes	No

					Information loss rate							
[53]	PPDP	Improved 2-means clustering and greedy algorithm-based k -anonymity	Adult (UCI)	2-means clustering with random center selection	- Information loss - Silhouette coefficient	Yes	No	No	No	No	Yes	
[8]	PPDP	Multi-source (k,d) -anonymity and (k,l,d) -diversity	- Adult (UCI) - Census (UCI)	- Incognito [16] - Flash - Top-down - Mondrian [19]	- Information loss - Run time	No	No	No	Yes	Yes	Yes	
[54]	PPDP	(r,k,ϵ) -anonymization based on multidimensional outlier detection, k -anonymity and ϵ -differential privacy	Adult (UCI)	k-anonymity and ϵ-DP based algorithms	- Information loss - Query error rate	No	No	Yes	Yes	Yes	Yes	
Proposed method	PPDP	A new multidimensional data anonymization algorithm for privacy-preserving data publishing	Adult (UCI)	- Classic and basic Mondrian [19,41] - Incognito [16] - Datafly [30] - Clustering based k-anonymity [45,46]	- Equivalence class size and number - Information loss - Anonymity score - Trade-off	Yes	Yes	Yes	No	Yes	Yes	

In recent years, new k -anonymization-based models have continued to be proposed for different data publishing scenarios. Su et al. [7] developed the KAPP algorithm, which combines a clustering method based on improved African vultures optimization with the t -closeness model to protect multidimensional data against skewness and similarity attacks. Yuan et al. [53] proposed an improved k -anonymity algorithm that uses an information loss function based on information quantity theory and a 2-means clustering method with a mean-center initialization to increase data availability in big data environments. Lu and Song [8] introduced multi-source (k,d) -anonymity and (k,l,d) -diversity models to prevent linkability attacks in aggregated Internet of Vehicles datasets. In addition, in our previous study [54], a hybrid model combining a multidimensional outlier detection mechanism, k -anonymity, and ϵ -differential privacy was presented. Unlike [54], which provides privacy by adding controlled noise to the dataset, the algorithm proposed in this study preserves the truthfulness of the published data and focuses on solving the boundary problem together with a utility-focused partitioning strategy. These recent models are also included in Table 1; however, none of them addresses the partitioning approach, the boundary problem, and outliers together, which is the main focus of this study.

2.1. Outlier Problem

Outliers are a problem area that directly affects both data utility and data privacy for all datasets to be anonymized [55]. Many publicly available real datasets contain records that do not conform to the general model of the population, known as outliers [56]. An adversary can easily manipulate this dataset using outliers. It is known that outliers also reduce the total data utility, in addition to the negative effects on data privacy. For this reason, some studies in the literature manage outliers by removing them from the entire dataset [33,57–59]; there are also studies that try to benefit from outliers [48,49,52].

In this study, the studies that benefit from outlier data without removing them from the dataset were used as a reference for the proposed algorithm. Therefore, these studies were examined and added to Table 1. Canbay et al. [49] improved the shortcomings of the study by Aydos et al. [52] and developed a new algorithm. They claim that their proposed model increases data utility and reduces computational cost by managing outlier data. In addition, it is argued that the proposed model provides a better solution in terms of cost-effectiveness compared to the ρ -gain model presented by Aydos et al. [52].

Determining where to use the outlier detection mechanism in the algorithm is a crucial issue because detecting outliers from equivalent classes generated by including them in the splitting process will become a more challenging task. On the contrary, using the outlier detection mechanism before starting the dataset splitting process is a more reasonable approach. Separating the outlier data from the entire dataset will enable more balanced and closer records in the equivalent classes to be formed using normal data in the split operations. In Canbay et al.'s work [49], there is a notable deficiency in this regard. In their study, the outlier detection mechanism is used for the equivalent classes formed after the data is partitioned.

Creating an effective outlier detection mechanism is another important issue in identifying an algorithm that can accurately detect outlier data. Different methods have been adopted for the use in mechanisms developed to detect outlier data. In the study presented by Canbay et al. [49], a density-based approach was adopted using the local outlier factor (LOF) algorithm [60] to determine outlier data. With the LOF algorithm, data are scored according to their density, and the records with LOF scores greater than one have been shown to have a higher probability of being outlier data [61]. Other studies that use the LOF algorithm for outlier detection include those by [62,63]. In the study introduced by Tang et al. [64], it is believed that it is not necessary to have a condition for a record

with low density to be declared as outlier data, as stated in the LOF algorithm. Therefore, the connectivity-based outlier factor (COF) algorithm was proposed. The COF works by selecting a series of nearest neighbors using a cluster-based shortest path [64]. It is a more effective method when outlier data is in the middle of two clusters with similar densities. In this study, the COF algorithm was used to detect outlier data in the dataset, as it is an advanced version of the LOF algorithm.

2.2. Boundary Problem

k -anonymization-based algorithms use different methods to partition the data space [19,24,26,40]. The expected outcome of any partitioning method is that the size of resulting equivalent classes should be equal to the predetermined k privacy-level parameter. However, many partitioning methods that use the divide-and-conquer strategy cannot guarantee this requirement [65–67]. This problem is referred to as the boundary problem in the literature. When there are fewer records in the equivalent classes than the k privacy-level parameter, it is defined as a lower-bound problem, while having more records is defined as an upper-bound problem. Lower-bound problems lead to privacy-level issues, while upper-bound problems result in data utility issues. KD-tree, a commonly used partitioning method in many algorithms, also faces this problem. With this method, the number of records in the resulting subgroups during data space partitioning ranges from k to $2k-1$. This limitation presents a weakness that leads to significant data utility problems for algorithms using KD-tree partitioning. The Mondrian algorithm, which is a commonly used k -anonymization-based algorithm, is also subject to this disadvantage.

There are some studies in Table 1 that address issues related to the boundary problem [48,50]. Both studies in the table deal with upper-boundary problems. Tang et al. [50] solved the upper-boundary problem using an approach that only applies to relaxed partitioning in the Mondrian algorithm. However, even in the Mondrian model, there are two different partitioning strategies: strict and relaxed. Therefore, the solutions developed should address the general case rather than a specific case. Canbay et al.'s study [48] is another study that aims to address the shortcomings of Tang et al.'s study [50]. In this study, Canbay et al. aimed to improve the Mondrian algorithm. They ensured that more than k records from each equivalence class obtained with KD-tree were sent to the outlier dataset by removing them from the equivalence classes. Although this study has made significant contributions to solving the upper-bound problem, the use of the LOF algorithm to detect the distance between data points in equivalence classes has some limitations on the data utility success of the model. Because the LOF algorithm is an outlier detection algorithm, its use for deciding which normal data to remove from the equivalence class is controversial. In addition, new problems arise with many normal data that are removed from the equivalence classes and sent to the outlier set and converted to singular records because the diversity remains in the previous normal set. Because when these normal data enter the division process again in the outlier set, the probability of matching with close data is considerably reduced. These equivalence classes, which will be formed as of the second iteration and which are not few in number, will negatively affect the data utility. Moreover, presenting the algorithm's success only on numerical data in the study also indicates that the applicability of the algorithm on real-world data is limited.

2.3. Partition Approach Problems

Addressing weaknesses arising from partitioning approaches in algorithms is a widely studied research topic by many researchers. Table 1 presents some studies that address problems stemming from partitioning approaches in k -anonymization-based algorithms [29,31,40,42,44,47]. These algorithms use various methods based on KD-tree,

clustering-based, or divide-and-conquer strategies as data space partitioning strategies. The main goal of these studies is to obtain equivalent classes consisting of the records that are closer to each other when partitioning the data space. In Table 1, Aghdam and Sonehara [29] and Ni et al. [42] used clustering-based partitioning strategies to partition the data space. Some studies also use various methods based on the divide-and-conquer strategy, which are similar to the KD-tree method and its variants, such as binary search tree algorithms. Among these studies, Cheng et al. [24] used the ball-tree method to partition the data space, while Den and Ye [68] utilized R-tree method. In addition, there are a variety of methods in the literature that work based on a divide and conquer strategy other than these two methods [69,70]. Some studies in Table 1 attempt to address the shortcomings of the KD-tree method rather than replacing it [26,28,47]. These studies generally aim to increase data utility by addressing the shortcomings of the KD-tree approach.

KD-tree is a proven effective method used in k -anonymization-based algorithms [19]. However, like any method in the literature, the KD-tree method also has some vulnerabilities in its structure [66,67,71]. Therefore, the algorithms using the KD-tree method as a partitioning approach are directly affected by various disadvantages of this method. Although there are many disadvantages mentioned in the literature for the KD-tree structure, k -anonymization-based algorithms only use the KD-tree to partition the data. Therefore, some of the disadvantages mentioned in the literature for the KD-tree structure can be addressed for algorithms based on k -anonymization that use this method. These problems include the inflexible data space partitioning approach, the use of a frequency-based splitting value, and the inability to guarantee the proximity of records within each resulting subgroup after the partitioning process, as the data space is divided into a binary tree model. These issues can be identified as development goals for algorithms based on k -anonymization that use this method.

2.4. Anonymization Technique Problems

Privacy-preserving data publishing algorithms are built on the principle that an adversary should not be able to link the published data to individuals in any way. Various anonymization techniques are used to ensure this privacy standard. However, not all anonymization methods used in the models and algorithms in the literature always produce the best results. Therefore, some algorithms define these anonymization techniques as a problem area and seek to improve upon this limitation. Table 1 includes some studies that focus on improving the anonymization techniques [33,34,39,40].

3. Proposed Privacy-Preserving Data Publishing Algorithm

In this study, a new model has been developed to address the information loss and anonymity problems in terms of data utility and data privacy. The proposed algorithm aims to deal with the partition approach, boundary problem, and outlier data issues. k -anonymization-based algorithms consist of two main steps: the partitioning of the data space and the anonymization of the resulting equivalent classes. The basic steps mentioned above for the proposed algorithm have been modified and innovations have been added to achieve its realization. In addition, very few of the algorithms in the literature on privacy-preserving data publishing can be implemented on real-world data. The proposed algorithm has been developed with this capability. Taking all of these factors into account, this study has developed a new k -anonymization-based model that can be applied to real-world data, handle outliers and boundary problems, and overcome the shortcomings of existing partitioning approaches in the literature.

