

TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Hiroki Kaminaga IVGM177295

# **Augmented k-anonymity in the realm of Generative Adversarial Networks**

Master's Thesis

Supervisor: Dirk Draheim

Prof. Dr.

Co-Supervisor: Genlang Chen

A/Prof. Dr.

Zhejiang University

in China

Tallinn

2019

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Hiroki Kaminaga IVGM177295

## **K-anonüümsus GAN kontekstis**

Magistritöö

Juhendaja: Dirk Draheim  
Prof. Dr.

Kaasjuhendaja: Genlang Chen  
A/Prof. Dr.  
Zhejiang Ülikool  
Hiinas

Tallinn

2019

# **Author's Declaration**

I hereby certify that I am the sole author of this thesis. All the materials used, references to the literature and the work of others have been referred. This thesis has not been presented for examination previously.

Author: Hiroki Kaminaga

08.05.2019

# Abstract

As society and industry become more sophisticated and data becomes more important in all fields, the importance of managing data and related privacy is also increasing. On the other hand, the evolution of generative model algorithms such as General Adversarial Networks (GAN) has led to the production of highly similar data to the original data. This paper recommends using such generated data instead of the original data for privacy protection. At that time, the existing widely used k-anonymity is incorporated into the GAN architecture in order to introduce an indicator of privacy protection level of generated data. This allows the data provider to adjust the balance of data quality and privacy protection level in an explainable way when sharing data with third parties. In addition, it has been confirmed by experiments that using GAN maintains higher data quality at the same privacy protection level than the conventional method.

Keywords: k-anonymity; Generative Adversarial Networks; K-means clustering; microaggregation; database security; Statistical Disclosure Control; data utility; privacy protection

This thesis is written in English and is 36 pages long, including 8 chapters, 1 appendix, 16 figures, and 10 tables.

# Annotatsioon

Ühiskonna ja tööstuse arenedes ning andmete tähtsuse kasvades kõigis valdkondades, kasvab ka andmete haldamise ja privaatsuse olulisus. Generatiivse mudeli algoritmid, nagu Üldised Vastandlikud Võrgud (ing k Generatiivsed võistlusvõrgud – GAN), on viinud originaalandmetele ülisarnaste andmete tootmiseni. Antud töö soovib kasutada privaatsuse tagamiseks sellisel viisil loodud andmeid originaalandmete asemel. Laialdaselt kasutuses olev k-anonüümsus on kaasatud GAN-arhitektuuri, et tutvustada genereeritud andmete privaatsuskaitse taseme indikaatorit. See võimaldab andmete haldajal kohandada andmete kvaliteedi ja privaatsustaseme tasakaalu, kui andmeid tuleb jagada kolmanda osapoolega, talle arusaadaval viisil. Lisaks on eksperimentidega tõestatud, et GANi kasutamine tagab samal privaatsustasemel kõrgema andmete kvaliteedi kui konventsionaalsed meetodid.

Märksõnad: k-anonüümsus; Generatiivsed võistlusvõrgud; K-tähendab klastrit; mikroagregaat; andmebaasi turvalisus; Statistilise avalikustamise kontroll; andmete kasulikkus; privaatsuse kaitse

Magistritöö on inglise keeles, 36 lehekülge, 8 peatükki, 1 lisa, 16 joonist, 10 tabelit.

# Abbreviations and concepts

CCA	Chosen Ciphertext Attack
COA	Ciphertext Only Attack
CPA	Chosen Plaintext Attack
D	Discriminator
DCR	Distance to the Closest Record
Dec	Decryption
Enc	Encryption
FM	Failed Model
G	Generator
GAN	Generative Adversarial Networks
GEN	Generated data
GPU	Graphics Processing Unit
HMC	Hamiltonian Monte Carlo
ID	Identifier
KPA	Known Plaintext Attack
KDE	Kernel Density Estimation
MCMC	Markov chain Monte Carlo
NUTS	No-U-Turn Sampler
ORI	Original data
QID	Quasi-Identifier
RSA	Rivest, Shamir, & Adleman (public key cryptosystem)
RQ	Research Question
SDC	Statistical Disclosure Control
VEEGAN	Variational Encoder Enhancement to Generative Adversarial Networks
WGAN	Wasserstein Generative Adversarial Networks
$k$	parameter of k-anonymity
$K$	parameter of K-means clustering
$r$	radius

# List of Tables

Table 1: Example of k-anonymity	21
Table 2: The true values for the estimation	31
Table 3: Information on GAN architecture	31
Table 4: Results of Model 1 with the mixed Gaussian 8 distributions dataset	34
Table 5: Results of Model 1 with the dataset for Bayesian Linear Regression	35
Table 6: Mean and maximum value of the distance between each datum and its nearest centroid that belong to the same cluster on the data in Figure 10	37
Table 7: Mean and maximum value of the distance between each datum and its nearest centroid that belong to the same cluster on the dataset for Bayesian Linear Regression	37
Table 8: Results of Model 2 with the mixed Gaussian 8 distributions dataset	38
Table 9: Results of Model 2 with the dataset for Bayesian Linear Regression	39
Table 10: Results with the microaggregated dataset using K-means clustering (the comparison target)	40

# List of Figures

Figure 1: Overall objective of this research	12
Figure 2: Example of (3,3)-threshold secret sharing	16
Figure 3: The architecture of GAN	20
Figure 4: Bayesian Linear Regression Model	25
Figure 5: Example of the augmented k-anonymity, where $k = 3$	27
Figure 6: The mixed Gaussian 8 distributions dataset (left) and its KDE result (right)	29
Figure 7: Dataset for Bayesian Linear Regression	30
Figure 8: A model of how to fulfill the augmented k-anonymity	33
Figure 9: Determining the centroids by using K-means clustering	34
Figure 10: Example of K-means clustering	37
Figure 11: Architecture of Estonian X-Road	43
Figure 12: Possible Application of the augmented k-anonymity	44
Figure 13: Definition of FM1 (left) and the way of determining $r$ (right)	50
Figure 14: Example of FM2 where $k = 3$	51
Figure 15 (reuse of Figure 5): Example of FM3 where $k = 3$	52
Figure 16: Explanation of model 6, where $k = 50$	52



# Table of Contents

<b>Author’s Declaration</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Annotatsioon</b>	<b>5</b>
<b>Abbreviations and concepts</b>	<b>6</b>
<b>List of Tables</b>	<b>7</b>
<b>List of Figures</b>	<b>8</b>
<b>Table of Contents</b>	<b>9</b>
<b>1. Introduction</b>	<b>11</b>
<b>2. Research Questions</b>	<b>12</b>
<b>3. Related Work</b>	<b>15</b>
3.1. Secure Computation	15
3.1.1. Secure Computation using Homomorphic Encryption	15
3.1.2. Secure Computation using Secret Sharing	16
3.2. Bayesian Approach	18
3.3. Table GAN	19
<b>4. Technological Background</b>	<b>20</b>
4.1. Generative Adversarial Networks	20
4.2. K-anonymity	22
4.3. K-Means Clustering	23
4.3.1. Lloyd’s Algorithm	23
4.4. Attack Model	23
4.5. Kernel Density Estimation (KDE)	25
4.6. Bayesian Linear Regression	26
4.7. No-U-Turn Sampling (NUTS)	27
4.8. Microaggregation (for a comparison target)	27
<b>5. Proposed Concept: Augmented k-anonymity</b>	<b>28</b>
<b>6. Proposed Models and Experimental Results</b>	<b>30</b>
6.1. Datasets and Validation Methods	30
6.1.1. Mixed Gaussian 8 distributions	30
6.1.2. Datasets for Bayesian Linear Regression	30
6.2. Models and Results	32
6.2.1. GAN architecture	32
6.2.2. Summary of Models and Experimental Results	33
6.2.3. Model 1 and Results	34

6.2.4. Model 2 and Results	38
<b>7. Discussion</b>	<b>42</b>
7.1. Discussion	42
7.2. Possible Application	44
<b>8. Conclusion</b>	<b>46</b>
<b>Bibliography</b>	<b>47</b>
<b>Appendix</b>	<b>51</b>

# 1. Introduction

Statistical Disclosure Control (SDC) provides a guideline to protect a data subject based on a that no one should be identified from a released data. SDC contains two basic approaches: rules-based and principles-based [31]. While the rules-based approach determines a dataset is safe for sharing or not according to some hard rules, the principles-based treats such judgment as a matter of probability. Therefore, the principles-based approach reflects rules-of-thumb character and academic techniques. The scope of this paper is this principles-based approach under the same purpose of the SDC. In this scope, plenty of researches have been studied to maintain data utility and keep privacy confidential. Data perturbation, secure computation, and synthesizing data by using Bayesian Networks are prime methods; however, serious limitations and coping methods have also been studied in these traditional studies. For instance, [1] develops a procedure to remove noise (see details in Chapter 3).

In this paper, we propose synthesizing data based on k-anonymity and Generative Adversarial Networks (GAN)<sup>1</sup>. k-anonymity presents an index of how safe a dataset is in terms of privacy leakage. GAN is a deep-learning algorithm and a type of generative models. GAN trains the probability space of original data (we define training data as original data in this paper) and generates data according to the probability space. We demonstrate that generated data remain higher data quality than microaggregation by K-means clustering, that is one of the conventional and similar to our proposed method. The proposition augments k-anonymity, thus this enables data providers to control a balance between data utility and privacy protection level in their generated data when sharing it with thirds party.

---

<sup>1</sup> The code locates on <https://gist.github.com/Hirokiii/6c3fec99a38176f7f7b757b7ca3bed40>

## 2. Research Questions

The main research question of this paper is “how data providers can share their privacy-concerned data in a secure way” in order to establish collaborations with other institutes or business groups more. Sequentially, in order to verify this research question specifically, this divides into the following two sub-research questions.

RQ1: How to evaluate the “secure” level?

RQ2: How much statistically meaningful data can be generated in such a secure way?

Though we divide the main research question into these two questions for convenience, they are not mutually exclusive. When both validities are confirmed, the validity of the main question shall be achieved. We create a model concept: the augmented  $k$ -anonymity that maps RQ1, build six models in total (two success models are in Chapter 6 and the other four in Appendix) and implement them to examine practicality and concept repeatability following to the design science approach [40]. Details are described below.

Let  $k$  for  $k$ -anonymity be a natural number and parameter for privacy protection. For instance, where  $k = 2$ , there exists at least  $k$  records in a dataset, that hold the same information so that the possibility of identifying a data subject will be 50 %. In the verification of RQ1, we introduce a new concept of the augmented  $k$ -anonymity applying the existing  $k$ -anonymity concept. Under this concept, the quality of generated data decreases as  $k$  increases. This would enable the data provider to manipulate a balance in privacy protection and data utility. We build multiple models as a means to implement this concept specifically and verify the realization of the concept using each model and one artificial dataset. The data used at this validation is 8 gaussian distribution used for checking the GAN-specific problem: mode collapse [6].

On the contrary, in the examination of RQ2, the models in previous used remained the same, we sample artificially generated from a specific hierarchical Bayesian model with true values, and this is used as original data for synthesizing. Bayesian Linear Regression analysis will be conducted on the data generated based on this original data and we compare the results with true values. At this validation, if the accuracy of the analysis results decreasing with the increase of

$k$ , the concept will be archived, that is, establishing the protection of privacy and producing the generation data that adheres statistically significant meanings.

