A Kernel Density Estimation Method to Generate Synthetic Shifted Datasets in Privacy-Preserving Task

Muhammad Syafiq Mohd Pozi*, and Mohd. Hasbullah Omar
School of Computing, Universiti Utara Malaysia, 06010, Sintok, Kedah, Malaysia
{syafiq.pozi, mhomar}@uum.edu.my

Abstract

In order to perform comprehensive analytic task, it requires the availability of any particular complete dataset in the first place. However, due to privacy concern, the specific demand on sharing full dataset to third parties is hardly to be fulfilled. New methods using systematically synthetic data generation in order to preserve the data privacy have recently been explored and identified as a suitable approach to address the privacy concern. Throughout this work, a privacy-preserving probability based synthetic data generation framework for supervised based data analytic is proposed. Using a generative model that captures and represents the probability density function of dataset features, a new privacy-preserving synthetic dataset is synthesized, such that, the new dataset is statistically different from the original dataset. Then, we simulate a supervised learning task using two different machine learning classifiers, as a method to compare the utility of original and the new privacy-preserving synthesized dataset. From the experimental results, we found that the proposed synthetic generation model can produces a new privacy-preserving synthesized dataset, that has similar data utility as to the original dataset.

Keywords: Privacy Preservation, Dataset Shift, Data Anonymization, Differential Privacy

1 Introduction

Nowadays, we are living in a data-driven society [17]. Regardless of our current geographical position, we either generate, or consume data at any point of time, whether we realize it or not. Some of these data are tend to be public, but some are strictly confidential and private. For example, public data could be a post in social network platforms, but the source of the post, such as IP address of the poster is considered as private.

With the public availability of sophisticated data science tools [5, 11], various analytic tasks can be performed to infer how the data were generated in the first place. Using statistical and machine learning analysis, one can approximately estimate the probability distribution function of particular data [32, 23]. It is then possible to deduce how the data was generated for a given particular context, e.g. medical [35], utility [28] and finance [14]. Data in such domains however usually contain personal information which can describe the data producer or any person related to the data. Hence, it is compulsory to ensure that analytics is not performed on raw data as this is likely to lead to the invasion of the data owners' privacy.

Privacy-preserving process needs to be performed on the raw data to ensure that the privacy of any entity related to the data is protected. Methods for doing this can be categorized into two types: *data anonymization* and *differential privacy*. Data anonymization is used to anonymize the data by increasing the data generalization property, often measured through k-anonymity [33]. Other methods such as

Journal of Internet Services and Information Security (JISIS), volume: 10, number: 4 (November 2020), pp. 70-89
DOI: *10.22667/JISIS.2020.11.30.070*

*Corresponding author: School of Computing, Universiti Utara Malaysia, 06010, Sintok, Kedah, Malaysia Tel: +60-(0)4-928-5217, Web: `https://tinyurl.com/y4hjf5bh`

data perturbation [34], data randomization [30] and data masking [31] have also been used for privacy-preservation task.

The second privacy-preserving method is differential privacy [37, 6, 16]. Query answering algorithms that satisfy differential privacy must produce noisy query answers such that the distribution of query answers changes very little with addition, deletion, or modification of any tuple. The formal definitions of differential privacy are as follows:

Definition 1.1. (Unbounded Differential Privacy [6]). A randomized algorithm A satisfies unbounded ε-differential privacy if the relation $P(A(D_1) \in S) \leq e^{\varepsilon} P(A(D_2) \in S)$ exists for any set S and any pairs of databases D_1, D_2, where D_1 can be obtained from D_2 by either adding or removing one tuple.

Definition 1.2. (Bounded Differential Privacy [7]). A randomized algorithm A satisfies unbounded ε-differential privacy if the relation $P(A(D_1) \in S) \leq e^{\varepsilon} P(A(D_2) \in S)$ for any set S and any pairs of databases D_1, D_2, where D_1 can be obtained from D_2 by changing the value of exactly one tuple.

Under bounded differential privacy, both datasets D_1 and D_2 have fixed size n, while unbounded differential privacy has no such restriction. The privacy guarantees of differential privacy are as follows:

(1) No assumption is made as for how the data or query results are generated.

(2) An individual's information is protected even if an attacker knows about all other individuals in the data.

(3) The query results are robust to arbitrary background knowledge.