3.1. Developed Outlier Detection Mechanism

To create an effective outlier detection mechanism, it is important to choose the right outlier algorithm and use it in the most appropriate place. Firstly, important studies conducted in this field in the literature were examined, and the most optimal algorithm to detect outliers was determined. The COF algorithm [64] is used in the outlier detection mechanism developed for the proposed algorithm.

Definition 1 (COF). The COF is a measure obtained by dividing the actual distance at a point by the actual distance at its k -nearest neighbors [64]. In a dataset, objects are denoted by o and p , where $p \in D$ and k is a positive integer. The COF value is determined for a point p in relation to the k -nearest neighbors by the following procedure:

$$COF_k(p) = \frac{|N_k P|.ac - dist_{nk(p)}(p)}{\sum_{o \in N_k(p)} ac - dist_{N(k)}(o)} \quad (1)$$

The mechanisms that cannot accurately detect outliers are more likely to harm the data utility of the algorithm rather than increase it. In addition, a new approach has been adopted from the algorithms in the literature that develop outlier detection mechanisms to positively differentiate the proposed algorithm. With this new step, while dividing the data space, it is ensured that the outlier data cannot participate in the partitioning process from the very beginning. In the proposed algorithm, the dataset is divided into two parts in the preprocessing stage called the normal set and the outlier set. The normal set is calculated by subtracting the outlier set from the total set.

$$Normal_Set = \{Total_Set - Outlier_Set\} \quad (2)$$

The data partitioning steps continue over the normal set in each iteration, ensuring the acquisition of subgroups cleansed of outliers. This is a fundamental goal of all generalization-based anonymization algorithms since the outliers within the equivalent classes during the anonymization of records lead to excessive generalization of records. Therefore, the outlier data detection mechanism guarantees that the negative impacts of outliers on both data utility and data privacy are eliminated.

The data splitting process with the cleaned normal set is an important step towards achieving the goal of obtaining equivalent classes consisting of the closest records. In Figure 1b, the outlier data present in the total set has been detected using the COF algorithm. Figure 1c shows the normal data space to be partitioned, which has been cleansed of the outlier data.

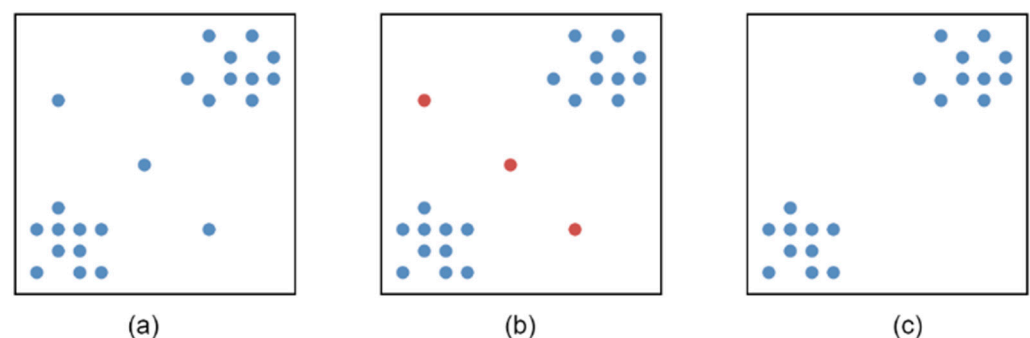


Figure 1. (a) The total set; (b) detection of outliers using COF; and (c) a normal set cleaned of outliers and ready to be partitioned. Red-colored nodes represent outliers, whereas blue-colored nodes represent normal data.

In the implementation of the developed outlier detection mechanism, the COF algorithm is applied to the entire dataset only once, during the preprocessing stage, before any partitioning operation is performed. As formulated in Definition 1, the COF score of a record expresses how strongly the record deviates from its nearest neighbors in terms of the average chaining distance; accordingly, the records with high COF scores, that is, the records that are isolated from their neighborhoods, are labeled as outliers and transferred to the outlier set, while the remaining records form the normal set given in Equation (2). It should also be emphasized that the outlier set obtained in this stage is not discarded: as shown in Section 3.4 and Algorithm 1, it is reprocessed by the same partitioning procedure in the subsequent iterations of the algorithm, so that the final anonymized dataset covers all of the records in the original dataset.

3.2. Developed Boundary Problem Solution

There are some studies in the literature on boundary problems, but considering the shortcomings discussed in the related work section, it is evident that there is a need for new studies in this field. In this study, a new partitioning strategy that solves the boundary problem and can be adapted to a wide range of existing generalization-based anonymization algorithms is proposed, while adhering to the trade-off balance. The working principle of the new partitioning strategy for the proposed algorithm is presented in Figure 2. Moreover, the developed algorithm guarantees there cannot be fewer than k privacy-level parameter values in an equivalent class. This demonstrates that the developed algorithm not only achieves gains in terms of data utility but also preserves data anonymity levels by considering the trade-off balance. This is clearly demonstrated in the study using anonymity score and trade-off balance metrics.

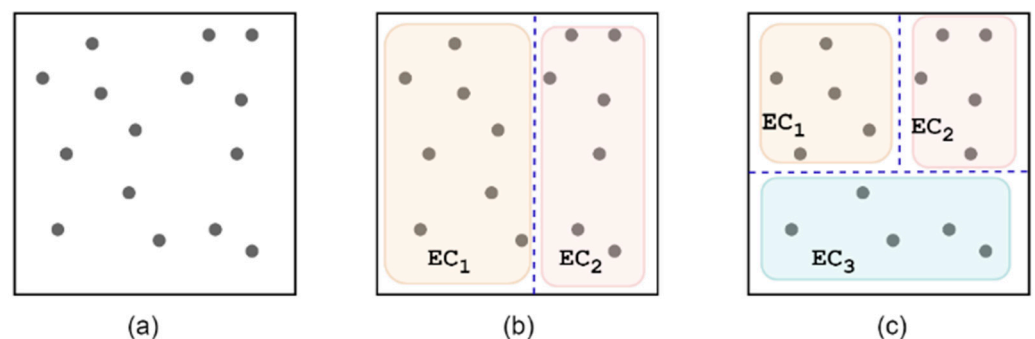


Figure 2. (a) The total set; (b) equivalent classes created by divide and conquer algorithms; and (c) equivalent classes created by the proposed algorithm.

In Figure 2b, the representative equivalent classes (ECs) obtained by k -anonymization algorithms based on a divide and conquer strategy such as KD-tree, ND-tree, ball-tree, etc. [65,66,70], are shown for a dataset with 15 records and $k = 5$. In such a partitioning process where the privacy-level parameter is initially set to 5, the expected outcome from the algorithm is to ensure that each equivalent class contains 5 records. However, as seen in Figure 2b, the expected outcome could not be achieved in practice due to the boundary problem, which harms the data utility. If this problem is explained with the discernibility metric (DM) that measures the data utility of equivalent classes:

- To measure the data utility of equivalent classes, we use the formula $DM = EC_1^2 + EC_2^2 + \dots + EC_n^2$ to represent the number of records in each equivalent class. In the DM metric, a lower score is better.
- The data utility in Figure 2b is $DM_1 = EC_1^2 + EC_2^2 = 8^2 + 7^2 = 113$.
- The data utility in Figure 2c is $DM_2 = EC_1^2 + EC_2^2 + EC_3^2 = 5^2 + 5^2 + 5^2 = 75$.

- The result $DM_1 > DM_2$ clearly shows that a solution considering the upper-boundary problem provides higher data utility compared to classical solutions.

In Figure 2c, the situation is the ultimate goal of every algorithm dealing with the boundary problem. In this partitioning approach, the size of each equivalent class is initially set to the value of k . However, having the size of equivalent classes equal to k alone does not always give the best result. When solving the boundary problem, operations that will harm the data utility should also be avoided. This is where the importance of the partitioning approach used in solving the boundary problem comes into play. The proposed algorithm provides a solution to the boundary problems while also obtaining equivalent classes consisting of records that are closest to each other. This feature distinguishes the proposed algorithm from previous boundary-problem-oriented approaches in the literature and may contribute to improved data utility.

3.3. Developed Partition Approach Strategy

In the proposed algorithm, the KD-tree method is used as the partitioning approach. In addition, other methods in the literature using divide-and-conquer strategies such as ND-tree, ball-tree, etc. [66,70], can also be applied to the proposed algorithm. In this respect, the proposed algorithm stands out and contributes to the literature. Since the KD-tree is a binary tree, the data is divided into two parts to obtain leaves or subgroups. Various frequency-based division values such as median and mode are used in the division operations. The expected result of these division operations is to capture the nearest elements in the smallest possible size. However, as the number of attributes in the datasets increases, it becomes more difficult to capture this proximity, and the resulting equivalent classes obtained through the generalization process cause significant data loss. In this study, a solution is proposed for the problem of the KD-tree method, which divides the data space into a binary tree model but cannot guarantee the proximity of records within each subgroup. Additionally, another important feature of the utility-focused decision mechanism presented in this study is that it can be applied to both categorical and numerical data. To better illustrate the working principle of the proposed utility-focused decision mechanism, a hierarchical table example for a city quasi-identifier (QID) is provided in Figure 3.

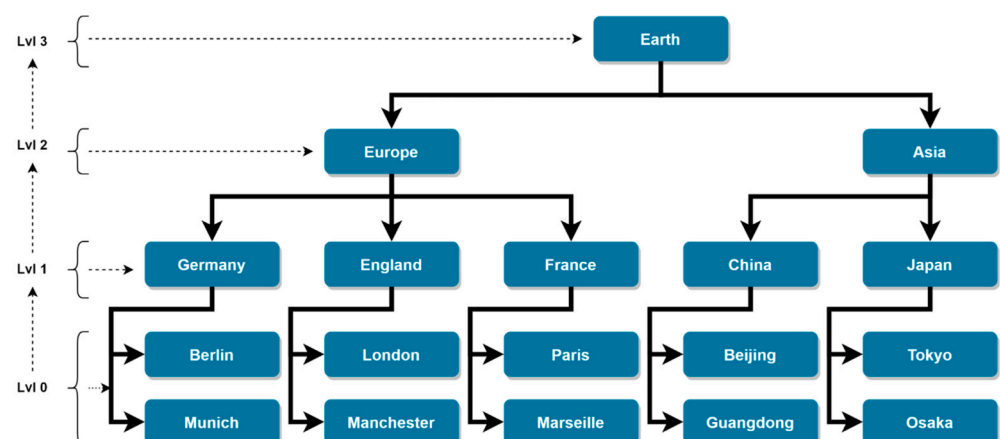


Figure 3. Generalization hierarchies for city QID.