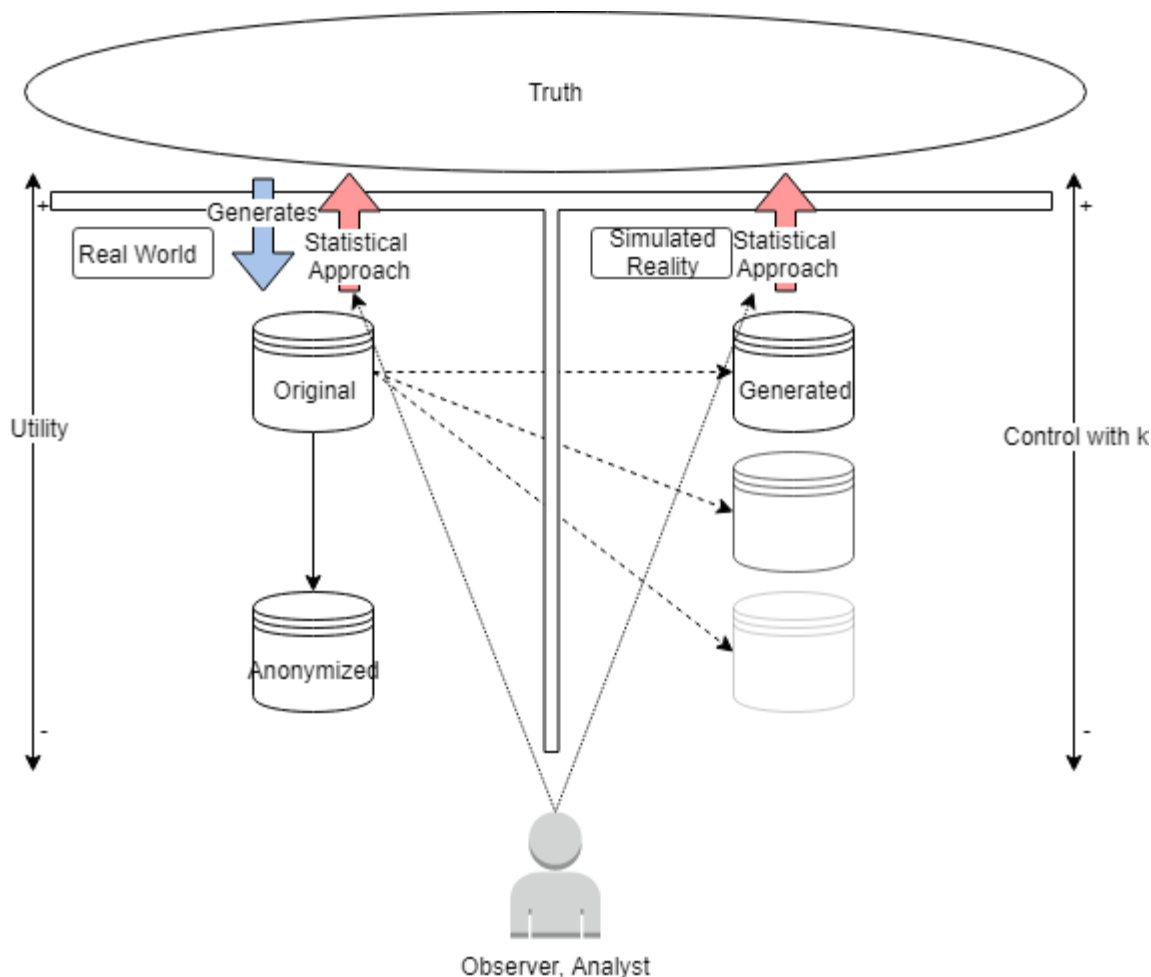


Figure 1: Overall objective of this research

Figure 1 depicts the overall objective of this research. Now, it is assumed that the observer who is the recipient of the generated data by our proposed method is more interested in the truth behind the data than the original data itself. “Truth” is the so-called grand truth, the goal of statistics. By knowing this truth, the observer can predict the future. All data in the real world is assumed that generated by adding noise to these truths, however, it sometimes contains private information and, therefore, this prevents sharing with third parties. In practice, while  $k$ -anonymity, a type of pseudonymization, is broadly used because of its ease to implement, this may cause a serious reduction of the data quality [46]. In this respect, the proposed method would be superior; a combination of  $k$ -anonymity and GAN, we call this augmented

k-anonymity. Using this method would enable to manipulate the quality of the generated data (concurrently the privacy protection level) which could not be achieved conventionally without plenty of parameter settings and model choices (related to RQ1). Furthermore, comparison with truth values is performed to confirm how much quality the generated data remain (related to RQ2) and more quality is maintained than microaggregation by K-means clustering. Moreover, due to the nature of the augmented k-anonymity in the realm of machine learning, we can set  $k$  to a value larger than conventional k-anonymity and microaggregation. Thus, we do not apply Inception Score [3, 33] and Frechet Inception Distance [16] as GAN evaluation metrics, instead, we adopt absolute loss value from the true estimation values (see section 6.1.2).

## 3. Related Work

### 3.1. Secure Computation

Secure computation is that multiple database holders obtain only their calculated results from confidential information on the equal ground while keeping their information secret. At this time, the secret information cannot be known to others.

The concept is exemplified in the example of Millionaire's Problem [44]. Now, Alice and Bob have their own wealth  $x_A$  and  $x_B$  (for simplicity  $x_A \neq x_B$ ). They want to know who is richer without disclosing their amount of wealth. The secure computation results  $y_A$  received by Alice can be defined as the following.

$$y_A = f(x_A, x_B) = \begin{cases} 0, & \text{if } x_A > x_B \\ 1, & \text{otherwise} \end{cases}$$

In the above equation, the case where the function  $f$  for judging larger or smaller is a reliable third party is called an ideal model, and the calculation based on a specific protocol is termed a real model [48]. The following subsections describe how to realize the real model, here must be noted that there exist two considerable limitations of on secure computation. One is operable arithmetic calculation is only addition and multiplication because of the propers of secure computation and the third party can only obtain the calculation results. In short, those restrict what analysts can do with secure computation. Second is this technology does not yet have commercial support, and computing time does not meet the demands of a realistic business [42].

#### 3.1.1. Secure Computation using Homomorphic Encryption

Homomorphic encryption refers to additive homomorphic encryption, multiplicative homomorphic encryption, and fully homomorphic encryption [11, 12, 24] that combines the

properties of both formers. For additive homomorphic encryption, the following formula is completed.

$$Dec(Enc(x_1) \oplus Enc(x_2)) = x_1 + x_2$$

Note that *Dec* means decryption of ciphertext, *Enc* means encryption of plaintext,  $\oplus$  indicates the addition of ciphertexts, and  $+$  means the addition of plaintexts. That is, the result of the addition of ciphertexts is equal to the result of the addition of plaintexts. Elliptic curve ElGamal encryption [9] is a good example of such additive homomorphic encryption. In addition, the multiplication homomorphic cryptosystem is established by converting the operation of the above equation from additive to multiplicative, and RSA cryptosystem [32] is this type of encryption. Moreover, fully homomorphic encryption satisfies both additive and multiplicative formulas at the same time[11, 12].

In the case of secure computation using homomorphic encryption, even if data is leaked from the server by any chance, since the data is encrypted, the contents of the data will not be clarified unless the decryption key is also leaked [47].

### 3.1.2. Secure Computation using Secret Sharing

First of all, we start with an introduction of the explanation of the  $(k, n)$  - threshold secret sharing scheme [34]. Let  $f(x)$  be a certain function of  $k - 1$  dimension and  $f(0)$  value be what we want secret. In the beginning, we define  $n$  points on  $f(x)$  with an algorithm called a *deal*. In that time, the original  $f(x)$  cannot be clarified from  $k - 1$  points, and  $f(x)$  is uniquely determined only after  $k$  points are obtained. Consider  $f(x) = 2x + 1$  of a linear function as a simple example. The value you want to keep secret is  $f(0) = 1$ . At this moment,  $f(x)$  cannot be uniquely obtained from  $(x, y) = (1, 3)$  on  $f(x)$ , but when another point  $(-1, 1)$  is given,  $f(x)$  can be obtained, and the secret value  $f(0) = 1$  can be exposed. That is, the linear function utilizes the property that is uniquely determined when two points are given.

Next is about  $(m, m)$  - threshold secret sharing scheme [10] by additive share and secure computation by secret sharing.



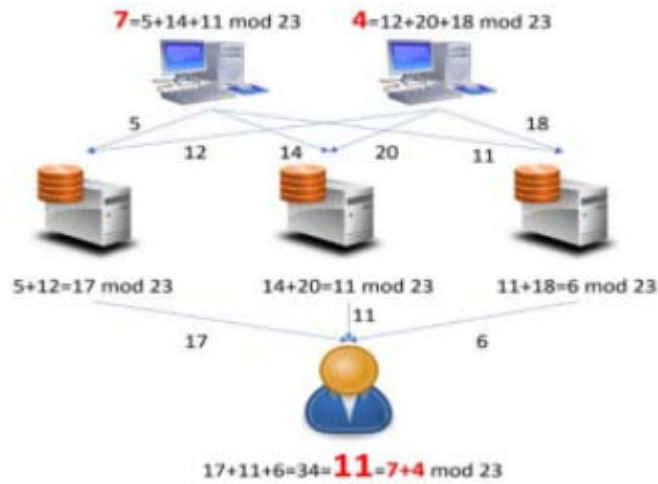


Figure 2: Example of (3,3)-threshold secret sharing

Let  $Z/qZ$  be the remainder set  $0, 1, 2, \dots, q - 1$  when the element of the integer set  $Z$  is divided by  $q$ . If  $q = 23$  as shown in Figure 2, the values 7 and 4 you want to keep secret are  $Z/23Z$ . Assuming that  $m = 3$ ,

$$7 = 5 + 14 + 11 \pmod{23}$$

$$4 = 12 + 20 + 18 \pmod{23}$$

are established. The secret values 7 and 4 are distributed into 3 elements by the additive share.  $\pmod{q}$  represents that the remainder when the left side and the right side of the expression are divided by  $q$  is equal. At this time, the secret value 7 cannot be restored from two values of the values 5, 14, 11. Up to this point is the explanation of the  $(m, m)$ -threshold secret sharing scheme.

Move on to the explanation of secure computation by secret sharing using this example. One share is taken out from the share of each secret value and the added value in  $\pmod{23}$ . In the example, this is the value 17 obtained dividing 17 by  $q = 23$ , which is the sum of 5 (out of the share of 7) and 12 (out of the share of 4). Let his group (5, 12) be called a party and we can have  $m$  parties in total. Finally, by dividing the sum of the values of each party by  $q$ , that value is equal to the sum of secret values (see the last line in Figure 2). That is, using secret sharing, it is possible to acquire the operation value of secure computation without leaking secret values to the outside.

In the case of secure computation using secret sharing, even if the data leaks from the server by chance, if the leaked server is less  $m-1$ , the contents of the data will not be clarified.

## 3.2. Bayesian Approach

There is a bunch of research taking Bayesian Networks to create synthetic data for the purpose of remaining original data confidential [45]. Some researchers suggest that using a hierarchical Bayesian approach to generate data so that such synthetic data will be more robust to the general model or network mis-specification [45]. One of the advantages of using Bayesian methods for synthetic data is its accountability. Graphical models and each parameter can clearly explain the original data structure and why data provider choose a particular model, however, this can be the considerable vulnerability. Nevertheless, the graphical models illustrate relationships of each node, such allows do not denote causal relationships [45]. Moreover, the Bayesian approach demand designers to select a suitable model and a prior distribution, but this can be arbitrary. Arbitrarily selecting a particular model and a prior means choosing no other model, which is acceptable for certain analysis algorithms but does not take other data analysis approaches into account (obviously Bayesian analysis should work well). Even if any other approach could be taken into consideration and a sophisticated model could be formed, then the requests for publishing generated data would be diminished and only such a universal model would be demanded. However, in reality, it will only happen to very specific datasets. On the other hand, our proposed methods have significant advantages for these problems. First of all, a data provider would not be required pre-knowledge, choosing several parameters or priors for instance. Though the network needs to be scaled to the shape of the original data, this can be automated. Furthermore, prior modeling of the data structure is unnecessary. This is equivalent to not being able to explain the distribution of the generated data, but this should be no problem because even the true distribution of the original data is unknown to anyone. Rather, arbitrary decision making about original data should be avoided as it would be a source of information for attackers. Not only these, but the proposed method can also control the quality of the generated data, and gives a logical explanation of the quality-degree. Although it will be described later, there is a possibility that it may be identified as generated data being too similar to original one even though the generated data does not have a one-to-one correspondence with the original data anymore. In practice, it is better for the data provider to enable to control the balance between data quality and privacy rather than being able to create unlimited high-quality generated data.