However, these differential privacy guarantees are loose interpretations of formal definition provided by the differential privacy [16]. All these assumptions can be easily broken. The first claim can be easily broken if there are some statistical properties or a particular mathematical function that links individual data together. For example, the relation between three data fields, such as x_1, x_2, x_3 could be of a form of $x_1 \times x_2 = x_3$ (i.e., x_3 might be representing the value of an area for a given region).

The second claim can be easily broken when an adversary can determine the missing link through evidence of participation. For instance, there is no chance of preventing an adversary from identifying hidden x_1 in a collection of data. An adversary could easily deduce that $x_1 = \frac{x_3}{x_2}$, hence, recovering x_1.

The third claim that says differential privacy is robust to arbitrary background knowledge has been formalized and studied by [15]. To be precise, the third claim stated that differential privacy is robust when certain subsets of the tuples are known by an adversary. However, multiple queries, consisting of past queries, can be used and combined together in order to identify the exact value of the masked dataset. Such background knowledge includes previously released exact query answers. For example, two different queries that extract x_1 and x_3 can be combined together to retrieve x_2.

Regardless, any kind of privacy-preservation process should not degrade the utilization of data (data utility). Data sanitation such as data generalization based on k-anonymity [33], data perturbation [9], and data randomization [27] might remove some important data from data user, making it hard for extracting useful patterns from that data.

In this paper, we propose to overcome the limitations of differential privacy through shifted synthetic dataset in order to maintain the data utility. In particular, our main contribution can be described as a framework to produce synthetic dataset in which the distribution function of synthetic data is different from the original data distribution function without degrading the data utility metric, when compared to the original data. Given a dataset, we thoroughly discuss how to interfere with the statistical properties of the dataset without degrading the data utility performance. The results produced in this paper are

analyzed based on how much the privacy is preserved and how much the data utility performance has changed. The data utility performance is measured through classification task.

This paper is an extension of [25], to which we add larger experiments (feature correlation analysis and data visualization) due to method enhancement , as well as thorough and wider discussion of the proposed method. In addition to that, we also reformulate the previous model to make it more generic. Unlike in [25], the new model proposed in this paper does not take the class relation of the data, y, as a requirement to be considered during the shifting task. In specific, previous works requires different parameter for each class, while there is no such requirement in this work. Hence, more randomness can be introduced into the newly synthesized data, which will make it hard for anyone with no access to original data to infer the characteristics of original data from the synthesized data.

The outline of this paper is as follows. Section 3 describes how we perform differential privacy through generating shifted synthetic dataset. Section 4, we measure the capability of the proposed generative model in preserving the data utility through classification task on commonly used datasets in machine learning literature [2]. Following, section 4.4 shows the way in which we perform privacy property analysis in the context of this paper. In Section 5, we highlight the limitations of our generative model. Finally, Section 7 concludes this paper.

2 Background of Study

Generally, three entities directly involved with data can be distinguished. Those entities are as follows:

(1) *Data owner*: Data owner defines, generates and provides information about the legit owner of data assets and the acquisition, use and distribution policy implemented by the data owner.

(2) *Data keeper*: Data keeper stores and preserves the data. Data keeper also acts as the intermediary entity that connects data owner and data user.

(3) *Data user*: Data user uses the data for processing and especially for analytic tasks, acquitted from data keeper.

The process of data sharing is usually in one direction, from data owner to data keeper and finally to data user. Figure 1 illustrates this process based on a scenario of an Internet user, Internet service provider (ISP) and some government.

In Figure 1, Internet user is the data owner, Internet service provider (ISP) is the data keeper, and government is the data user. In ideal scenarios, data keeper is obliged to protect the privacy of data owner. However, in reality, this is not really the case. Governments for example, requires data keepers, such as ISP, to share or handover their customers usage data for profiling purposes [22, 3].

Therefore, the privacy of the data owner should be protected and preserved. In this paper, for the privacy of the data is to be truly preserved, 4 main privacy properties need to be properly adhered to, which are *Anonymity*, *Unlinkability*, *Pseudonymity*, *Deniability* [24, 4, 13]. These properties are described such as follows:

(1) *Anonymity* ensures that a user may use a resource or service without disclosing the user's identity.

(2) *Unlinkability* ensures that a user may make multiple uses of resources or services without others being able to link these uses together.

(3) *Pseudonymity* ensures that a user may use a resource or service without disclosing his or her identity, but can still be accountable for that use.