A calculation process is performed using the HCP [21,27]. The metric for categorical attributes and certainty penalty scores are assigned in this way (see the step of the proposed algorithm in Section 3.4). When a generalization process is applied to the city QID in an equivalent class that includes Berlin, Paris, and Beijing in Figure 3, the common ancestor of these three data is the earth, so the generalization process will be at level 3.

Therefore, according to this result, the penalty score assigned to QID in such an equivalent class is equal to 1, which is the worst-case scenario called over-generalization, as $HCP_{Q_{Num}}(T) = (3 + 3 + 3)/(3 \times 3)$. If the data in the equivalent class consisted of Berlin, Munich, and another German city, and their common ancestor was Germany, the generalization would be at level 1. Thus, the penalty point assigned to the QID in such an equivalent class would be $HCP_{Q_{Num}}(T) = (1 + 1 + 1)/(3 \times 3)$, which equals 1/3. If the data in the equivalent class is not subject to any generalization process, then the penalty point assigned to the QID will be $HCP_{Q_{Num}}(T) = (0 + 0 + 0)/(3 \times 3)$, which equals 0, representing the best-case scenario in terms of data utility.

For numerical attributes, NCP is used to assign certainty penalty scores (see the step of the proposed algorithm in Section 3.4). The NCP metric is used to evaluate the accuracy loss in the equivalence classes [72]. It is a measure of the proximity of data in equivalent classes. For instance, considering an equivalent class containing individuals aged 36, 42, and 45 in a dataset where individuals have a minimum age of 10 and a maximum age of 90, the calculation is performed using the NCP metric. In this case, it is obtained as $NCP_{Q_{Num}}(T) = (45 - 36)/80$. The $NCP_{Q_{Num}}(T)$ score obtained here will be the best case as it approaches 0, and the worst case as it approaches 1. Thus, using these two metrics, the data utility of the resulting equivalent classes obtained through partitioning can be determined. The equivalent classes with certainty penalty scores lower than the normal set are separated and sent back to the partitioning process, thus overcoming the data loss problem caused by the generalization of equivalent classes obtained through data partitioning.

3.4. Algorithm Description

The proposed algorithm starts processing steps with the detection of outlier data using the COF algorithm in the preprocessing stage and removing the outlier data from the total set. It offers a new partitioning strategy designed to address current problems of k -anonymization-based algorithms and can also be adapted to divide-and-conquer methods such as KD-tree. In addition, the utility-oriented optimal boundary problem-solving mechanism applied to data partitions obtained between k and $2k-1$ with KD-tree prevents data loss in the generalization phase. To ensure the proper functioning of the utility-focused optimal boundary problem-solving mechanism, each data part obtained with KD-tree is sequentially traversed, and the certainty penalty of the equivalent classes (EQCP) is calculated. The EQCP is achieved by assigning precision penalty scores separately for categorical and numerical attributes. The steps of the proposed algorithm are presented below, the workflow is given in Figure 4 and the pseudocode is given in Algorithm 1.

Algorithm 1. Multidimensional data anonymization algorithm for PPDP

Input: D (dataset to be anonymized), k (privacy level), i (recursion depth index, $i = 0$ for the initial call)

Output: P^* (set of final equivalence classes, each containing at least k records)

```

1:  result ← ∅                                ▷ set of final equivalence classes
2:  outliers ← ∅                                ▷ outlier set
3:  if  $i = 0$  then                                ▷ preprocessing: outlier detection
4:    outliers ← COF(D)
5:     $D \leftarrow D \setminus \text{outliers}$                 ▷ normal set
6:  end if
7:  partitions ← {D}
8:  while partitions ≠ ∅ do                        ▷ KD-tree partitioning
9:    partition ← extract one partition from partitions
10:   if  $|\text{partition}| < 2k$  then                    ▷ partition cannot be split further
11:     result ← result ∪ {partition}

```

```

12:   else
13:       frequency ← calculateFrequency(partition)
14:       dimension ← chooseDimension(partition, frequency)
15:       median ← findMedian(partition, dimension)
16:       low, high ← split(partition, dimension, median)
17:       if |low| ≥ k and |high| ≥ k then
18:           partitions ← partitions ∪ {low, high}
19:       else
20:           presult ← presult ∪ {partition}
21:       end if
22:   end if
23: end while
24: for each p ∈ presult do           ▷ boundary problem solution
25:     if EQCP(p) > 0.5 then
26:         poutliers ← poutliers ∪ p
27:         presult ← presult \ {p}
28:     end if
29: end for
30: if |poutliers| ≥ k and poutliers ≠ D then    ▷ recursion stops when the outlier set no
longer changes
31:     presult ← presult ∪ Algorithm1(poutliers, k, i + 1)    ▷ recursive call
32: else if poutliers ≠ ∅ then
33:     presult ← split_to_k_partitions(presult ∪ poutliers)
34: end if
35: return presult

```

The stages that take place during the operation of the proposed algorithm are given below:

1. The dataset to be anonymized is provided as input to the algorithm.
2. Preprocessing is applied.
 - 2.1. The outlier detection mechanism comes into use.
 - 2.2. The COF algorithm is used to detect outliers in the total dataset.
 - 2.3. The dataset is divided into two subsets: normal and outlier.
3. KD-tree is applied to the normal set.
 - 3.1. The frequency of the data on the selected dimension is calculated.
 - 3.2. The median of these frequencies is found.
 - 3.3. Using the median value, the data space is divided into two subgroups: left partition and right partition.
 - 3.4. Partitions are obtained for each subgroup with a minimum size of k and a maximum size of $2k-1$.
4. The optimum solution to the boundary problem for the equivalent classes obtained from the normal sets and the data split process focused on utility from the equivalent classes starts.
 - 4.1. The equivalent classes obtained from the normal set are traversed one by one and flagged for each visited equivalent class.
 - 4.2. Find the equivalent classes that cause the boundary problem in the normal set and send them to the outlier set.
 - 4.3. For the equivalent classes whose sizes are equal to the privacy-level parameter, each attribute is evaluated separately according to whether it is categorical or numerical.

4.4. If the attribute is categorical, the calculations are made using $HCP_{Q_{Num}}(T)$, and the certainty penalty scores between 0 and 1 are assigned to attributes based on their common ancestor in the hierarchy table of the data.

- Let T be an equivalence class and Q be a QID attribute in the equivalence class. Assuming the levels of the anonymized records in the hierarchy table are lh , the number of records of the attribute is m , and the maximum hierarchy level of each attribute dimension is mh , the $HCP_{Q_{Num}}(T)$ of any categorical attribute is calculated as follows:

$$HCP_{Q_{Num}}(T) = \frac{\sum_j^m lh_j}{m * mh_{num}} \quad (3)$$

4.5. If the attribute is numerical, the calculations are made using the $NCP_{Q_{Num}}(T)$ metric, and if the distance between the maximum and minimum values of the numerical attributes in the equivalence class is small, scores close to 0 are obtained, and if it is large, scores close to 1 are obtained.

- Let T be an equivalence class and Q be a QID attribute in the equivalence class. In such a case, the $NCP_{Q_{Num}}(T)$ of any numerical attribute is calculated as follows:

$$NCP_{Q_{Num}}(T) = \frac{\max_{Q_{Num}}^T - \min_{Q_{Num}}^T}{\max_{Q_{Num}} - \min_{Q_{Num}}} \quad (4)$$

In the above formula, the numerator is obtained by subtracting the smallest QID value from the largest QID value in the T equivalence class. The denominator is obtained by subtracting the smallest QID value from the largest QID value in the total set.

4.6. If the certainty penalty scores assigned to the equivalent classes are greater than 0.5, to preserve data utility, all data in the equivalent class are sent to the outlier set, otherwise continue with step g.

- Certainty penalty for equivalent classes: Let us assume that there are i numerical and j categorical attributes, and a total of d attributes, in the equivalent class. If the certainty penalty scores assigned to categorical attributes are $HCP_{Q_j}(T_n)$, and the certainty penalty scores assigned to numerical attributes are $NCP_{Q_i}(T_n)$, then the certainty penalty scores $EQCP(T_n)$ assigned to equivalent classes are calculated using the following formula:

$$EQCP(T_n) = \frac{\sum_i^k NCP_{Q_i}(T_n) + \sum_j^l HCP_{Q_j}(T_n)}{d} \quad (5)$$

4.7. If there are no more equivalence classes to flag in the normal set, the process is terminated.

- Check whether the data space can be partitioned in the normal and outlier sets and continue the process if the data space can be partitioned, if there is no splittable data space or the iterations result in the same values from the outlier set and proceed to step 7.
- The outlier set, which consists of the records obtained by removing the ones detected by the COF algorithm in the preprocessing stage and the equivalent classes that cause the boundary problem and reduce the data utility in the normal set, is accepted as the new dataset for the next recursive partitioning process; however, since the outlier detection with the COF algorithm is performed only once during the initial preprocessing stage, the procedure continues from the KD-tree partitioning stage (step 3) rather than repeating the preprocessing step.
- Each equivalent class in the normal set is generalized and anonymized equivalent classes are obtained.
- All equivalent classes are combined, and an anonymous dataset is obtained.

Figure 4 illustrates the flowchart of the suggested algorithm, which aims to provide a more comprehensive explanation of the algorithm.