### 3.3. Table GAN

With the same purpose of our research, there is a so-called table GAN [28] in the same field of the research topic using GAN. The table GAN adds the *information loss* and the *classification loss* to the loss functions and defines a third neural network called a *classifier* for deriving the classification loss. The classifier is trained in advance using the original data and its label to learn the correlation between the label and the attributes of the original data. In addition, obviously, it prevents the generation data incompatible with the label. The information loss function inputs the previous tensor via the discriminator sigmoid function in the output layer to indicate that the difference between the mean and the standard deviation will be zero between the original and the generated data.

As stated in the paper [28], the original Discriminator also learns to some extent semantic information such as mean and standard deviation. Therefore, their information loss and classification loss are merely aimed at improving the quality of the generated data. In their dissertation, it is claimed without the reassurance that re-identification attack and attribute disclosure cannot occur because the generated data by nature do not have a one-to-one correspondence with the original data. Then, based on the premise, they simply attempt to improve the quality of the generated data; however, we pose a question. Although the distance to the closest record (DCR) between the original and the generated data is used as their evaluation method [28], if the generated data is high quality enough and any value is less than  $\epsilon$  to the original, by means of the nearest neighbor method, it would become possible to narrow down or identify a target data subject. In other words, if the attacker knows the value for a certain attribute of the target person, the other attribute values may be exposed, or it might lead to the problem of a false light to other people with possessing values close to the target person.

Also, their experiments do not mention the GAN-specific problem: mode collapse. The data they use is actual data, but even raw data is the result of adding noise to the true model, hence it is more desirable to compare data using true value and generated data instead of original and generated data. The important thing is to attach a privacy protection function that can control to generated data and verify that the analysis result of the generated data is meaningful. Our proposed method satisfy these points.

## 4. Technological Background

### 4.1. Generative Adversarial Networks

Ian Goodfellow et al. invent Generative Adversarial Networks (GAN) in 2014 [13]. This belongs to the generative model, which estimates the training data distribution in its learning process and generates samples based on such probability distribution after the learning. GAN consists of two different neural networks: a generative model  $G$  and a discriminative model  $D$ .  $D$  inputs original data  $x$  and generated data by  $G$  one after the other.  $D$  learns the original data distribution and generated data distribution in order to classify its inputs into original one or generated one. On the other hand,  $G$  approximates its distribution to that of the original data, in other words,  $G$  aims to deceive  $D$ 's judgment.

Let  $p_{data}$  be the distribution over the original data.  $G$  inputs noise  $z$  from a variable distribution (usually this is uniform distribution or Gaussian distribution)  $p_z(z)$  and maps to a data space  $G(z; \theta_g)$ , where  $p_g$  represents its distribution. In the same way,  $D(x; \theta_d)$  denotes the data space of the discriminator, which yields a single scalar that can be a probability. Hence,  $D(x)$  expresses the probability that the input  $x$  pertains to  $p_{data}$ , not  $p_g$ . In training of GAN,  $D$  attempts to maximize its  $D(x)$  and, simultaneously,  $G$  search the best  $\theta_g$  to minimize  $(1 - D(G(z)))$ . For convenience sake of computing, GAN adopts  $\log D(x)$  and  $\log (1 - D(G(z)))$  as its value function  $V(G, D)$ . In summary,  $D$  and  $G$  play the min-max game according to the following equation.

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

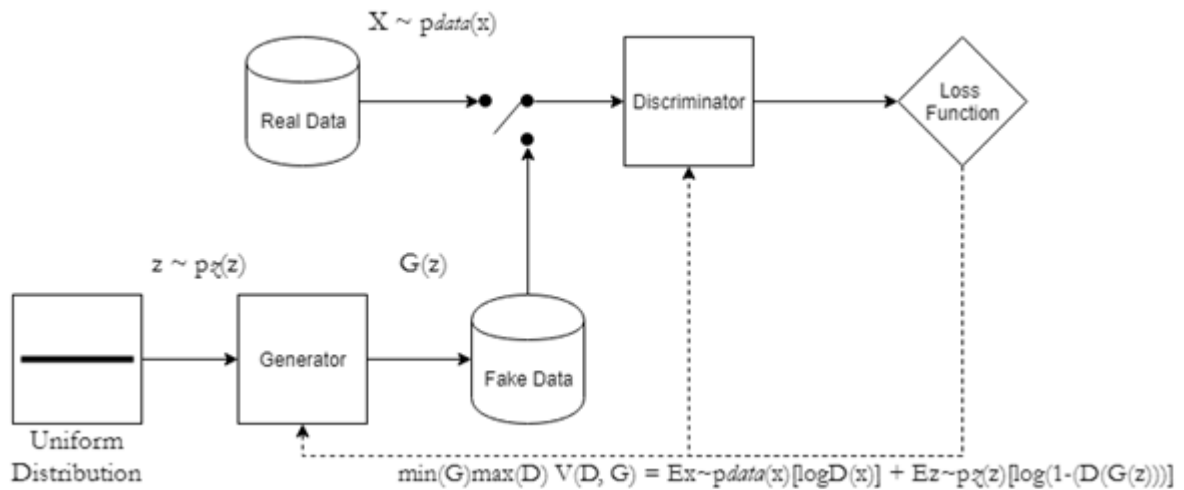


Figure 3: The architecture of GAN

While GAN has dramatically evolved since it appeared in 2014 and is archiving recent successes, it is extremely difficult to be trained. A massive amount of research has been conducted and its evolution can be grouped into three categories.

The first is the mathematical development of the loss function used in GAN. The original GAN uses Kullback-Leibler (KL) divergence [22], but in the f-GAN's paper [27], it recaptured it in a more general variational divergence and found that KL divergence is just one particular case. Furthermore, the emergence of Wasserstein GAN (WGAN) [2] can be mentioned as a significant turning point. In WGAN, earth mover distance is adopted as loss function. As a result, although it has been proved that it does not fall into the mode collapse [26] logically and it converges to the global optimization point as the learning time passes, it is known that the actual product becomes blurred one. Based on these, improved WGAN (or WGAN-gp) [14] gives the gravity penalty and evolved to a higher degree.

The second is the development of GAN's learning methods. In GAN, a type of over-fitting called mode collapse is well known. The generator does not generate a specific probability space when it falls into mode collapse. One of the causes of this is that Discriminator is too powerful to classify the initial generated data so that Generator fails to learn efficiently. In order to solve this, Unrolled GAN [25] learns Generator based on the behavior of Discriminator several steps ahead. VEEGAN [36] tries to reduce generated data to noise space.

The third is about how to handle high-resolution images. Style GAN [41] learns sequentially from those with relatively low resolution called Progressive-Growing. In addition, Style GAN enables to change each style of the image generated by using the Mapping Network. The generated image level of Big GAN[5] has resulted in quality that is no longer distinguishable by the human eye.

In our experiments, we use Unrolled GAN [25]. Unrolled GAN updates  $G$  after several (usually five) times learning of  $D$ , and updates  $D$ . This is because, in general, especially at the beginning,  $D$  easily distinguishes its input so that  $G$  struggles in gaining useful feedbacks. Unrolled GAN achieves efficient learning by providing  $G$  with a little advanced future of  $D$ .

## 4.2. K-anonymity

Table 1: Example of k-anonymity, where  $k = 2$ , ID is deleted, and QID = {ZIP, Birth, Gender}

ID	ZIP	Birth	Gender	Salary
/	12345	The 1980s	M	5,700
/	12345	The 1980s	M	900
/	67890	The 1990s	F	3,000
/	67890	The 1990s	F	1,600
/	12345	The 1980s	M	2,100

The original paper on k-anonymity [39] focused on how to prevent the re-identification attack by linking multiple datasets. Re-identification attack by linking occurs when a target dataset, where the identifier or ID such as name is deleted, gets joint other datasets. Attributes used across such several datasets are called quasi-identifier or QID. ZIP-code, birth date, gender can be QID. That is, QID is information that makes it possible to identify an individual like an ID by combining a plurality of QIDs. According to the study [38], experiments on 1990 U.S. Census summary data, 87% of the population can be uniquely determined only by 5-digit ZIP, birth date, and gender. This research discovers that it is not sufficient to just delete ID when sharing data safely. k-anonymity provides a privacy protection indication of how safe the dataset is. Specifically, a

dataset is said to satisfy  $k$ -anonymity where at least  $k$  rows or records possess the same values in all QID. In other words, attackers can narrow down the individual only to  $k$  people.

### 4.3. K-Means Clustering

K-Means clustering is one of the most popular algorithms in the field of data science. It is classified to unsupervised learning. Given an input vector  $X := \{x|x_1, \dots, x_n\}$ , the goal of K-means clustering is classifying the input  $X$  into  $K$  different groups attempting to minimize the following equation [15].

$$J(X, C) = \sum_{i \in n} \|x_i - c(a(i))\|^2$$

$C$  is a set of clusters and  $a(i)$  means an index of assigned the closest centroid for each  $i$ , that is  $c(a(i))$  indicates that the assigned closest centroid for each  $i$ . Equation illustrates that minimizing the Euclidean distances of each of the  $K$ -cluster centroids and the  $n$  data points in  $d$  dimensions.

#### 4.3.1. Lloyd's Algorithm

The most common algorithm to achieve the concept of K-means clustering was expressed by Lloyd in 1982 [23]. His algorithm is composed of three steps [15].

Step 1: initializing the  $k$  centroids

Step 2: Until the algorithm converges:

Step 2(a): assigning each  $n$  point to its closest cluster centroid

Step 2(b): shifting each centroid to the mean of its assigned centroid

### 4.4. Attack Model

Generally, cryptanalysis or attackers are aiming to detect a private key, then they can decrypt any ciphertexts. In the context of our research, we can assume that the original data, which we want to keep it confidential, equal to plaintexts and the generated data corresponds to ciphertexts. This assumption implies that GAN serves as an encryption system. As of the time of this writing, no

one could succeed in understanding deep learning completely, in any single datum flow, not in the meaning of meta-flow, so that it is impossible to reproduce the original data from the generated data. Thus, it is better to presume that GAN, in our research, is a one-way function such as the hash function rather than an encryption system because encryption must be a set with decryptability.

Basic attack models [7] are for assessing a cryptographic system under some certain circumstances (models). Though we make a premise that GAN can be associated with a one-way function, adopting the attack model to consider the privacy security of the proposed techniques is supremely beneficial. In line with this purpose, we replace the attack target: from detecting the private key to identifying a data subject. Hence, our goal is that attackers cannot identify any specific data subject from the generated data in any following model.

- Ciphertext Only Attack (COA)

The most “weak” model of attack. Attackers can only access to ciphertexts. In the context of our research, we can assume that the attackers can only obtain the generated data. This is the most occasional case, however, this model is not sufficient when we suppose security of data against attacks.

- Know Plaintext Attack (KPA)

In this model, plaintexts are also accessible to attackers. Even though the attackers are given a certain number of pairs of plaintexts and its ciphertexts, they cannot choose arbitrary sets. In our data-sharing situation, one can access a limited pair of original data and generated data, but not the GAN system.

- Chosen Plaintext Attack (CPA)

Attackers become more “powerful” in this model. They can choose any plaintext to be encrypted and gain those ciphertexts (access to the encryption method). In our case, they can utilize the GAN system freely or request to obtain any generated data unlimitedly while they own original data.