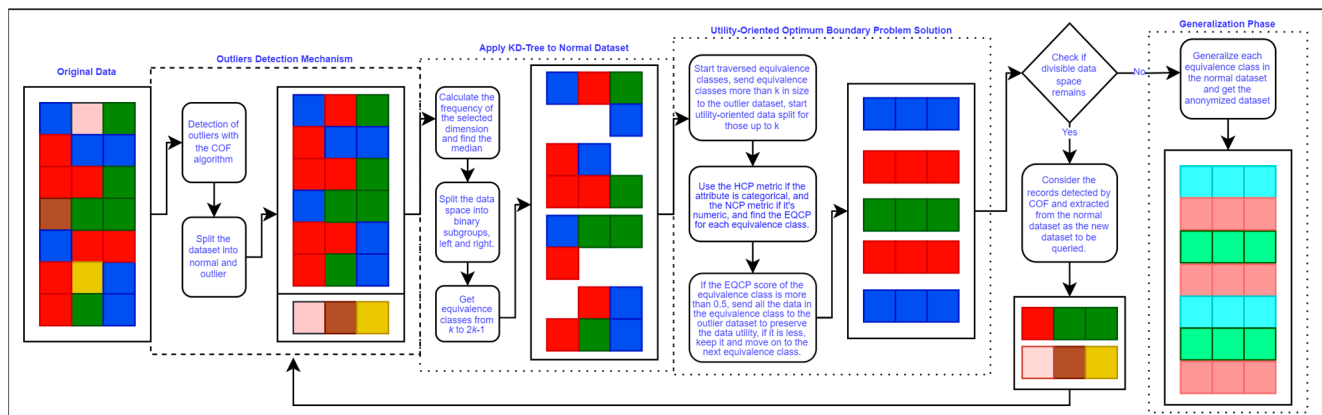


Figure 4. Workflow of the proposed algorithm.

The proposed algorithm preserves the k -anonymity requirement by construction. The KD-tree partitioning step splits a partition into two subgroups only if both resulting subgroups contain at least k records; otherwise, the partition is retained as a final equivalence class. Hence, every equivalence class added to the result set satisfies the size condition by construction, and because a partition whose size is smaller than $2k$ cannot be split further, the size of each final equivalence class is bounded between k and $2k-1$. The boundary problem solution mechanism does not violate this property: an equivalence class flagged by the EQCP criterion is removed from the result set as a whole and transferred to the outlier set, which is then reprocessed with the same partitioning procedure under the same size condition. Finally, if the residual outlier set contains fewer than k records, these records are not published as a separate class; instead, they are distributed among the closest existing equivalence classes through the `split_to_k_partitions` procedure. Consequently, the k -anonymity requirement holds for every published equivalence class.

The threshold value of 0.5 used in the EQCP-based boundary decision is a principled rather than an arbitrary choice. Since both the NCP and HCP components of the EQCP score are normalized to the interval $[0, 1]$, the EQCP score of an equivalence class also lies in $[0, 1]$, where 0 corresponds to no information loss and 1 corresponds to the worst-case over-generalization. The midpoint value of 0.5 therefore acts as a natural neutral decision boundary: equivalence classes that lose more than half of their potential utility are considered to suffer from the boundary problem and are sent back to the repartitioning process, whereas the classes preserving more than half of their utility are retained. Lower threshold values make the mechanism more aggressive, transferring more equivalence classes to the repartitioning stage, which may further reduce information loss at the cost of additional iterations, while higher values make the mechanism more conservative. Since the EQCP score is normalized, the midpoint value is dataset-independent; therefore, 0.5 is used as a balanced default threshold in this study.

3.5. Computational Complexity and Termination Analysis

The computational cost of the proposed algorithm is determined by three main components: the COF-based outlier detection, the KD-tree-based partitioning, and the EQCP-based boundary problem solution. The COF algorithm requires the computation of the nearest neighbors of each record. When the distances are computed over the d quasi-identifiers, this stage has a worst-case time complexity of $O(n^2 \cdot d)$ for a dataset with n records; it is executed only once, in the preprocessing stage. The KD-tree partitioning follows the

classical divide-and-conquer scheme: at each level of the tree, computing the attribute frequencies, selecting the splitting dimension over d quasi-identifiers, and finding the median require $O(n \cdot d)$ operations, and since the depth of the tree is $O(\log(n/k))$, the total cost of the partitioning stage is $O(n \cdot d \cdot \log(n/k))$. The EQCP evaluation of the resulting equivalence classes requires a single pass over all records and attributes, that is, $O(n \cdot d)$. Each repartitioning iteration is applied only to the records transferred to the outlier set; because the outlier set produced in an iteration is a strict subset of the records processed in that iteration and the algorithm terminates when the outlier set can no longer be partitioned or remains unchanged between two consecutive iterations, the number of iterations is finite and the recursive procedure is guaranteed to terminate. Therefore, the overall time complexity of the proposed algorithm is $O(n^2 \cdot d)$ in the worst case, dominated by the one-time COF stage, with an $O(n \cdot d \cdot \log(n/k))$ cost per partitioning iteration.

In terms of memory, the algorithm stores the dataset, the partition structures, and the neighborhood information required by the COF algorithm. If the pairwise distances are computed on demand and only the nearest-neighbor information is maintained, the space complexity is $O(n \cdot d)$; if a full pairwise distance matrix is stored instead, the memory requirement increases to $O(n^2)$. It should be emphasized that all of the compared multidimensional partitioning-based algorithms, such as Mondrian and its variants, share the same $O(n \cdot d \cdot \log(n/k))$ partitioning cost; the additional overhead introduced by the proposed method consists of the one-time COF preprocessing and the linear-time EQCP evaluations, which do not change the asymptotic order of the partitioning stage. Hence, the data utility and privacy gains reported in Section 4 are achieved at a modest additional computational cost, and an empirical runtime and scalability study on large-scale datasets is planned as future work.

4. Experimental Results and Evaluation

In this section, we analyze the performance of the proposed method. The algorithm was developed using PyCharm (version 2023.3.2) on a computer equipped with an Intel(R) Core (TM) i7-9750H 2.60 GHz CPU, 16 GB of memory, and a Windows 10 (64 bit) operating system. To measure the success of the proposed algorithm, the equivalence class size and number metrics, information loss metrics, anonymity score metric, and trade-off metric, which are frequently used in the literature, are used. The Adult dataset was used as the dataset.

All of the evaluated algorithms are deterministic: for a fixed dataset and a fixed privacy level k , each algorithm always produces the same anonymized output. Therefore, the reported metric values are deterministic rather than run dependent, and the comparative evaluation focuses on a wide range of privacy levels (k values) and on seven complementary utility and privacy metrics rather than on run-to-run variance.

The proposed algorithm is compared with the Datafly, Basic Mondrian, Mondrian, Top-down, and Cluster algorithms. The experimental comparison was limited to these baselines, which have publicly available and reproducibly implementable procedures, in order to ensure that all of the methods were evaluated under comparable conditions. More recent approaches that also address outliers, the boundary problem, or utility-aware anonymization are discussed conceptually in Section 2 and positioned with respect to the proposed algorithm in Table 1, because their full implementation details and parameter settings are not consistently available.

4.1. Dataset

In this study, the experiments are performed using the UCI Adult data [73]. Since the Adult dataset is frequently used in the literature, the success of our proposed algorithm can easily be compared with other algorithms. This dataset is available online from the

University of California-Irvine, Machine Learning Repository [74]. During the experiments, 30,162 of the 32,561 records of the Adult dataset, which did not contain missing values, were used. In addition, nine out of 15 features [75] in the adult dataset were used. Of these, eight are QIDs and one is a sensitive attribute. Of these eight QIDs, one is numeric and seven are categorical.

The Adult dataset is a real-world dataset derived from the U.S. Census Bureau records and constitutes the de facto standard benchmark in the privacy-preserving data publishing literature; the use of this benchmark enables a direct comparison with previous anonymization studies under comparable conditions. However, since the evaluation is limited to a single dataset, the generalizability of the results should be interpreted with this limitation in mind.

4.2. Equivalence Class Size and Number Metrics

In this section, the success of the proposed algorithm is demonstrated in terms of the metrics that calculate the size and number of equivalent classes.

4.2.1. Discernibility Metric

The DM is a metric that measures the similarity of a record to others within an equivalence class, with each record receiving an equal penalty score [22,76]. The size of the equivalence class in the T table is denoted as $|Z|$, and the penalty value assigned to a record belonging to the corresponding equivalence class is also $|Z|$. The DM formula can be expressed as follows:

$$DM(T) = \sum_{\text{EquivClasses } Z} |Z|^2 \quad (6)$$

The DM results of the proposed algorithm are compared with other studies in Figure 5. As the k value increases, the DM values of all algorithms increase. The proposed algorithm performs the best for almost all k values, with only the Cluster algorithm obtaining results close to it. The DM indicates that the proposed algorithm is more successful than the other algorithms.

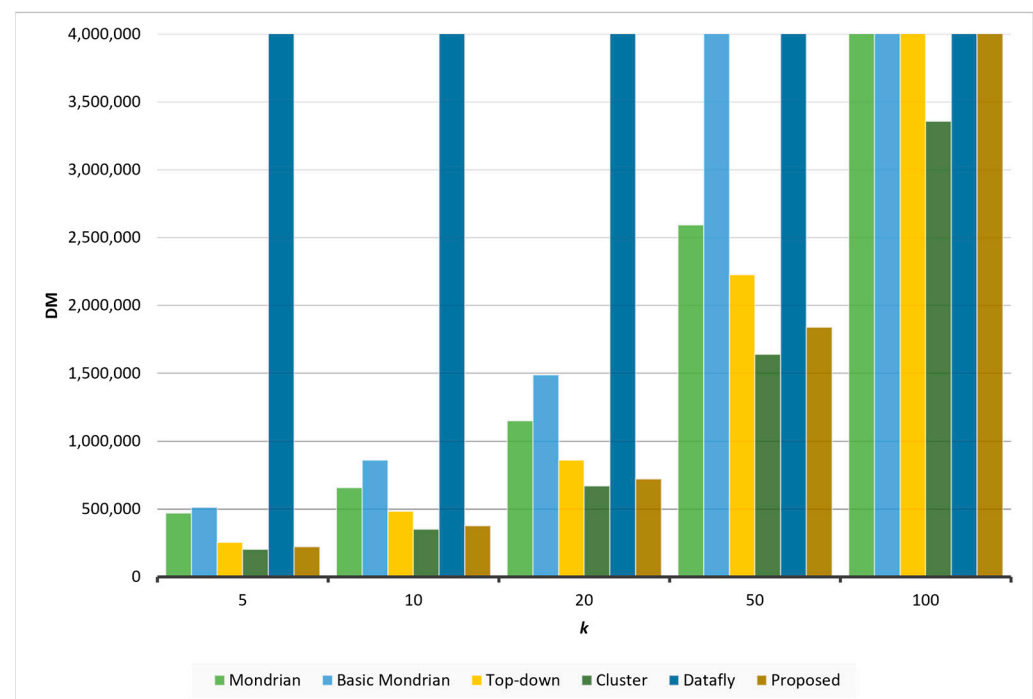


Figure 5. Comparison of discernibility metric results of the proposed algorithm with the other studies.

4.2.2. The Normalized Average Equivalence Class Size

The Cavg metric, which is used as an information measure, was first introduced in [19] and used in [17]. It measures the degree of data loss caused by applying the k -anonymization model to datasets by calculating the average size of the resulting equivalence classes [76]. The Cavg score reflects how well the partitioning approaches the optimal state where each equivalence class is generalized to a group of similar equivalence classes. The Cavg score of an anonymized table can be calculated using the following formula [31]:

$$Cavg = \frac{total_records}{total_equiv_classes} / (k) \quad (7)$$

The Cavg metric score for an anonymized table is calculated by dividing the total number of records in the dataset by the total number of generated equivalence classes, and then dividing the result by the specified privacy level k . This measures the similarity between the created equivalence classes and the best case where each equivalence class contains k records. An ideal Cavg score is one, indicating that the partitioning approaches the optimal state where each equivalence class contains k records [31].

In Figure 6, the performance of algorithms is evaluated in terms of the Cavg metric and presented in the graph. The Cavg measures the data utility based on the sizes of equivalent classes. A value of one is considered as a reference value for the Cavg metric, and algorithms that achieve results close to this value indicate better performance. When the graph is examined, it can be seen that the proposed algorithm performs better than other algorithms for all parameter values.

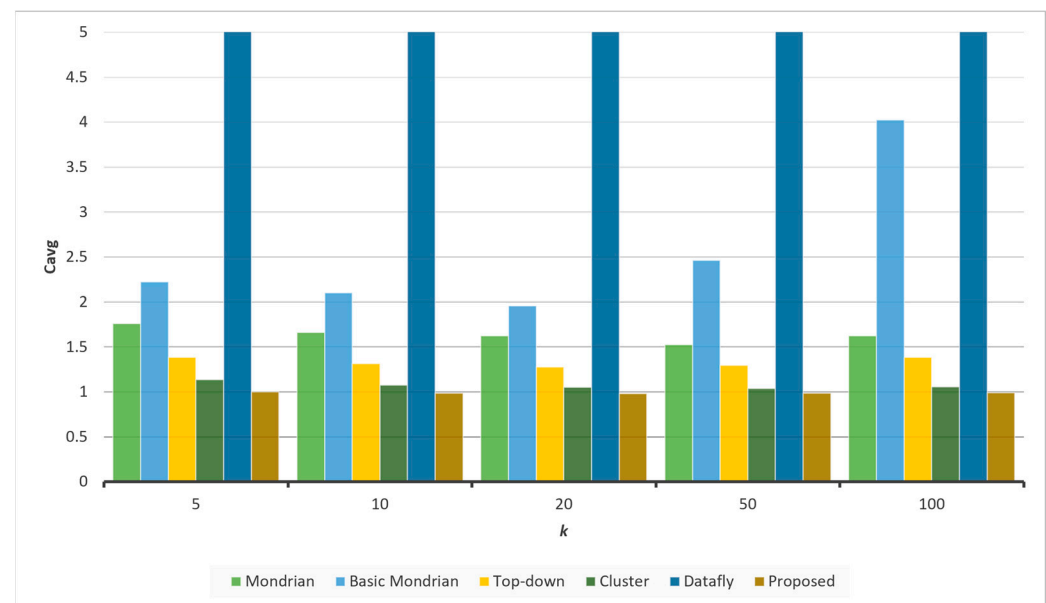


Figure 6. Comparison of normalized average equivalence class size results of the proposed algorithm with the other studies.

4.2.3. Number of Equivalence Classes

The number of equivalence classes (NES) function shows the number of equivalent classes that arise as a result of the partitioning approach applied to datasets. The purpose of using the NES function as a metric is to demonstrate how far the number of equivalence classes formed deviates from the reference value determined by the initially given privacy-level parameter. This is because the number of equivalent classes is an indicator that directly affects both data privacy and data utility.

In Figure 7, the numbers of subgroups that were formed on the Adult dataset as a result of the data partitioning process applied by the algorithms working based on the k -

anonymity model are presented. The reference numbers of equivalent classes that should be formed under normal conditions are also given in the graph, depending on the values of k . The most ideal situation is when the sizes of all equivalent classes are equal to k . According to these results, it can be observed that the proposed algorithm in the study shows an absolute success compared to other algorithms.

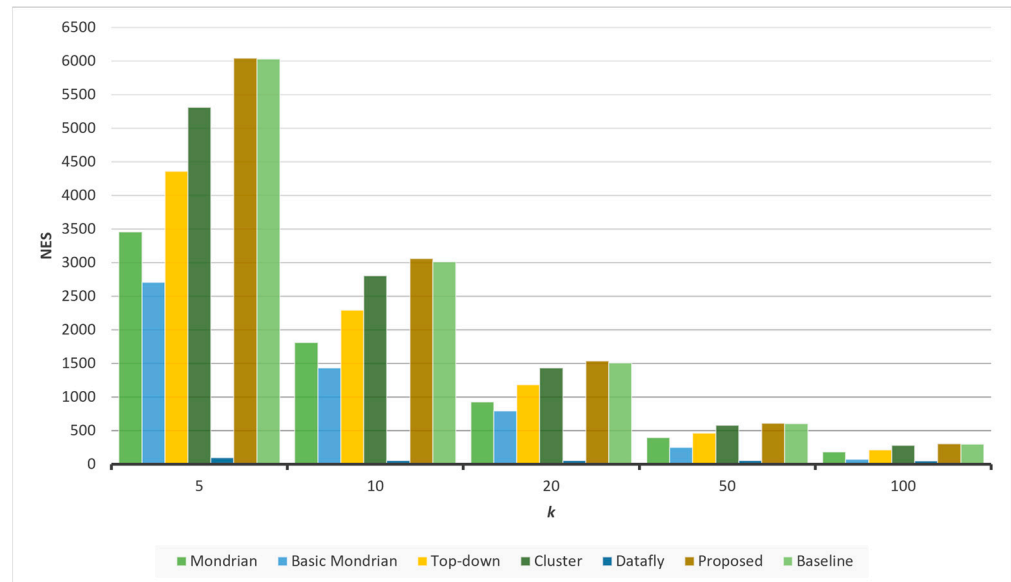


Figure 7. Comparison of number of equivalence classes of the proposed algorithm with the other studies.

4.3. Information Loss Metrics

Information loss metrics are used to assess the impact of privacy-preserving algorithms on data utility. In this section, a detailed explanation of the information loss metrics used to evaluate the proposed privacy-preserving algorithm is provided and the performance results of the algorithm against these metrics are presented.

4.3.1. Hierarchical Certainty Penalty

HCP is a general metric that calculates information loss by independently assigning penalty scores to each attribute according to the level of hierarchy in the generalization hierarchy table [21,27]. The HCP is obtained by first dividing the levels of each attribute dimension in the hierarchy table of anonymized records by the maximum hierarchy level of that dimension, and then dividing the result by the total number of records and the number of attributes. The HCP formula for the anonymized records in the hierarchy table of the T dataset, where the levels in the hierarchy table are lh , the total number of records is m , the number of attribute dimensions is d , and the maximum hierarchy level of each attribute dimension is mh , is given by the following equation.

$$HCP(T) = \frac{\sum_i^d \sum_j^m \frac{lh_{ij}}{mh_i}}{m * d} \quad (8)$$

Figure 8 presents the results of the HCP metric we utilize to measure the information loss in the anonymized dataset. It is a commonly used metric in the literature for datasets containing categorical and numerical data. The algorithms with lower HCP scores performed better in the graph. According to these results, as the value of k increases, the HCP of all algorithms gradually increases. This is because as the value of k increases, all algorithms distort more information to achieve k -anonymity. This negatively affects and

increases the HCP value. When we examine the performance of the proposed algorithm, we observe that it performs better than other algorithms for all k parameters.

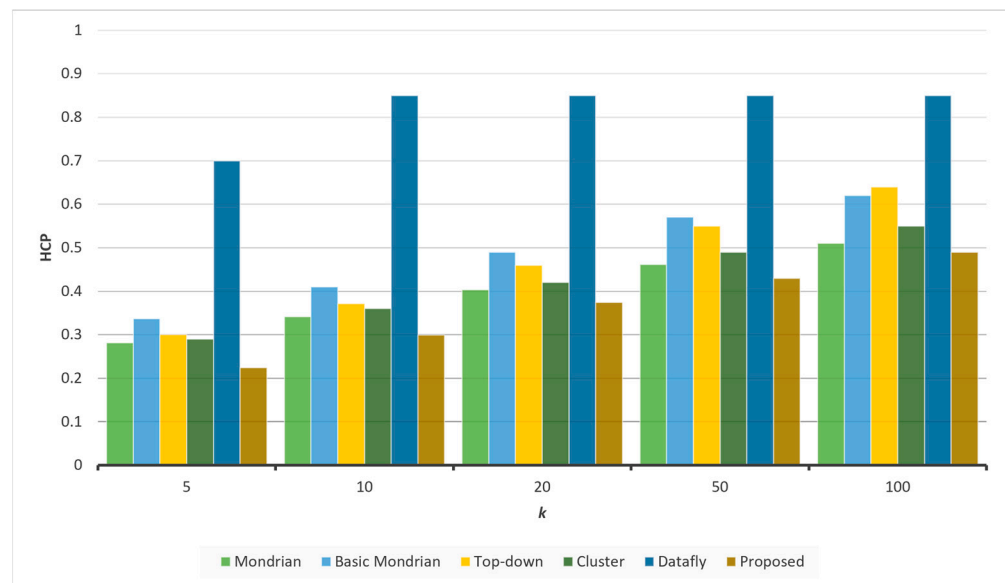


Figure 8. Comparison of hierarchic certainty penalty results of the proposed algorithm with the other studies.