- Chosen Ciphertext Attack (CCA)



The most “strong” model among these four models. Now attackers can choose any ciphertext arbitrarily and access to its corresponding plaintext (access to the decryption methods). Namely, attackers can obtain any pair of original data and generated data.

All in all, CCA is the most dangerous case for all the above models. Precisely, in our research, breaching any original data is the must-avoid thing. For that reason, CCA might signify no meaning to assume privacy safety. Therefore, as a matter of constructive discussion, we modify the CCA a bit: even though attackers gain the whole or a part of the original data, they cannot identify the original data subject from the generated data. In other words, there are no one-to-one correspondence between the original and generated data and no other information that allows connecting a generated datum with its data subject. This is our fundamental of the following discussions.

#### 4.5. Kernel Density Estimation (KDE)

Kernel Density Estimation (KDE) is a non-parametric method that estimates a Probability Density Function (pdf) from sample data. KDE is an extension to Parzen Window Method [29].

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n W\left(\frac{x - x_i}{h}\right)$$

where  $W(x)$  output 1 if maximum absolute value of  $x^{(i)}$  is less than  $1/2$ , otherwise output 0 and parameter  $h$  is called bandwidth which controls how much smooth the estimated pdf will be, still, this method does not cope with sample data continuously. KDE solves this problem. Let  $K(x)$  be a Kernel function which satisfies the following two equations.

$$\int_D K(x)dx = 1$$

$$K(x) \geq 0, \forall x \in D$$

In our experiments, we use the Gaussian Kernel.

## 4.6. Bayesian Linear Regression

The Bayesian statistics are in accordance with Bayesian rules derived from the equation transformation of joint probability. The Bayesian methods are increasingly adopted as an academic approach [21]. A posteriori  $P(\theta | D)$  after obtaining the data is a product of a likelihood  $P(D | \theta)$  and a priori  $P(\theta)$ .

$$P(\theta | D) = \frac{P(D | \theta) P(\theta)}{P(D)}$$

While frequentists seek for fixed values in their approach, Bayesians employ uncertainty distributions. Both of them have advantages, however, this time we use Bayesian for the purpose of performing analysis that is compatible with machine learning, because of the feature of sequential learning from data, and more sophisticated than mere linear regression analysis to examine the generated data quality that can withstand higher analysis.

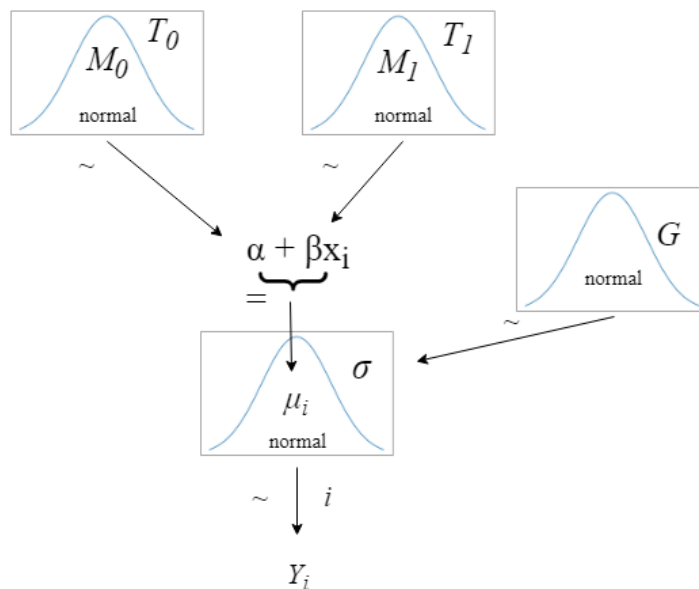


Figure 4: Bayesian Linear Regression Model

Regression analysis is to estimate a model that determines the relation of input and output based on observation data. In Bayesian regression analysis, each coefficient is treated probabilistically.

For instance of Figure 4, the output is determined stochastically based on the normal distribution, and the parameters of the normal distribution, the mean and the variance are probabilistically determined from different normal distributions.

#### 4.7. No-U-Turn Sampling (NUTS)

No-U-Turn Sampler (NUTS) [17] is a Markov chain Monte Carlo (MCMC) algorithm and an extension to Hamiltonian Monte Carlo (HMC). HMC is one of the well known and high-performance MCMC algorithms, however, its operation accuracy considerably depends on two user setting parameters: the number of steps  $L$  and the step size  $\epsilon$ . Pretty small  $L$  causes inexpedient random walk behavior and overlarge  $L$  squanders computation [17]. NUTS does not require the choice of the  $L$  and automates tuning the  $\epsilon$ , but remain the same performance level of HMC. In the experiments, we employ the moduled NUTS in PyMC3 [30].

#### 4.8. Microaggregation (for a comparison target)

Microaggregation is one of the perturbative methods for secure data-sharing. Microaggregation operation consists of two steps: Partition and Aggregation [8]. Partition is a clustering method that categorizes original data into several groups. Aggregation extracts a representative value of data belonging to one cluster (for instance, centroid) and replace each of the belonging data with that values. Therefore, due to these properties, microaggregation is compatible with k-anonymity.

As a comparison target in the experiment, we adopt microaggregation by K-means clustering [19] that is most similar to our proposed method among the existing methods (see Chapter 6 for detail of our method). For the purpose of a control experiment, we simplify the method proposed by [19] and each cluster by K-means clustering does not necessarily have  $k$  data (see Chapter 5, Limitation).

## 5. Proposed Concept: Augmented k-anonymity

The k-anonymity is a protection model of privacy, introduced by Sweeney [39], that modifies the contents of a dataset to satisfy k-anonymity, that is, there are  $k$  different rows that contain the same values for its set of quasi-identifier. Hence, though attackers can narrow down  $k$  person's data, they cannot uniquely identify one data owner. In the case of  $k = 2$ , the probability to succeed in identifying the correct data owner will be 50%. Consequently, the larger the  $k$  become, the more obscure the data will be. That is to say, the closer the  $k$  is to 1, the more similar the k-anonymised dataset will be to the original dataset. When a data provider does care about the privacy of data subjects, the  $k$  should be large enough to protect privacy.

We advocate a concept of the augmented k-anonymity as the main proposal to address RQs. The augmented k-anonymity inherits the above original k-anonymity, however, it is supposed to be used in a GAN's training process as an additional loss function.

### Definition. Augmented k-anonymity

Let  $ORI(x_1 \dots x_n)$  be original data and  $GEN(y_1, \dots, y_n)$  be generated data.  $GEN$  is said to satisfy augmented k-anonymity if  $\forall y$  in  $GEN$  has  $k$  different points of  $ORI$  at equal distance.



Figure 5: Example of the augmented k-anonymity, where  $k = 3$

**Example. Figure 5 adhering to augmented k-anonymity**

Figure 5 illustrates the concept of the augmented k-anonymity where  $k = 3$ .  $d := \{d_1, d_2, d_3 | d_1 < d_2 < d_3\}$  represents a distance between a target generated datum (blue dot in “Original Data”) and  $k$  nearest neighbors in original data. Augmented k-anonymity is achieved where  $d_1 = d_2 = d_3$ , so that the target generated datum equally correspond to  $k$  different original data. No more one-to-one correspondence exists between the original data and the generated data.

**Limitation. Augmented k-anonymity.**

Let  $D$  be the number of dimension of data. If exactly each generated datum satisfies augmented k-anonymity, the generated data distribution can be uniquely determined, where  $k = D + 1$ , patterned, where  $k < D + 1$ , or impossible to exist, where  $k > D + 1$ . If it will be uniquely determined or patterned, these features could be a strong clue to regenerate the original data and the risk of privacy breach will soar. Therefore, the augmented k-anonymity aims to satisfy the augmented k-anonymity definition on an average scale or as the target of an additional loss function in GAN.

## 6. Proposed Models and Experimental Results

### 6.1. Datasets and Validation Methods

#### 6.1.1. Mixed Gaussian 8 distributions

We use the mixed Gaussian 8 distributions as the first synthetic dataset according to the convention in the realm of research of GAN and RQ1. This dataset is created, especially, for confirming whether the GAN occurs the mode collapse or not. To check this, KDE illustrates the results. The confirmation is performed by visual observation. The size is 1000. The number of attributes is 2 and that of clusters is 8.

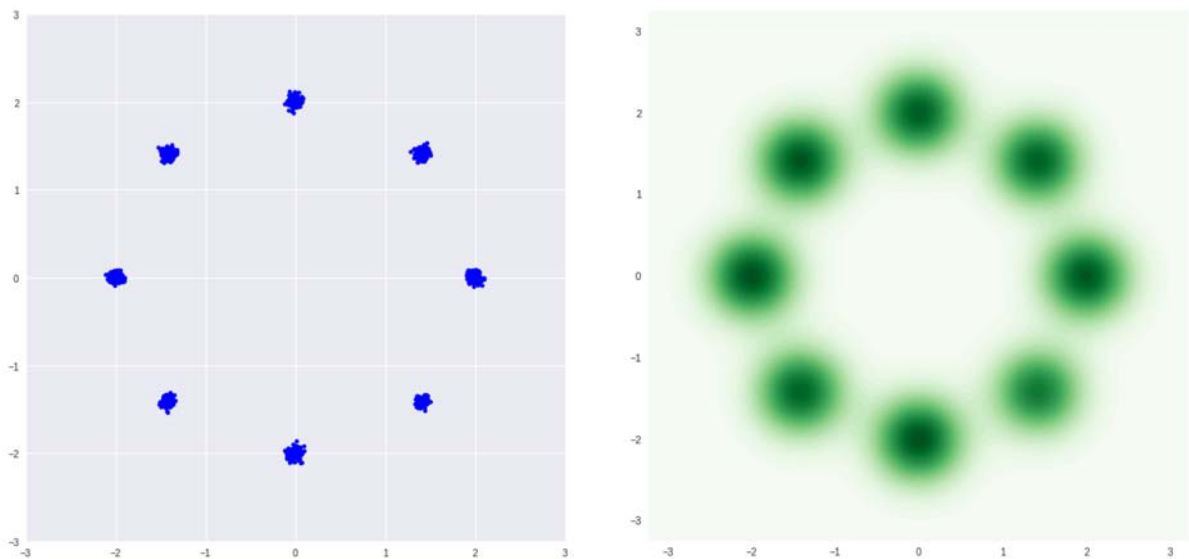


Figure 6: The mixed Gaussian 8 distributions dataset (left) and its KDE result (right)

#### 6.1.2. Datasets for Bayesian Linear Regression

We perform validation experiments under the RQ2: “How much similar to the original data the generated data is?”, and to answer to this RQ2 in a measurable way, we replace it with “How much data quality the generated data remain?”. As a data model, we adopt a Bayesian Linear

Regression model and apply the same dataset used in the PyMC3 official website<sup>2</sup> (but the data size is different for the purpose of data quality). The generated data is examined in No-U-Turn sampler (NUTS) [17] in the PyMC3 library.