4.3.2. Query Error Rate

The query error rate measures the effectiveness of a data processing method by analyzing the accessibility of the data after processing. This method involves applying SQL queries to both the raw data and the processed data and then calculating the difference using the relative error method, which indicates how much the method deviates from the expected outcome. The relative error is a type of error that shows how the results are close to the true value proportionally. It is more meaningful than absolute error because the results can vary in size depending on the circumstances. If A is the true value and a is the approximate value obtained from the calculation, the relative error is calculated using the formula below:

$$\delta a = \left| \frac{A - a}{A} \right| \quad (9)$$

The raw data and anonymized data were subjected to several SQL queries to demonstrate the proposed algorithm's success in terms of the query error rate. The queries implemented are shown below, and the results obtained from these queries are presented in Figure 9.

- Query 1: SELECT COUNT () FROM DATA_SET WHERE (WORKCLASS = "PRIVATE" OR WORKCLASS = "SELF-EMP-NOT-INC" OR WORKCLASS = "SELF-EMP-INC") AND AGE_MIN = 35
- Query 2: SELECT COUNT () FROM DATA_SET WHERE RACE = "BLACK" AND NATIVECOUNTRY = "UNITED-STATES"
- Query 3: SELECT COUNT () FROM DATA_SET WHERE (WORKCLASS = "PRIVATE" OR WORKCLASS = "SELF-EMP-NOT-INC" OR WORKCLASS = "SELF-EMP-INC") AND AGE_MIN > 55 AND RACE = "WHITE"
- Query 4: SELECT COUNT () FROM DATA_SET WHERE (WORKCLASS = "PRIVATE" OR WORKCLASS = "SELF-EMP-NOT-INC" OR WORKCLASS = "SELF-EMP-INC") AND AGE_MIN > 70 AND RACE = "WHITE"

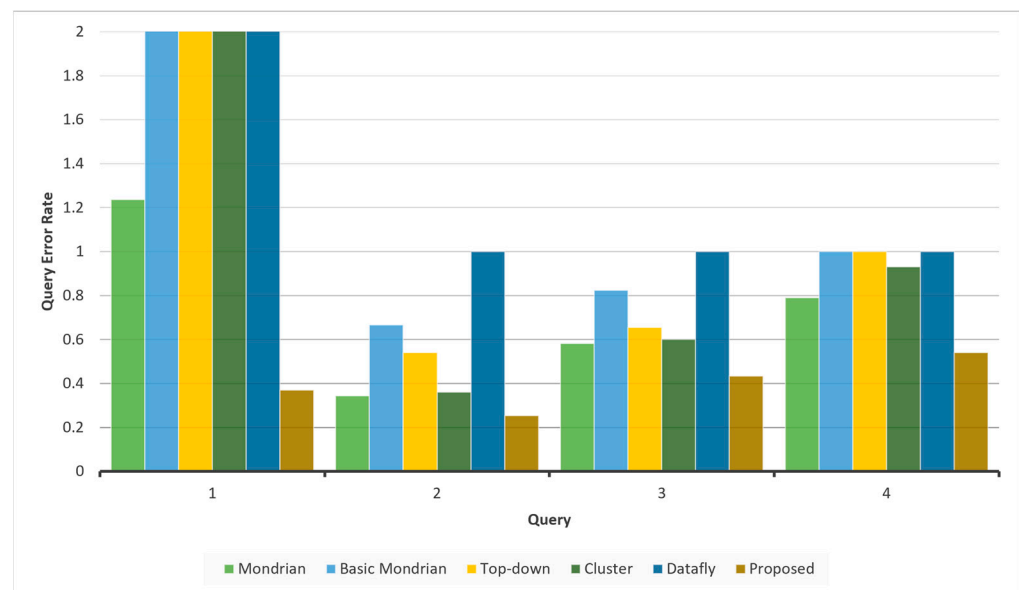


Figure 9. Comparison of query error rate results of the proposed algorithm with the other studies.

According to the results presented in Figure 9, for algorithms to be considered successful in terms of the error rate metric, their results should be close to zero. The proposed algorithm performed better than other algorithms in hierarchical queries such as Query 1, simple queries like Query 2, and sensitive queries such as Query 3 and Query 4. Query 3 and Query 4 are specific query types targeting more specific fields with values that are numerically lower than others in the dataset. Therefore, the success of the proposed algorithm in the results of these two query types highlights the gains of the algorithm in terms of data utility.

4.4. Anonymity Score Metric

An attacker should not be able to identify records belonging to individuals in a privacy-preserved dataset. The anonymity score metric is used to check whether the original record can be reached by querying each protected piece of data obtained by applying the proposed algorithm to the total set. Since the compared algorithms in this study are based on k -anonymization, the expected outcome in each query is that the returned count is at least k , the privacy-level parameter given initially. As it is known, the basic logic of k -anonymization is to ensure that there are at least $k-1$ records belonging to a record in an equivalence class [14,77]. The situation measured in this metric is exactly the extent to which this expected situation is achieved. Additionally, to better compare the output results obtained from all algorithms, the sigmoid function is used in this metric. The sigmoid function ensures that query results are between zero and one, making the results more understandable and effective.

Figure 10 shows how successful the proposed algorithm is in terms of the anonymity score metric. For this purpose, the anonymized dataset presented by each algorithm applied to the Adult dataset was taken, and the queries were performed for each record. The output results were then compressed in the (0,1) range using the sigmoid function. A value close to one for this metric indicates that a well-protected dataset output has been obtained, while a value close to zero indicates the opposite. In this metric, the expected result from the algorithms is not to outperform the other algorithms but to approach the value of one as much as possible. Because it should be known that each positive operation made in the level of data privacy actually means taking one step closer to excessively anonymizing the dataset. As this reduces the data utility, the operations performed with regard to the trade-off balance are more meaningful. When the graph is examined, it is

seen that the proposed algorithm achieves better results than the Cluster and Top-down algorithms, close to the Mondrian-based algorithms. The Datafly algorithm, on the other hand, exposed the dataset to over-generalization. Considering the trade-off, it is observed that the proposed algorithm has an acceptable anonymity score.

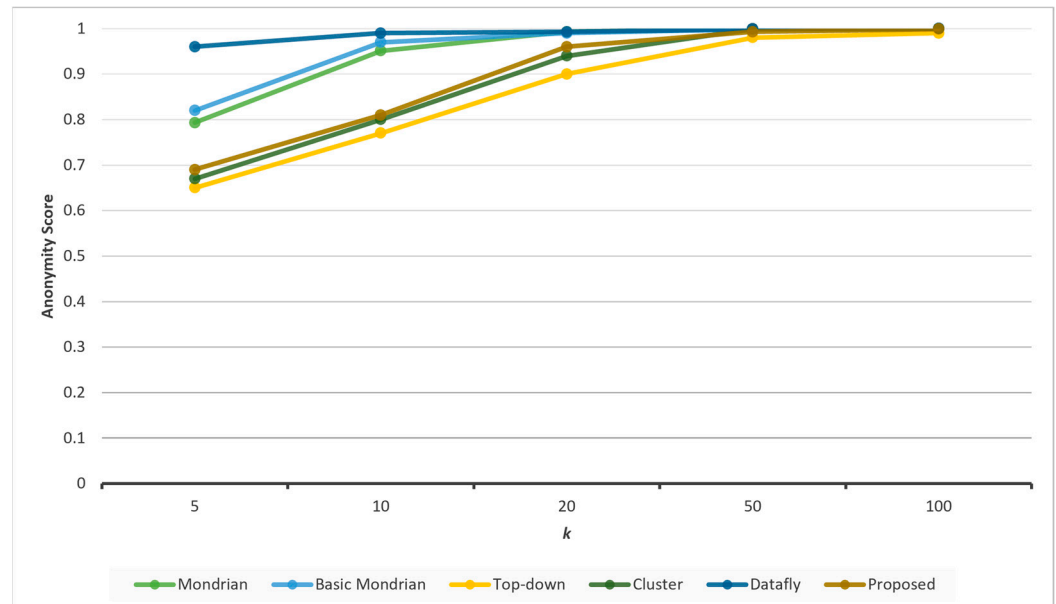


Figure 10. Comparison of anonymity score metric results of the proposed algorithm with the other studies.

4.5. Trade-Off Metric

Privacy-preserving data publishing aims to address two main issues. The first is to provide solutions for data privacy problems arising from each record in the dataset. The second is to ensure the data utility in the anonymized dataset remains at the highest level. Trade-off, in anonymization processes, is about determining the most appropriate balance between the requirements of data privacy and data utility. There are some studies in the literature that have been conducted on the trade-off metrics [54,78]. The general idea in these studies is to use metrics that measure both data utility and anonymity scores to determine the extent to which the model achieves a trade-off balance between the two requirements. The calculation of the trade-off metric is provided below:

$$\text{Trade-off metric} = \frac{\text{Information lost metric}}{\text{Anonymity score metric}} \quad (10)$$

To obtain the trade-off metric result, both a data utility metric and a data privacy metric are needed. In this context, the HCP metric was used as the data utility metric, and the anonymity score metric was utilized as the data privacy metric to obtain the trade-off metric result. Therefore, the most meaningful metric in this study is presented in Figure 11. Since the metrics used to calculate the trade-off metric are inversely proportional and the data utility metric is in the numerator, the criterion for this metric to be successful is to be close to zero value. The criteria for success of the data utility metric is to be close to zero, while for the data privacy metric, it should be close to one. When Figure 11 is examined, the proposed algorithm obtains lower trade-off metric values than the compared algorithms for each k value, indicating a more favorable balance between data utility and privacy in this experimental setting.

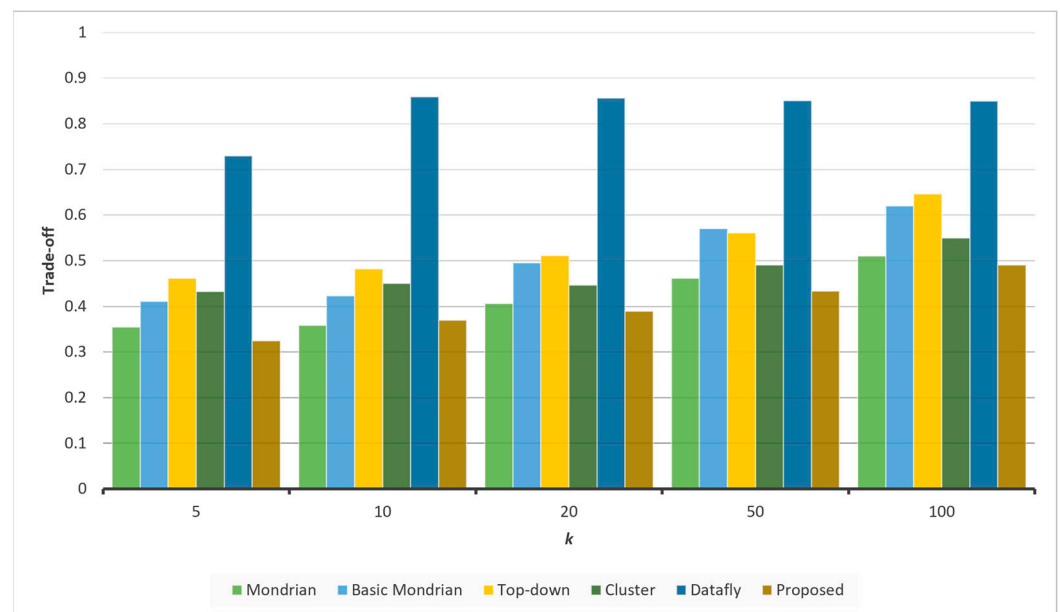


Figure 11. Comparison of trade-off metric results of the proposed algorithm with the other studies.

4.6. Privacy Threat Model and Discussion of Privacy Guarantees

The privacy threat model considered in this study is the identity disclosure (record linkage) attack, in which an adversary who knows the quasi-identifier values of a target individual attempts to link that individual to a specific record in the published dataset. The proposed algorithm guarantees that every published equivalence class contains at least k records that are indistinguishable with respect to their quasi-identifiers; therefore, the probability of correctly re-identifying an individual through a record linkage attack is bounded by $1/k$. Moreover, the developed outlier detection mechanism strengthens this protection in practice, because outlier records, which are the most vulnerable to re-identification due to their rare attribute combinations, are handled separately instead of forming small and distinctive equivalence classes.

On the other hand, like all k -anonymity-based models, the proposed algorithm does not, by itself, protect against attribute disclosure attacks. In a homogeneity attack, all of the records in an equivalence class share the same sensitive value, which allows an adversary to infer the sensitive attribute of an individual without identifying the exact record. In a background knowledge attack, an adversary uses external information to eliminate candidate records within an equivalence class. These attacks target the distribution of the sensitive attribute rather than the size of the equivalence classes and are therefore outside the protection scope of the k -anonymity model itself.

However, the proposed algorithm is fully compatible with sensitive-attribute-oriented privacy models. Since l -diversity and t -closeness are defined as additional constraints on the equivalence classes, they can be directly incorporated into the proposed framework by extending the partition acceptance condition: an equivalence class would be accepted only if it satisfies both the size condition of k -anonymity and the required diversity or closeness condition on the sensitive attribute. Similarly, differential privacy mechanisms can be applied on top of the outputs of the algorithm to provide formal guarantees against adversaries with arbitrary background knowledge. Integrating these models into the proposed framework and quantitatively evaluating the resistance of the resulting algorithm against homogeneity, background knowledge, and attribute disclosure attacks is planned as future work.

5. Conclusions

Privacy-preserving data publishing algorithms have been developed to deal with many different problems in the literature. The presence of problems such as the partitioning approach, the boundary problem, and outliers remains an area many researchers want to address. In this study, a new multidimensional algorithm addressing three important limitations in the field, namely the outlier problem, the boundary problem, and the partitioning approach problem, has been proposed. The proposed algorithm is also computationally efficient in terms of its asymptotic complexity and is designed to address the three main problems mentioned above and can be applied to both categorical and numerical data. Several performance metrics with different evaluation criteria were applied and tested to demonstrate the success of the proposed algorithm. The experimental results indicate that the proposed algorithm achieves competitive or improved performance compared with the other algorithms across all of the evaluated metrics. Additionally, the proposed algorithm contributes to the literature by integrating outlier handling, boundary problem control, and utility-oriented partitioning within a single framework, by being applicable to real-world data and by improving the trade-off balance between data utility and data privacy, which are the main objectives of algorithms and models in the literature but are difficult to achieve in practice.

In addition, several directions remain open for future work. First, the proposed algorithm will be evaluated on additional real-world datasets with different characteristics, such as datasets with predominantly numerical quasi-identifiers or higher dimensionality, in order to further demonstrate its generalizability. Second, a systematic sensitivity analysis of the EQCP threshold, together with an empirical runtime and scalability study on large-scale datasets, will be conducted. Finally, the integration of the proposed framework with sensitive-attribute-oriented privacy models such as l -diversity and t -closeness, as well as with differential privacy, will be investigated to strengthen its resistance against homogeneity and background knowledge attacks.

Author Contributions: Conceptualization, B.C.K. and C.E.; methodology, B.C.K. and C.E.; software, B.C.K.; validation, B.C.K., C.E. and O.K.; formal analysis, B.C.K.; investigation, B.C.K.; resources, B.C.K. and C.E.; data curation, B.C.K.; writing—original draft preparation, B.C.K.; writing—review and editing, C.E. and O.K.; visualization, B.C.K.; supervision, C.E. and O.K.; project administration, C.E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Scientific and Technological Research Council of Türkiye, under project No. 122E670. This work was also supported in part by Cardiff University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Adult dataset used in this study is publicly available online at the University of California-Irvine, Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/adult>, accessed on 15 March 2026).

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

References

1. Kara, B.C.; Eyupoglu, C. Privacy and Security Problems in Healthcare 4.0. In Proceedings of the 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 22–24 October 2020; pp. 1–12.
2. Sajjadi Mohammadabadi, S.M.; Kara, B.C.; Eyupoglu, C.; Uzay, C.; Tosun, M.S.; Karakuş, O. A Survey of Large Language Models: Evolution, Architectures, Adaptation, Benchmarking, Applications, Challenges, and Societal Implications. *Electronics* **2025**, *14*, 3580.

3. Mehta, B.B.; Rao, U.P. Improved l-diversity: Scalable anonymization approach for Privacy Preserving Big Data Publishing. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 1423–1430.
4. Jayapradha, J.; Prakash, M.; Alotaibi, Y.; Khalaf, O.I.; Alghamdi, S.A. Heap Bucketization Anonymity—An Efficient Privacy-Preserving Data Publishing Model for Multiple Sensitive Attributes. *IEEE Access* **2022**, *10*, 28773–28791.
5. Georgiadis, G.; Poels, G. Towards a privacy impact assessment methodology to support the requirements of the general data protection regulation in a big data analytics context: A systematic literature review. *Comput. Law Secur. Rev.* **2022**, *44*, 105640.
6. Kara, B.C.; Eyupoglu, C.; Uysal, S.; Bayrakli, S. Collection of an e-Health Dataset and Anonymization with Privacy-Preserving Data Publishing Algorithms. *Electrica* **2023**, *23*, 658–665.
7. Su, B.; Huang, J.; Miao, K.; Wang, Z.; Zhang, X.; Chen, Y. K-Anonymity Privacy Protection Algorithm for Multi-Dimensional Data against Skewness and Similarity Attacks. *Sensors* **2023**, *23*, 1554.
8. Lu, X.; Song, W. Efficient Multi-Source Anonymity for Aggregated Internet of Vehicles Datasets. *Appl. Sci.* **2024**, *14*, 3230.
9. Sweeney, L. *Simple Demographics Often Identify People Uniquely*; Data Privacy Working Paper 3; Carnegie Mellon University: Pittsburgh, PA, USA, 2000; pp. 1–34.
10. Sweeney, L. k-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570.
11. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramanian, M. ℓ -Diversity: Privacy Beyond k-Anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 3.
12. 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 April 2006–20 April 2007; pp. 106–115.
13. Dwork, C. Differential Privacy. In *Automata, Languages and Programming*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4052, pp. 1–12.
14. Kara, B.C.; Eyupoglu, C. Anonymization Methods for Privacy-Preserving Data Publishing. In *Smart Applications with Advanced Machine Learning and Human-Centred Problem Design. ICAIAME 2021; Engineering Cyber-Physical Systems and Critical Infrastructures*; Springer: Cham, Switzerland, 2023; Volume 1, pp. 145–159.
15. Gümüş, H.T.; Eyüpoğlu, C. Anonymization Techniques and Attack Types in Big Data: Application Examples. *İstanbul Commer. Univ. J. Sci.* **2022**, *21*, 422–441.
16. LeFevre, K.; DeWitt, D.J.; Ramakrishnan, R. Incognito: Efficient full-domain K-anonymity. In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, Baltimore, MD, USA, 14–16 June 2005; pp. 49–60.
17. Xu, J.; Wang, W.; Pei, J.; Wang, X.; Shi, B.; Fu, A.W.-C. Utility-based anonymization for privacy preservation with less information loss. *ACM SIGKDD Explor. Newsl.* **2006**, *8*, 21–30.
18. Fung, B.C.M.; Wang, K.; Yu, P.S. Anonymizing classification data for preserving privacy. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 711–725.
19. LeFevre, K.; DeWitt, D.J.; Ramakrishnan, R. Mondrian multidimensional K-anonymity. In Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, USA, 3–7 April 2006; p. 25.
20. Ashkouti, F.; Khamforoosh, K.; Sheikahmadi, A. DI-Mondrian: Distributed improved Mondrian for satisfaction of the L-diversity privacy model using Apache Spark. *Inf. Sci.* **2021**, *546*, 1–24.
21. Samarati, P. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.* **2001**, *13*, 1010–1027.
22. Bayardo, R.J.; Agrawal, R. Data privacy through optimal k-anonymization. In Proceedings of the 21st International Conference on Data Engineering (ICDE), Tokyo, Japan, 5–8 April 2005; pp. 217–228.
23. Iyengar, V.S. Transforming data to satisfy privacy constraints. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 279–288.
24. Cheng, C.; Xiaoli, L.; Linfeng, W.; Longxin, L.; Xiaofeng, W. Algorithm for k-anonymity based on ball-tree and projection area density partition. In Proceedings of the 14th International Conference on Computer Science and Education (ICCSE), Toronto, ON, Canada, 19–21 2019; pp. 972–975.
25. Zhang, Y.; Tong, W.; Zhong, S. On Designing Satisfaction-Ratio-Aware Truthful Incentive Mechanisms for k-Anonymity Location Privacy. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2528–2541.
26. Wang, X.; Luo, Y.; Jiang, Y.; Wu, W.; Yu, Q. Probabilistic optimal projection partition KD-Tree k-anonymity for data publishing privacy protection. *Intell. Data Anal.* **2018**, *22*, 1415–1437.
27. Sweeney, L. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 571–588.
28. Wang, X.; Luo, Y.; Jiang, Y. An optimal projection partition k-anonymity algorithm of KD-tree. *J. Nanjing Univ.* **2016**, *52*, 1050–1064.