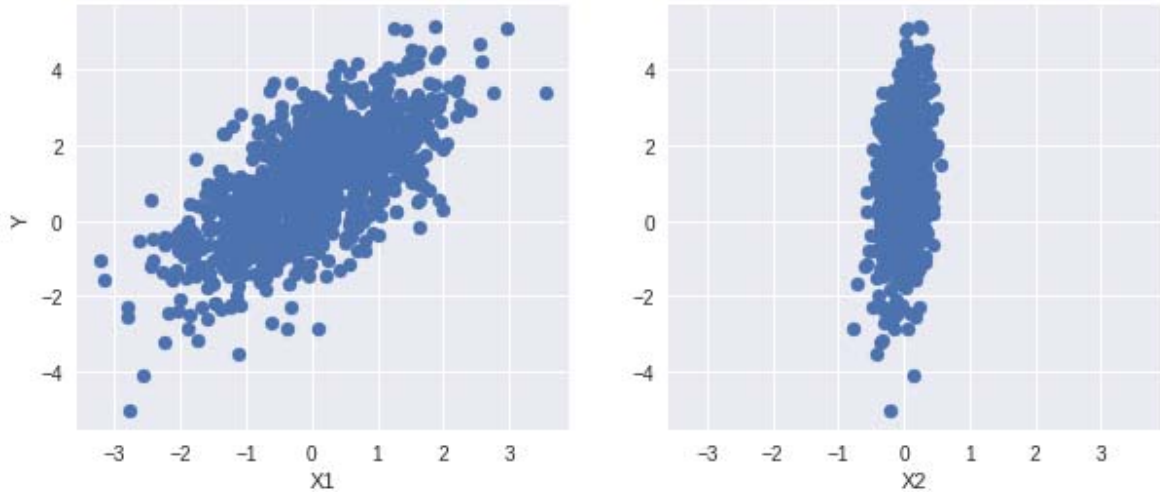


Figure 7: Dataset for Bayesian Linear Regression

The size is 1000 and the number of attributes is 3. The  $Y$  is generated on the basis of the following formula:

$$Y \sim N(\mu, \delta^2)$$

$$\mu = \alpha + \beta_1 X_1 + \beta_2 X_2$$

$Y$  is deterministic random variables according to the Gaussian distribution, where mean is  $\mu$  and variable  $\delta^2$ .  $\mu$  is a linear function with random variables  $X_i$ , where  $\alpha$  is the intercept,  $\beta_i$  is the coefficient of  $X_i$ . As prior for Bayesian Linear Regression, we assume the following stochastic distributions.  $|N(0, 1)|$  indicates non-negative Gaussian distribution.

$$\alpha \sim N(0, 100)$$

$$\beta_i \sim N(0, 100)$$

$$\delta \sim |N(0, 1)|$$

The true values for the estimation  $\alpha, \beta_1, \beta_2, \delta$  are 1, 1, 2.5, 1 respectively (see Table 2). Therefore, the assumption on the prior for  $\alpha$  and  $\beta_i$  is quite poor, less information.

<sup>2</sup> [https://docs.pymc.io/notebooks/getting\\_started.html](https://docs.pymc.io/notebooks/getting_started.html)

Table 2: The true values for the estimation

Variables	True Value
$\alpha$	1
$\beta_1$	1
$\beta_2$	2.5
$\delta$	1

We define the sum of the absolute value error between the true values and estimated values as a *score*. Thus, the smaller the *score*, the closer the estimated values are to the true values.

$$score = |\alpha - \hat{\alpha}| + |\beta_1 - \hat{\beta}_1| + |\beta_2 - \hat{\beta}_2| + |\delta - \hat{\delta}|$$

## 6.2. Models and Results

### 6.2.1. GAN architecture

Table 3: Information on GAN architecture

Batch size	100
Number of epoch	10000
Generator's input	Uniform distribution([-1, 1], shape=(Batch size, 10))
The type of GAN	Unrolled GAN
Unrolled steps	5
Generator Nodes	10 > 32 > 64 > Number of attributes of original data
Discriminator Nodes	Number of attributes of original data > 64 > 32 > 1
Batch Normalization & Drop Out	None of these are used



Activation Function	At the last layer of Generator: tanh, At the last layer of Discriminator: sigmoid, The other: Leaky ReLU [43]
---------------------	---

Batch size is relatively larger than usual against the dataset ( $N=1,000$ ). This value is the one that performed best throughout the experiments. What can be inferred from this value is that the larger the batch size, the more accurate it can be grasped as a whole feature of original data, which seems to be better when trying to control the generated data quality as a whole such as this research purpose. The reason for using the uniform distribution instead of the Gaussian distribution for the generator's noise generation is also because the uniform distribution has better results throughout the experiment. The uniform distribution seems to be well reflected other than the main features such as outliers of the original data. We put some results at the end of the volume, but with WGAN, the control of generated data quality by  $k$  does not work. This time only one type of GAN: Unrolled GAN has been tried. The number of unrolled steps followed the convention. Since the data size is small this time, the input dimension of the generator is dropped to 10, and the layers of the generator and discriminator, and the number of nodes is not large. Though it is reported that batch normalization [18] in Generator and dropout [37] in Discriminator prevent deep learning from overfitting and then they are beneficial to improve the quality of generated picture, in this case, the data sets are relatively small, and these methods are too powerful to learn only some features, so they are not used.

### 6.2.2. Summary of Models and Experimental Results

Both Model 1 and Model 2 with Gaussian datasets perform as planned: we succeed in causing mode collapse by aiming at increasing  $k$ . In the experiments using datasets for Bayesian Linear Regression, the generated data utility control by  $k$  is not achieved as planned in Model 1, while it realizes in the modified Model 2. As a result of the comparative experiments, we can confirm that our proposed method can preserve higher data utility and select larger privacy parameter  $k$  than the existing method.

### 6.2.3. Model 1 and Results

In Model 1, fixed points based on the features of the original data are derived, and the goal of training GAN is to close the distance between the fixed points and the generated data to zero.

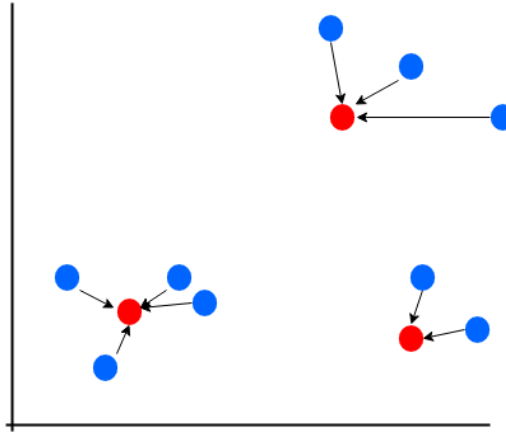


Figure 8: A model of how to fulfill the augmented k-anonymity. The red points are the fixed points and the blue points are the generated data.

Figure 8 demonstrates Model 1. Blue points represent the generated data. Red points are fixed points called *centroid(s)*. Now, there are 9 generated data and 3 centroids. GAN train its generated data distribution to perform that the distance between each generated datum and its nearest neighbor centroid will close to 0. As noted in the Limitation in the previous section, even though the number of the target generated data at each centroid differ, its average is 3 (the number of generated data divided by the number of centroids) at the present case. Thus, Figure 8 can be said to satisfy the augmented k-anonymity, where  $k = 3$ . When learning, the expected value of these distances is adopted as GAN's additional loss function.

How can we determine such centroids with considering reflecting the original k-anonymity concept? We propose applying an unsupervised machine learning algorithm: K-means clustering.

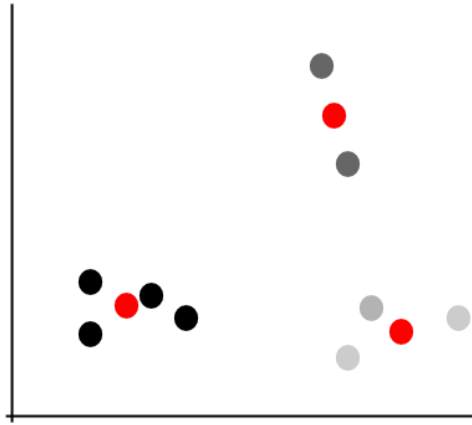


Figure 9: Determining the centroids by using K-means clustering

Grayscale points in Figure 9 depict the original data and they are classified into 3 by K-means clustering where  $K = 3$ . Red points are the centroid of each class.  $K$  (for K-means clustering) is determined by the length of original data and  $k$  (for the k-anonymity), that is, simply

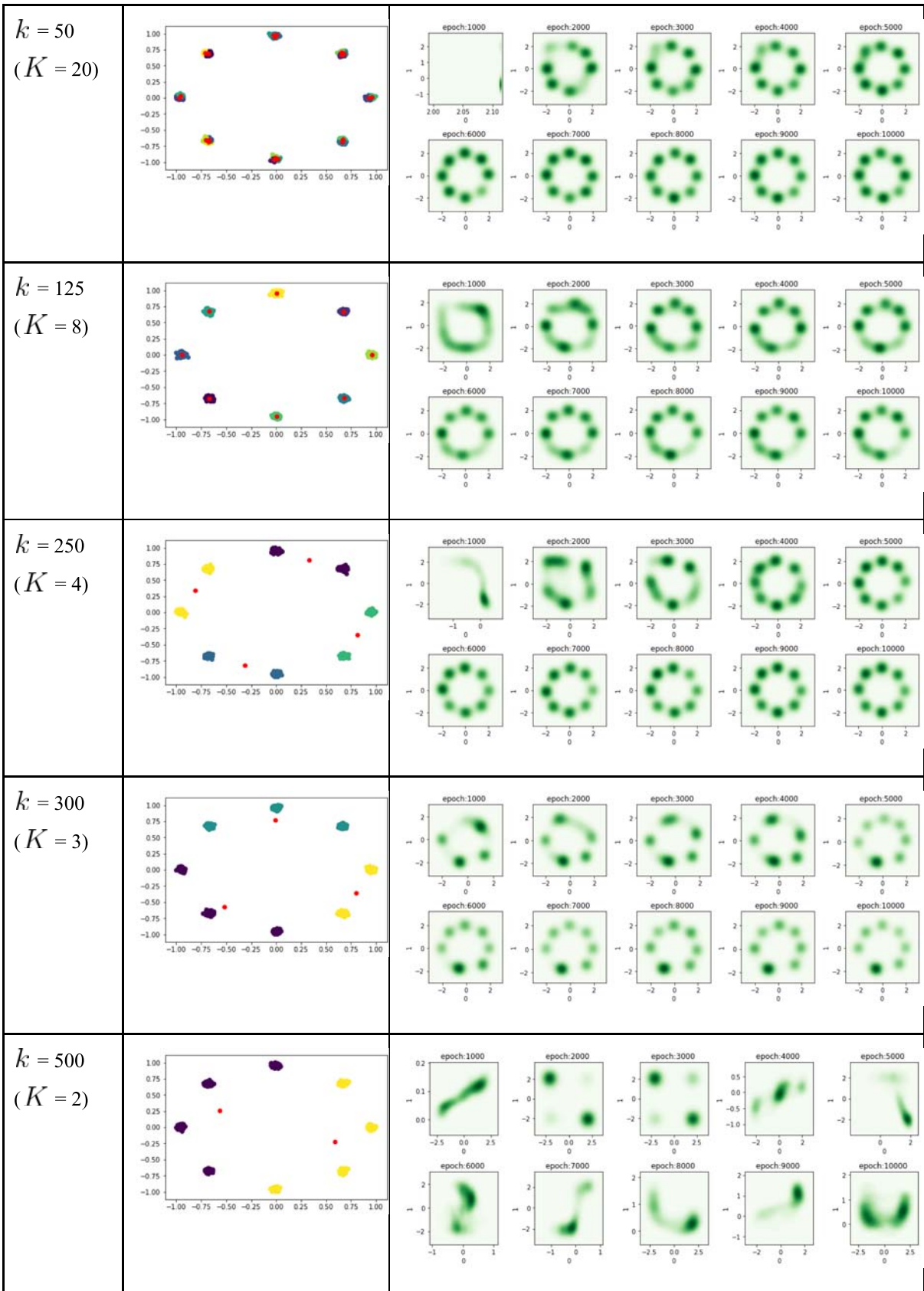
$$K = \text{length}(\text{original data}) // k$$

where  $K$  and  $k$  are natural numbers,  $//$  represents integer division.