29. Aghdam, M.R.S.; Sonehara, N. Achieving high data utility K-anonymization using similarity-based clustering model. *IEICE Trans. Inf. Syst.* **2016**, *99*, 2069–2078.
30. Sweeney, L. Datafly: A system for providing anonymity in medical data. In *Database Security XI: Status and Prospects*; Springer: Boston, MA, USA, 1998; pp. 356–381.
31. Djoudi, M.; Kacha, L.; Zitouni, A. KAB: A new k-anonymity approach based on black hole algorithm. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 4075–4088.
32. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184.
33. Nergiz, M.E.; Gök, M.Z. Hybrid k-Anonymity. *Comput. Secur.* **2014**, *44*, 51–63.
34. Hasan, A.S.M.T.; Jiang, Q.; Chen, H.; Wang, S. A new approach to privacy-preserving multiple independent data publishing. *Appl. Sci.* **2018**, *8*, 783.
35. Mohammed, N.; Chen, R.; Fung, B.C.M.; Yu, P.S. Differentially private data release for data mining. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 493–501.
36. Li, J.; Baig, M.M.; Sattar, A.H.M.S.; Ding, X.; Liu, J.; Vincent, M.W. A hybrid approach to prevent composition attacks for independent data releases. *Inf. Sci.* **2016**, *367*, 324–336.
37. Sattar, A.H.M.S.; Li, J.; Liu, J.; Heatherly, R.; Malin, B. A probabilistic approach to mitigate composition attacks on privacy in non-coordinated environments. *Knowl.-Based Syst.* **2014**, *67*, 361–372.
38. Baig, M.M.; Li, J.; Liu, J.; Ding, X.; Wang, H. Data privacy against composition attack. In Proceedings of the International Conference on Database Systems for Advanced Applications, Busan, Republic of Korea, April 2012; pp. 320–334.
39. El Ouazzani, Z.; El Bakkali, H. A new technique ensuring privacy in big data: K-Anonymity without prior value of the threshold k. *Procedia Comput. Sci.* **2018**, *127*, 52–59.
40. Liu, K.C.; Kuo, C.W.; Liao, W.C.; Wang, P.C. Optimized Data de-Identification Using Multidimensional k-Anonymity. In Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 1610–1614.
41. LeFevre, K.; DeWitt, D.J.; Ramakrishnan, R. Workload-Aware Anonymization. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 277–286.
42. Ni, S.; Xie, M.; Qian, Q. Clustering based k-anonymity algorithm for privacy preservation. *Int. J. Netw. Secur.* **2017**, *19*, 1062–1071.
43. Li, J.; Wong, R.C.W.; Fu, A.W.C.; Pei, J. Achieving k-anonymity by clustering in attribute hierarchical structures. In Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), Krakow, Poland, 4–8 September 2006; pp. 405–416.
44. Zheng, W.; Wang, Z.; Lv, T.; Ma, Y.; Jia, C. K-anonymity algorithm based on improved clustering. In Proceedings of the 18th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Guangzhou, China, 15–17 November 2018; pp. 462–476.
45. Byun, J.; Kamra, A.; Bertino, E.; Li, N. Efficient k-Anonymization Using Clustering Techniques. In Proceedings of the International Conference on Database Systems for Advanced Applications, Bangkok, Thailand, 9–12 April 2007; pp. 188–200.
46. Lin, J.L.; Wei, M.C. An efficient clustering method for k-anonymization. In Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society, Nantes, France, 29 March 2008; pp. 46–50.
47. Padmaja, R.; Santhi, V. An extended Mondrian algorithm—XMondrian to protect identity disclosure. *Adv. Parallel Comput.* **2021**, *38*, 480–490.
48. Canbay, Y.; Sagioglu, S.; Vural, Y. A new utility-aware anonymization model for privacy preserving data publishing. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6808.
49. Canbay, Y.; Vural, Y.; Sağiroğlu, Ş. OAN: Outlier record-oriented utility-based privacy preserving model. *J. Fac. Eng. Archit. Gazi Univ.* **2020**, *35*, 355–368.
50. Tang, Q.; Wu, Y.; Liao, S.; Wang, X. Utility-based k-anonymization. In Proceedings of the 6th International Conference on Networked Computing and Advanced Information Management, Seoul, Republic of Korea, 16–18 August 2010; pp. 318–323.
51. Xu, J.; Wang, W.; Pei, J.; Wang, X.; Shi, B.; Fu, A.W.C. Utility-based anonymization using local recoding. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 20–23 August 2006; pp. 785–790.
52. Vural, Y.; Aydos, M. q-gain: A utility based data publishing model. *J. Fac. Eng. Archit. Gazi Univ.* **2018**, *33*, 1355–1368.

53. Yuan, L.; Zhang, T.; Chen, Y.; Yang, Y.; Li, H. An Innovative K-Anonymity Privacy-Preserving Algorithm to Improve Data Availability in the Context of Big Data. *Comput. Mater. Contin.* **2024**, *79*, 1561–1579.
54. Kara, B.C.; Eyupoglu, C.; Karakuş, O. (r, k, ϵ) -Anonymization: Privacy-Preserving Data Publishing Algorithm Based on Multi-Dimensional Outlier Detection, k -Anonymity, and ϵ -Differential Privacy. *IEEE Access* **2025**, *13*, 70422–70435.
55. Kara, B.C.; Eyüpoğlu, C. A New Privacy-Preserving Data Publishing Algorithm Utilizing Connectivity-Based Outlier Factor and Mondrian Techniques. *Comput. Mater. Contin.* **2023**, *76*, 1515–1535.
56. Wang, H.W.; Liu, R. Hiding distinguished ones into crowd. In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Petersburg, Russia, 24–26 March 2009; pp. 624–635.
57. Yu, B.; Song, M.; Wang, L. Local isolation coefficient-based outlier mining algorithm. In Proceedings of the International Conference on Information Technology and Computer Science, Kiev, Ukraine, 25–26 July 2009; Volume 2, pp. 448–451.
58. Wang, H.; Liu, R. Hiding outliers into crowd: Privacy-preserving data publishing with outliers. *Data Knowl. Eng.* **2015**, *100*, 94–115.
59. Lee, H.; Kim, S.; Kim, J.W.; Chung, Y.D. Utility-preserving anonymization for health data publishing. *BMC Med. Inform. Decis. Mak.* **2017**, *17*, 104.
60. Breunig, M.M.; Kriegel, H.-P.; Ng, R.T.; Sander, J. LOF: Identifying Density-Based Local Outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; pp. 93–104.
61. Sun, L.; Zhou, K.; Zhang, X.; Yang, S. Outlier Data Treatment Methods Toward Smart Grid Applications. *IEEE Access* **2018**, *6*, 39849–39859.
62. Huang, T.; Zhu, Y.; Wu, Y.; Bressan, S.; Dobbie, G. Anomaly detection and identification scheme for VM live migration in cloud infrastructure. *Future Gener. Comput. Syst.* **2016**, *56*, 736–745.
63. Kim, S.; Cho, N.W.; Kang, B.; Kang, S.H. Fast outlier detection for very large log data. *Expert Syst. Appl.* **2011**, *38*, 9587–9596.
64. Tang, J.; Chen, Z.; Fu, A.W.; Cheung, D. A Robust Outlier Detection Scheme for Large Data Sets. In Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hong Kong, China, 16–18 April 2001.
65. Bentley, J.L. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* **1975**, *18*, 509–517.
66. Dolatshah, M.; Hadian, A.; Minaei-Bidgoli, B. Ball-tree: Efficient spatial indexing for constrained nearest-neighbor search in metric spaces. *arXiv* **2015**, arXiv:1511.00628.
67. Wang, J.; Wang, N.; Jia, Y.; Li, J.; Zeng, G.; Zha, H.; Hua, X.-S. Trinary-projection trees for approximate nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 388–403.
68. Deng, J.; Ye, X. Multi-dimensional K-anonymous algorithm based on R-tree. *Comput. Eng.* **2008**, *(1)*, 80–82.
69. Rafiee, M.; Abbasi, M. Pruned Kd-tree: A memory-efficient algorithm for multi-field packet classification. *SN Appl. Sci.* **2019**, *1*, 1537.
70. Qian, G.; Zhu, Q.; Xue, Q.; Pramanik, S. The ND-Tree: A Dynamic Indexing Technique for Multidimensional Non-ordered Discrete Data Spaces. In Proceedings of the 29th International Conference on Very Large Databases (VLDB), Berlin, Germany, 9–12 September 2003; pp. 620–631.
71. Kumar, N.; Zhang, L.; Nayar, S. What is a good nearest neighbors algorithm for finding similar patches in images? In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 364–378.
72. Terrovitis, M.; Mamoulis, N.; Kalnis, P. Local and global recoding methods for anonymizing set-valued data. *VLDB J.* **2011**, *20*, 83–106.
73. Kohavi, R.; Becker, B. Adult Data Set. Data Mining and Visualization, Silicon Graphics, 1996. Available online: <https://archive.ics.uci.edu/ml/datasets/adult> (accessed on 20 April 2022).
74. Lichman, M. UCI Machine Learning Repository. 2013. Available online: <https://archive.ics.uci.edu/ml/index.php> (accessed on 20 April 2020).
75. Eyupoglu, C.; Aydin, M.A.; Zaim, A.H.; Sertbas, A. An efficient big data anonymization algorithm based on chaos and perturbation techniques. *Entropy* **2018**, *20*, 373.
76. Li, J.; Wong, R.C.W.; Fu, A.W.C.; Pei, J. Anonymization by local recoding in data with attribute hierarchical taxonomies. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 1181–1194.
77. Eyüpoğlu, C. *Büyük Veride Etkin Gizlilik Koruması için Yazılım Tasarımı*; İstanbul Üniversitesi: Istanbul, Turkey, 2018.
78. Fung, B.C.M.; Wang, K.; Chen, R.; Yu, P.S. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.* **2010**, *42*, 1–53.

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.