## Results

Table 4: Results of Model 1 with the mixed Gaussian 8 distributions dataset

Values of $k$ (and $K$ )	Distributions of original data and centroids	KDE results
$k = 2$ ( $K = 500$ )		



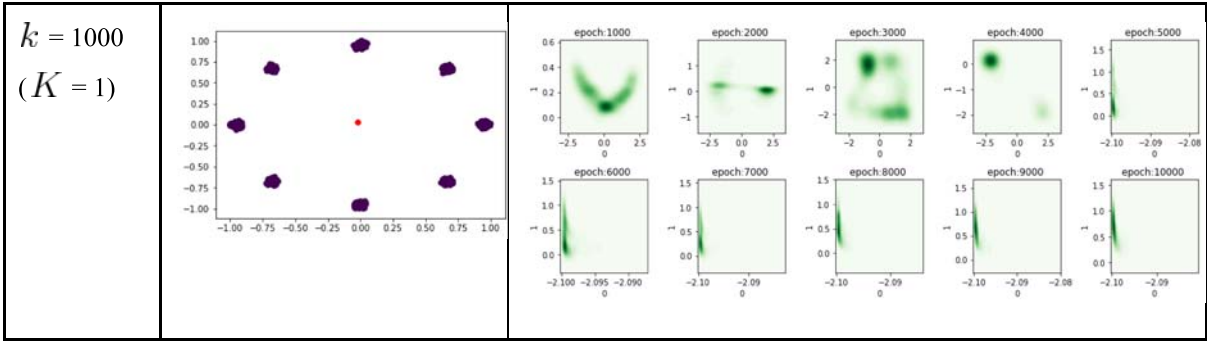


Table 5: Results of Model 1 with the dataset for Bayesian Linear Regression

	Truth	sample	$k = 0$	$k = 2$	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
alpha	1	0.906	1.054	0.305	0.508	0.974	0.971	1.229	0.892
beta1	1	0.948	0.852	0.732	1.175	1.935	1.174	1.38	1.128
beta2	2.5	2.607	2.854	3.432	3.902	0.041	3.024	3.632	1.857
sigma	1	0.962	0.875	0.694	1.071	0.852	1.374	1.197	0.692
score	0	0.291	0.681	<b>2.201</b>	<b>2.14</b>	<b>3.568</b>	<b>1.101</b>	<b>1.938</b>	<b>1.187</b>

Considering the concept of our proposed method; the larger the  $k$  become, the more obscure the data will be, the results with the mixed Gaussian 8 distributions dataset provide the expected results, that is, we can occur the mode collapse where  $k$  is relatively larger. In this time, because of the clear partitions of this dataset, we can assume that the mode collapse is controlled up to  $K = 8$ , however, the reason why where  $K = 4$  the mode collapse does not happen remains unknown. On the other hand, the results with the dataset for Bayesian Linear Regression disclose that the concept is not achieved.

### Possible Defect

Figure 10 exhibits the results of K-means clustering on random sample data (N=1000).  $K$  corresponds to  $k$  in Table 6. Table 6 explains the expected value (Mean) and maximum value (Max) of the distance between each datum and its nearest centroid that both belong to the same cluster after the K-means clustering. With this data, we can confirm that the Mean is always smaller than Max at each  $k$  and there is not much difference in Mean per  $k$ .

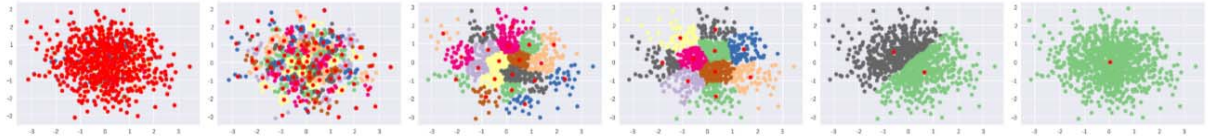


Figure 10: Example of K-means clustering

Table 6: Mean and maximum value of the distance between each datum and its nearest centroid that belong to the same cluster on the data in Figure 10

$k = 2$	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
Mean					
0.0275	0.1424	0.3576	0.5048	1.0172	1.2553
Max					
0.0933	0.5307	1.5879	2.3044	3.1596	3.9296

Table 7, on the dataset for Bayesian Linear Regression, presents the same properties in Table 6. For this reason, it is hypothesized that learning of GAN may not be stabilized due to additional loss value being too small.

Table 7: Mean and maximum value of the distance between each datum and its nearest centroid that belong to the same cluster on the dataset for Bayesian Linear Regression

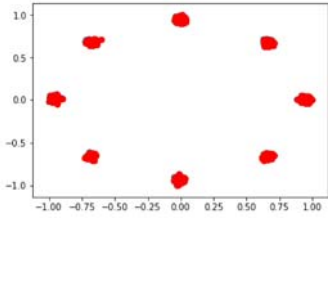
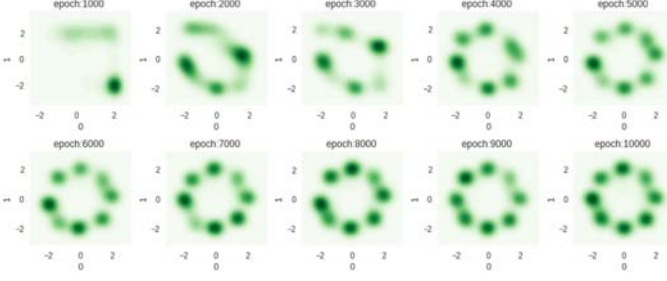
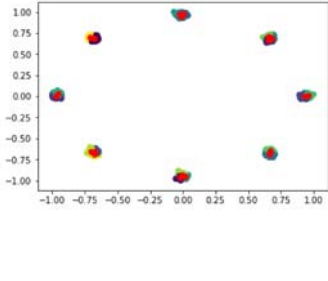
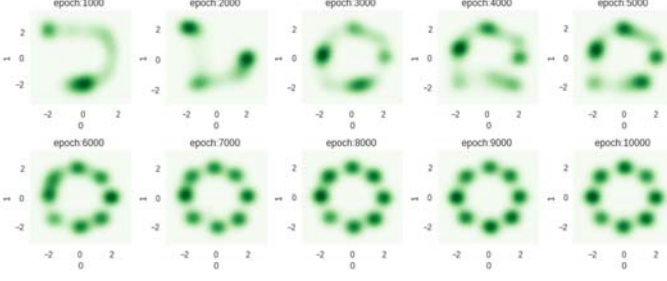
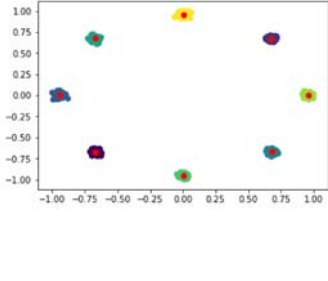
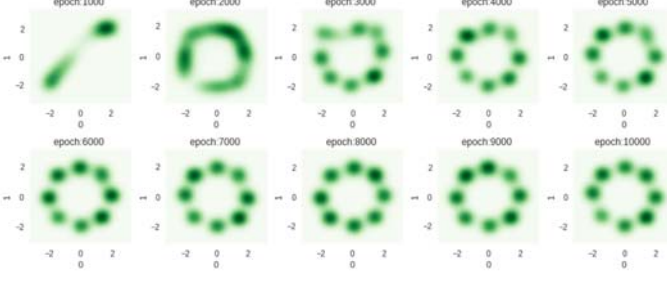
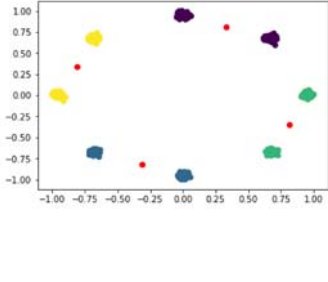
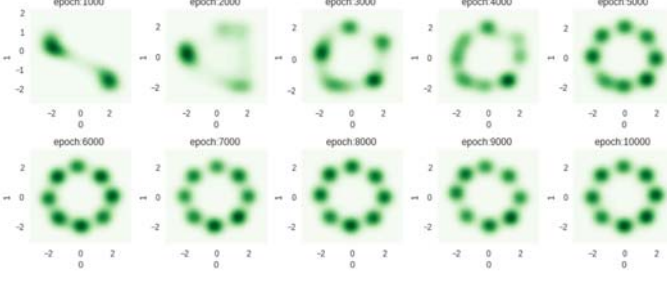
$k = 2$	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
Mean					
0.0250	0.0937	0.1741	0.2179	0.3596	0.4492
Max					
0.0777	0.2731	0.6836	0.8772	1.2150	1.4636

#### 6.2.4. Model 2 and Results

In the previous model, the expected value of the distance is used as the GAN's additional loss function, but in the Model 2, the farthest distance among the groups of the generated data belonging to each centroid at the closest is treated as the additional loss function value.

## Results

Table 8: Results of Model 2 with the mixed Gaussian 8 distributions dataset

Values of $k$ (and $K$ )	Distributions of original data and centroids	KDE results
$k = 2$ ( $K = 500$ )		
$k = 50$ ( $K = 20$ )		
$k = 125$ ( $K = 8$ )		
$k = 250$ ( $K = 4$ )		

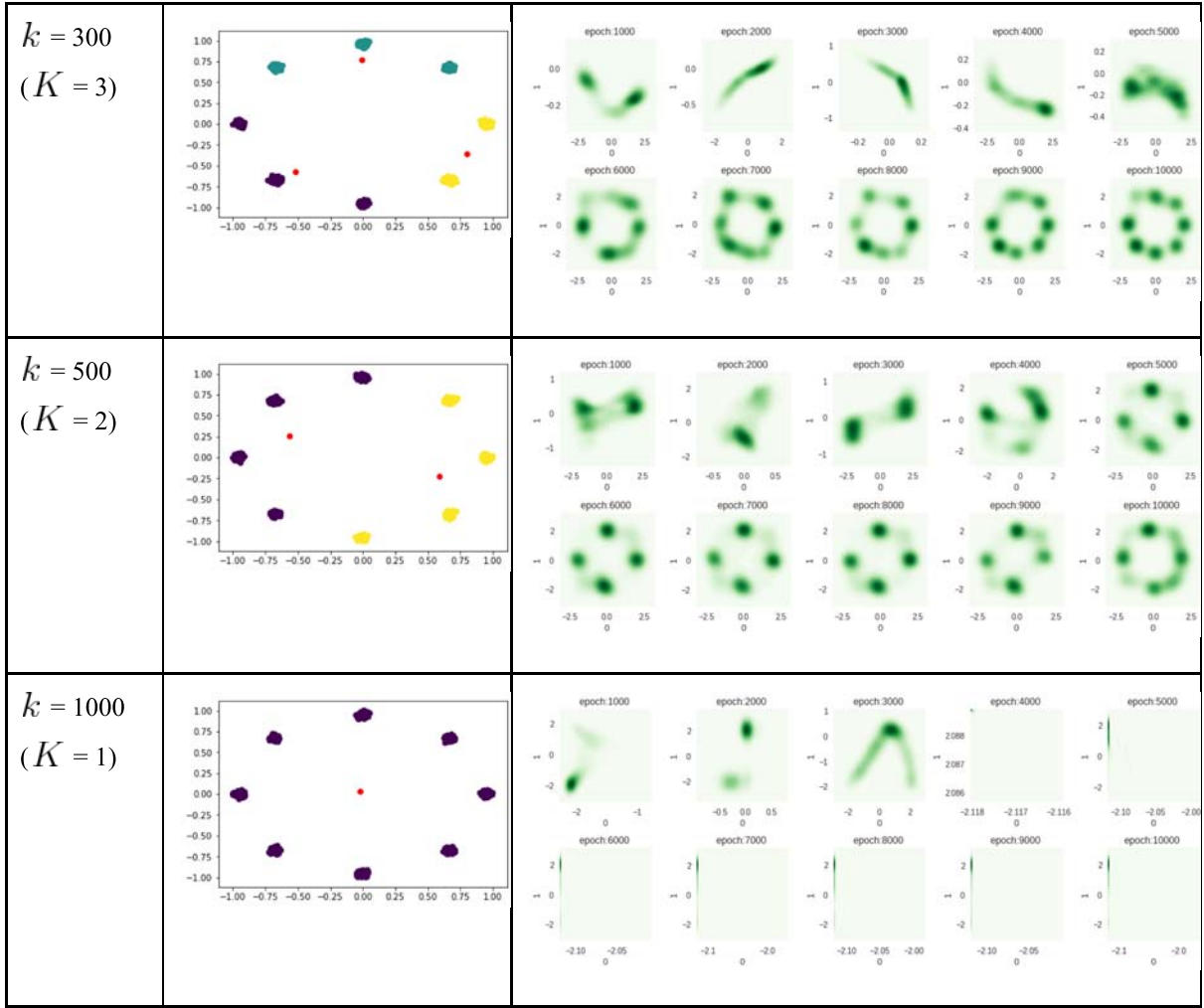


Table 9: Results of Model 2 with the dataset for Bayesian Linear Regression

	Truth	sample	$k = 0$	$k = 2$	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
alpha	1	0.906	1.054	0.792	0.949	1.1	1.166	0.689	0.28
beta1	1	0.948	0.853	0.858	1.191	1.57	1.073	0.909	1.588
beta2	2.5	2.607	2.854	3.046	1.862	2.658	0.884	4.361	3.445
sigma	1	0.962	0.878	1.107	0.775	0.672	1.169	0.738	1.035
score	0	0.291	0.677	<b>1.003</b>	<b>1.105</b>	<b>1.156</b>	<b>2.024</b>	<b>2.525</b>	<b>2.288</b>

Although some results are not as planned (e.g., KDE result where  $k = 250$  and regression result where  $k = 1000$ ), both results seem as expected in general and accord to our proposed method concept.



Table 10: Results with the microaggregated dataset using K-means clustering (the comparison target)

	Truth	sample	$k = 0$	$k = 2$	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
alpha	1	0.906	1.054	0.993	0.948	1.003	1.04	0.675	1.043
beta1	1	0.948	0.853	0.994	1.11	1.05	1.034	1.318	-0.509
beta2	2.5	2.607	2.854	2.696	5.666	11.25	11.927	0.813	-0.728
sigma	1	0.962	0.878	1.108	1.051	0.471	0.335	5.143	6.939
score	0	0.291	0.677	<b>0.317</b>	<b>3.379</b>	<b>9.332</b>	<b>10.166</b>	<b>6.473</b>	<b>10.719</b>

Table 10 is the result of the comparative experiment. It is demonstrated that standard implementations of the k-anonymity including microaggregation significantly reduce the data utility as [46] indicates. These implementations cannot handle large values of  $k$ , and in [19] the value of  $k$  is at most 10; however, in reality, where  $k = 10$ , the probability of being an data subject is not so low as 10%, and the remaining nine people sustain highly likely to be suspected of getting in a false light to the public<sup>3</sup>. On the other hand, it is noted that our proposed method can cope with much larger  $k$  maintain much higher data utility. As a result, the augmented k-anonymity based on GAN enables to significantly reduce data loss while achieving larger k-anonymity than the conventional method.

---

<sup>3</sup> A type of defamation or a tort concerning privacy. It might be considered for example that a person may get rejected to obtain insurance because of belonging to the same k-anonymized group as a patient who has a serious illness.

## 7. Discussion

### 7.1. Discussion

We start this research with the main question: “How data providers can share their privacy-concerned data in a secure way” for close or closer collaboration with a third party, and, to answer to this, we set two concrete sub-questions;

RQ1: How to evaluate the “secure” level?

RQ2: How much statistically meaningful data can be generated in such a secure way?

k-anonymity is the widely used concept of privacy protection. It is an index that explains how secure the k-anonymised dataset is. Though this concept is suitable to apply for RQ1, [46] reports that standard implementations of k-anonymity such as microaggregation substantially reduce utility of original data. On the other hand, the research on synthesizing data using GAN [28] merely aims for improving the quality of generated data. Hence, we try to integrate these two. However, k-anonymity can not be directly applied to GAN’s learning, we propose a concept of the augmented k-anonymity and several implementation models.

The results of Model 1 and 2 with the mixed Gaussian 8 distributions express the proposed concept well, that is, the larger the parameter  $k$  becomes, the more privacy protection will be applied. As a result, we can control the balance of data utility and privacy protection level of generated data. Still, we cannot be sure that how statistically meaningful to share such generated data.

Therefore, assuming Bayesian Linear Regression, which is a slightly advanced analysis method, we consider a true model, take sample data, train GAN, estimate true parameters based on generate data, and compare at multiple different  $k$ . The results of Model 2 proves that our proposed method produces higher quality data at the same privacy protection level than the other method and can handle much larger  $k$ .

Sequentially, we can answer the main question with the augmented k-anonymity in the realm of generative adversarial networks. This allows data providers to control the quality of the generated data with one parameter  $k$ , and to explain the privacy protection level in the existing framework. Unlike sharing only calculation results as found in secure computing, sharing of data similar to original ones provides analysts freedom.

However, it should be noted that, because the augmented k-anonymity extracts particular features of the original data, depending on the distribution of the original data, the quality of generated data cannot be determined by merely manipulating  $k$ . In practice, the analysis of target data will be required in advance. For example, like mixed Gaussian 8 distributions dataset, when the target data is clearly partitioned, additional loss value to GAN may not influence on generated data.

Furthermore, this method relies on the fact that the actual computational part of deep learning is currently a black box. As long as it is a black box, it is impossible to estimate the original data from generated data having no one-to-one correspondence, but it may also be possible when this black box is solved. However, it will still require a great deal of computing power.

From this, the proposed method can also be regarded as one type of encryption such as the hash function since only the encryption key, and no decryption key exists. It's like a one-way function from original data to generated data. Since the generation data is affected by the initialization conditions at the time of generation, it is considerably rare to obtain the same output for the same data. This property indicates that the method is resistant to attack.

There are two things to keep in mind when introducing in practice. In this method (depending on the data size), it takes a certain level of computational power and time when learning. The requirement of computational power needs to consult with the benefits and business impact of sharing data. A relatively quick solution for long-time consuming is to leverage the fixed pre-learned parameters, generate random numbers for each query, pass them to the proposed method, and generate data. This solution enables to synthesize data in a few seconds, but the risk of differential cryptanalysis [4] cannot be removed. To reduce this risk, one possible method could be receiving a query and then starting learning from scratch one by one. This takes large amount of time, but safety level will increase. In any case, when disclosing data, it is better to limit the distribution to those who have undergone certain procedures.

## 7.2. Possible Application

Figure 11 is about the architecture of X-Road in Estonia [20]. X-Road is a data exchange layer. This unifies stored data type and data access interfaces, and that provides an additional security server system to each database which organizations or companies want to connect to the X-Road. To access X-Road, users must be authenticated through e-ID (national ID card) and every data in X-Road is encrypted. X-Road is one of the most crucial factors of e-Estonia, however, we would like to propose a further development idea for boosting the economy.

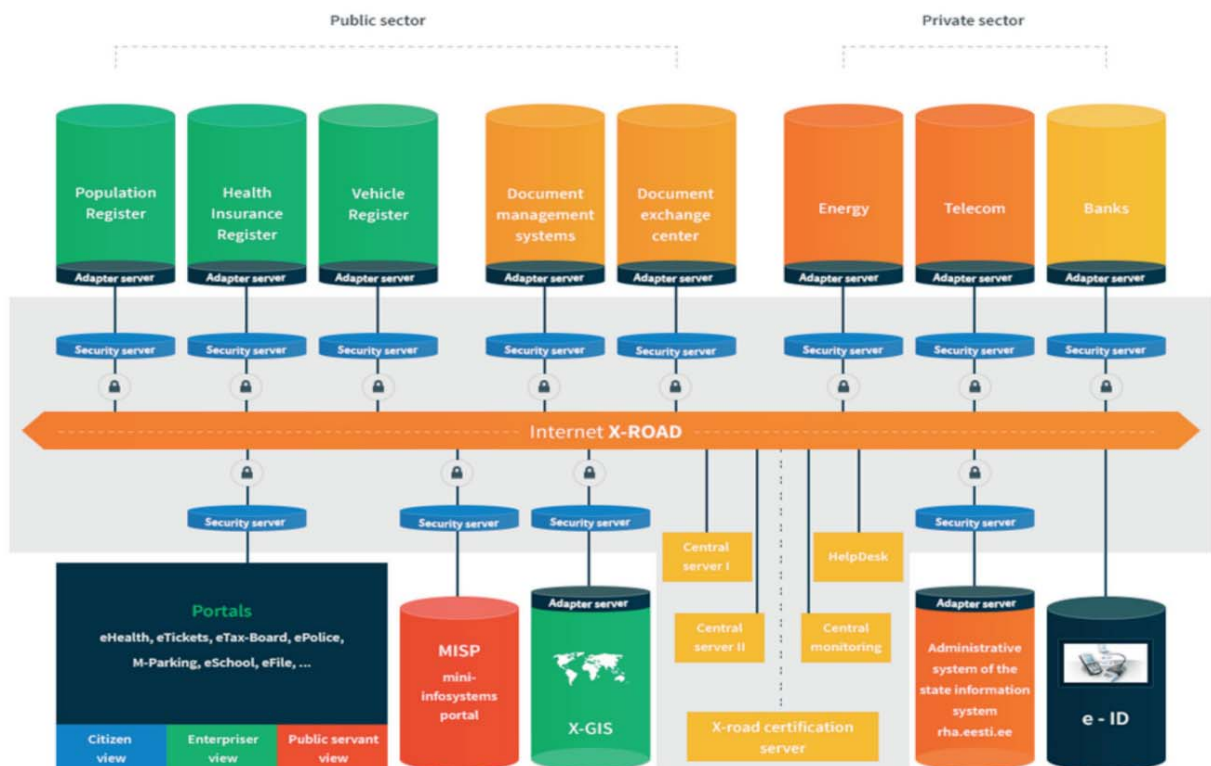


Figure 11: Architecture of Estonian X-Road. Adapted from ‘X-ROAD FACTSHEET’, by Republic of Estonia Information System Authority, 2014. <https://www.ria.ee/sites/default/files/content-editors/publikatsioonid/x-road-factsheet-2014.pdf>

Figure 12 represents a possible application of the augmented k-anonymity based on GAN. The following is the protocol according to the numbers in the figure.

1. A user sends a request for authentication to log in to the X-Road
2. The X-Road validate the user
3. The user chooses an available dataset for synthesizing
4. The database sends the dataset to the computational server (e.g., GPU) and this server produces generated data
5. The computational server returns the generated data based on user-chosen original data to the user

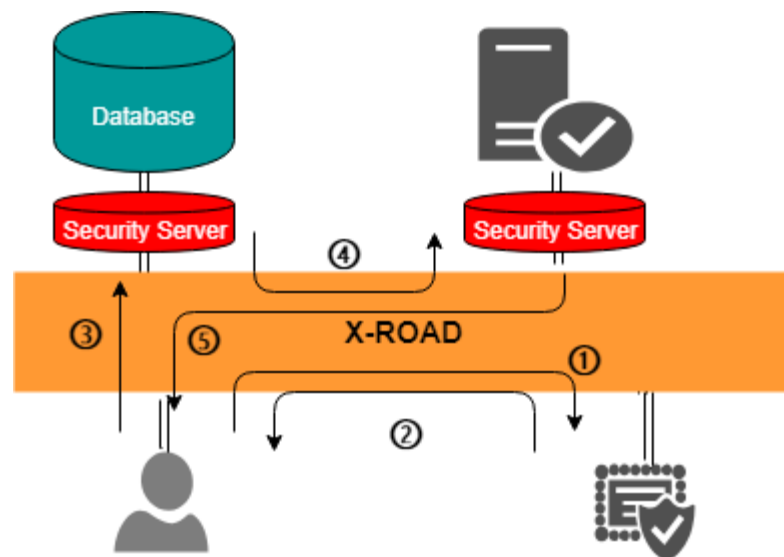


Figure 12: Possible Application of the augmented k-anonymity

That additional computational server is connected to the X-Road and locates behind the security server. Because every datum in X-Road is encrypted and generated data subjected to privacy protection processing, there would be no risk of original data leakage. Moreover, since we can collect user's information and their queries, and we can inquire of such users their purpose and analysis methods they will apply, we can recognize who is interested in what and what for. This information is considerably important in the big data era and would contribute to decision making in the government and designing public services.

## 8. Conclusion

Our proposed method; augmented k-anonymity in the realm of generative adversarial networks, enables data providers to synthesize data satisfying k-anonymity with remaining high data utility. By using method, they can share their privacy-concerned data with a third party safely and establish close collaborations.

# Bibliography

1. Agrawal, R., & Srikant, R. (2000, May). Privacy-preserving data mining. In *ACM Sigmod Record* (Vol. 29, No. 2, pp. 439-450). ACM.
2. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875.
3. Barratt, S., & Sharma, R. (2018). A note on the inception score. *arXiv preprint arXiv:1801.01973*.
4. Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1), 3-72.
5. Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.
6. Che, T., Li, Y., Jacob, A. P., Bengio, Y., & Li, W. (2016). Mode regularized generative adversarial networks. arXiv preprint arXiv:1612.02136.
7. Diffie, W., & Hellman, M. E. (1979). Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3), 397-427.
8. Domingo-Ferrer, J., & Torra, V. (2005). Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2), 195-212.
9. ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4), 469-472.
10. Feldman, P. (1987, October). A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*(pp. 427-438). IEEE
11. Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In *Stoc* (Vol. 9, No. 2009, pp. 169-178).
12. Gentry, C., & Boneh, D. (2009). A fully homomorphic encryption scheme (Vol. 20, No. 09). Stanford: Stanford University.
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems* (pp. 5767-5777).
15. Hamerly, G., & Drake, J. (2015). Accelerating Lloyd's algorithm for k-means clustering. In *Partitional clustering algorithms* (pp. 41-78). Springer, Cham.
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems* (pp. 6626-6637).
17. Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1), 1593-1623.
18. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
19. Kabir, Md Enamul and Mahmood, Abdur Naser and Mustafa, Abdul K. (2012) *K-means clustering microaggregation for statistical disclosure control*. In: 2012 International Conference on Advances in Computing (ICADC 2012), 4-6 July 2012, Bangalore.
20. Kalja, A. (2002). The X-road project. A Project to Modernize Estonia's National Databases. *Baltic IT&T review*, 24, 47-48.
21. Kruschke, J. K., Aguinis, H., & Joo, H. (2012). The time has come: Bayesian methods for data analysis in the organizational sciences. *Organizational Research Methods*, 15(4), 722-752.
22. Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79-86.
23. Lloyd, S. (1982). Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2), 129-137.
24. Lu, Wen-jie, Shohei Kawasaki, and Jun Sakuma. "Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data." (2017).
25. Metz, L., Poole, B., Pfau, D., & Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163.
26. Nagarajan, V., & Kolter, J. Z. (2017). Gradient descent GAN optimization is locally stable. In *Advances in Neural Information Processing Systems* (pp. 5585-5595).



27. Nowozin, S., Cseke, B., & Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems* (pp. 271-279).
28. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., & Kim, Y. (2018). Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10), 1071-1083.
29. Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3), 1065-1076.
30. Patil, A., Huard, D., & Fonnesbeck, C. J. (2010). PyMC: Bayesian stochastic modelling in Python. *Journal of statistical software*, 35(4), 1.
31. Ritchie, F., & Elliott, M. (2015). Principles-versus rules-based output statistical disclosure control in remote access environments.
32. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120-126.
33. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234-2242).
34. Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11), 612-613.
35. Simon, H. A. (1996). *The sciences of the artificial*. MIT press.
36. Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., & Sutton, C. (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems* (pp. 3308-3318).
37. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
38. Sweeney, L. (2000). Uniqueness of simple demographics in the US population. LIDAP-WP4, 2000.
39. Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 557-570.
40. Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1), 75-105.

41. Wang, X., & Gupta, A. (2016, October). Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision* (pp. 318-335). Springer, Cham.
42. Willemsen, J. (2011, August). Pseudonymization service for X-road eGovernment data exchange layer. In *International Conference on Electronic Government and the Information Systems Perspective* (pp. 135-145). Springer, Berlin, Heidelberg.
43. Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.
44. Yao, A. C. C. (1982, November). Protocols for secure computations. In *FOCS* (Vol. 82, pp. 160-164).
45. Young, J., Graham, P., & Penny, R. (2009). Using Bayesian networks to create synthetic data. *Journal of Official Statistics*, 25(4), 549.
46. Zhang, Z., Sun, Y., Xie, X., & Pan, H. (2015, August). An efficient method on trajectory privacy preservation. In *International Conference on Big Data Computing and Communications* (pp. 231-240). Springer, Cham.
47. 佐久間淳 (Jun Sakuma). (2016). データ解析におけるプライバシー保護(Privacy Preservation in Data Analytics). 講談社 (Kōdansya).

# Appendix

The followings are failed models. Those end up with the same results with the mixed Gaussian 8 distributions: no mode collapse for each different  $k$  at the end (at 10,000 epoch).

## Failed Model 1 (FM1)

FM1 is inspired by the definition of convergence. Where there exists  $k$  or more than  $k$  points of generated data in a circle of radius  $r$  from each original data, the definition of augmented  $k$ -anonymity will be satisfied. We adopt K-means clustering as a way of determining  $r$ . That is, half of the distance between the two farthest points belonging to each cluster divided by K-means clustering is taken as a radius  $r$ . If the data size is  $N$ , then  $K = N // k$ , where  $K$  and  $k$  are natural numbers,  $//$  represents integer division. The distance is L2 norm, Euclidean distance.

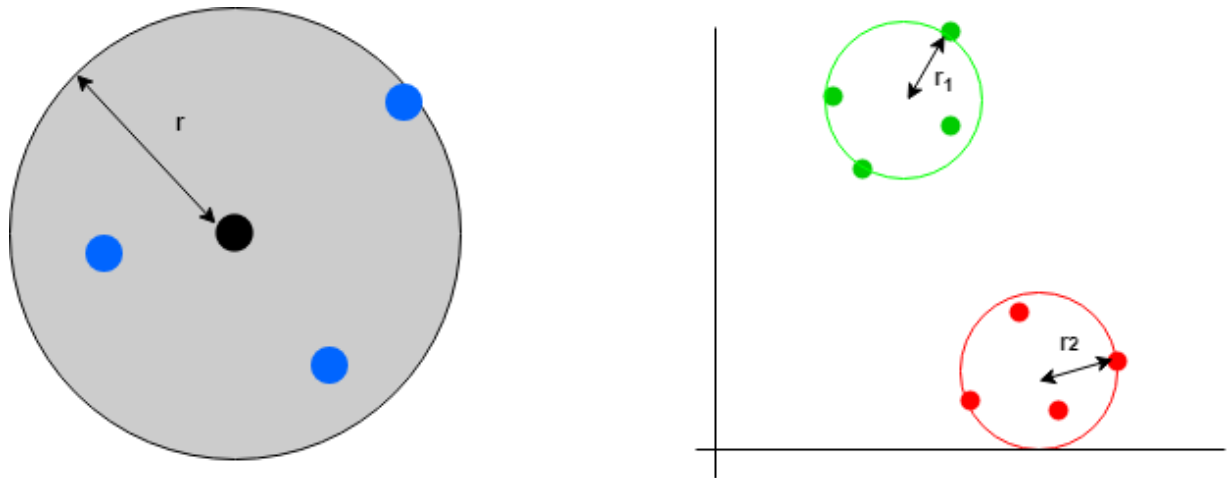


Figure 13: Definition of FM1 (left) and the way of determining  $r$  (right)

## Possible Defect

The model is intuitively easy to understand, but it can be seen that GAN does not train well partly because  $k$  is a discrete value. Also, although K-means clustering is used in determining  $r$ , it is hard to directly explain the relationship between  $r$  and augmented  $k$ -anonymity determined in this model.

## Failed Model 2 (FM2)

The basic framework is the same as that of FM1, but the objective of training in FM2 is closing the mean of distances between the  $k$ -th neighborhood and original datum to 0, rather than the number of discrete values.

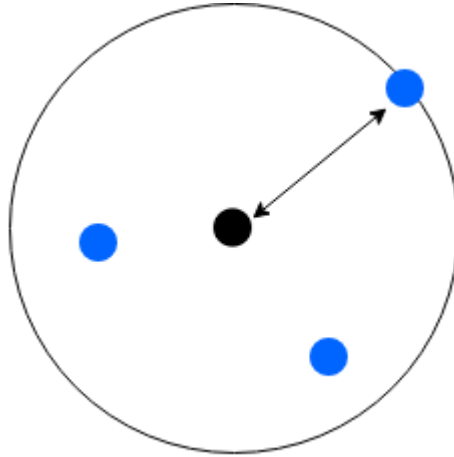


Figure 14: Example of FM2 where  $k = 3$

### Possible Defect

There are cases where the  $k$ -th neighborhood point may become a  $k$ -order or less neighborhood point for other original data.

## Failed Model 3 (FM3)

Quitting thinking the equidistance from each point of the original data, FM3 newly calculates the distance of each original data from each generated data, rearranges them in ascending order, and GAN train the standard deviation of such  $k$  distances to convergent to 0. If this standard deviation becomes 0, then  $d_1 = \dots = d_k$  is achieved and the definition of the augmented  $k$ -anonymity will be satisfied.



Figure 15 (reuse of Figure 5): Example of FM3 where  $k = 3$

### Possible Defect

The concept of the augmented  $k$ -anonymity is “the larger  $k$  becomes, the more blur the generated data will be” so that as the  $k$  becomes larger, the corresponding loss values for Generator’s loss should increase as well. However, because of the law of large number, as the  $k$  rises, the associated standard deviation approaches a constant value. This is against the concept. Therefore, it is necessary to handle features that increase with  $k$  instead of following the law of large numbers.

### Failed Model 4 (FM4)

This model provides a closer match to the original GAN structure. After the centroids are determined by K-means clustering, the sample is newly produced for batch size centering around the centroids by mixed of Gaussian, and GAN trains to make the difference between the entropy of these data and the generated data zero. In the original GAN, Kullback-Leibler divergence was used to calculate this entropy, but the more general; Jensen-Shannon divergence is used.

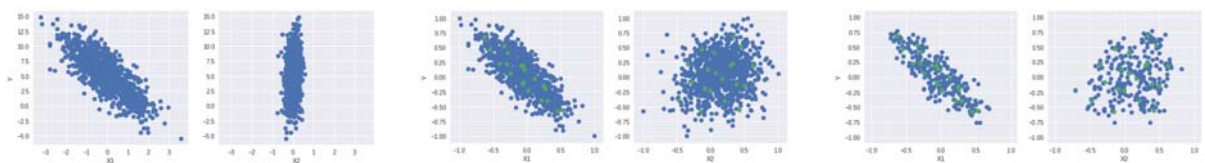


Figure 16: Explanation of model 6, where  $k = 50$ . The original data (left), 20 centroids in green color (middle), and resample by batch size (250) around the centroids by mixed of Gaussian (right). The data in the middle and right Figure 16 is normalized.

**Possible Defect:** unknown.

### **Note**

The codes are written in python and tensorflow; a machine learning library. The Model 1 and 2 that succeeded in the experiments are using the “py\_func” function to calculate the distance between the centroid and the generated data, as numpy array instead of tensor, converting only the loss values back into tensor. When the same calculation is performed with only tensor, the generated data converges to a certain point. It is maybe because backpropagation converges when only using tensor. Note that backpropagation is not applied to numpy array in the py\_func function.

When using WGAN (unrolled WGAN), we could no more control the balance of data utility and privacy protection level with parameter  $k$ . This maybe because unrolled WGAN too strong and ignore the additional loss function which backpropagation cannot be applied.