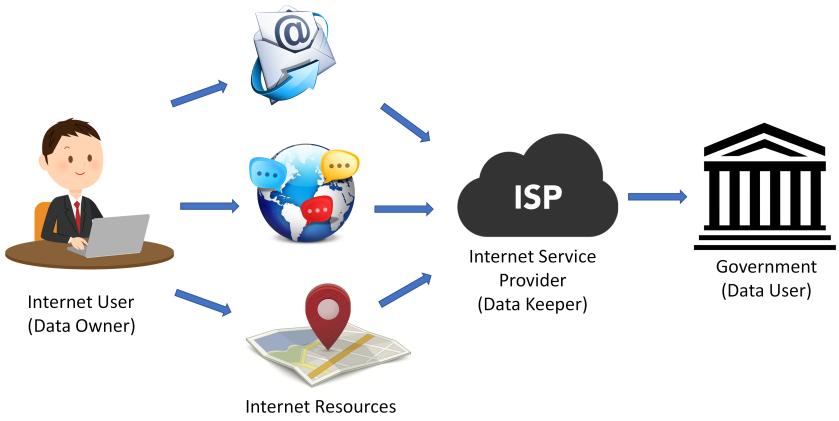

Figure 1: A simple scenario of how data flow from Internet users to government through Internet service provider.

(4) *Deniability* offers a mechanism to facilitate users in refuting and contradicting their involvement in using some particular resources.

Based on Figure 1, let us consider a scenario where a hypothetical government tries to identify a whistleblower that actively helped in leaking government secrets by publishing them online. When that particular user used several Internet resources, he or she must ensure to remain unknown to the Internet community for fulfilling the *anonymity* property. Every Internet resource that have been used (visited websites, Twitter post, etc.) by that anonymous user must not be linked to a single entity, hence, achieving the *unlinkability* property. Meanwhile, for example for accounting purposes, the billing for using the resources must be associated with that anonymous user. Therefore, there must be a mechanism to associate all these used resources to a single but anonymous entity, in order to achieve *pseudonymity* property. Finally, this anonymous user must be able to deny his or her involvement when using those resources in order to achieve *deniability* property.

3 Generative Model

Here, the generative model for synthesizing new privacy preserving synthetic dataset is described accordingly. The synthesizer is based on a probabilistic model that approximate the probability density function of original dataset features. That is, the synthesizer must me modelled from the original dataset in order to synthesize new privacy-preserving synthetic dataset.

Dataset X is defined such that $X = \{x_1^d, x_2^d, \ldots, x_n^d\}$. Dataset X consist of n records with d random variables associated with the records' features. The proposed generative model will map X into a new privacy-preserving dataset X' with respect to X. The new X' will be the one to be utilized by the data user.

The proposed generative model must however adhere to certain assumptions, such as follows:

Assumption 1. *As emphasized in Figure 1, data keeper acts as an intermediary between data owner and data user. We trust that data keeper must not disclose or reveal real data of data owner to anyone without data owner's consent.*

Assumption 2. *We also trust that data keeper, at any means, must not disclose or reveal the generative model parameters to other users.*

Assumption 1 and Assumption 2 are complementing each other. If any of those assumptions are broken, then the entire generative model scheme is void and could not be implemented properly.

3.1 Estimating Dependent Variables Through Feature Selection

The feature selection algorithm is used to select highly predictive features that have low correlation to each other. In particular we use Correlation-based Feature Subset Selection for Machine Learning (CFS) [12].

Initially, let us define a dataset X consisting of a set of d dependent features $m \in \mathbf{M}$, the following two assumptions are made when determining independent and dependent features:

Assumption 3. *Dependent features: For $m \in \mathbf{M}$, we assume that each feature $m \in \mathbf{M}$ is dependent on all features in $\mathbf{M}$, if the CFS algorithm did not select these features as the most relevant feature for a given class variable.*

Assumption 4. *Independent features: Each feature in the selected features is independent to each other, such that $m \notin \mathbf{M}$, if the CFS algorithm selected these features as the most relevant feature for a given class variable.*

The reason for performing feature selection on the dataset is to allow more variance to be introduced into each random variable, independent of each other, without affecting data utility. As a result, not only the new synthetic dataset differs in absolute values, but the correlation between each feature on the new synthetic dataset will also differ compared to the original dataset. After feature selection process, we perform distribution estimation on each independent variable and each set of dependent variables.

3.2 Distribution Estimation

It is pretty common for any particular dataset X has clear relation with target variables Y, usually in the form of a column matrix, $\{y_1, y_2, \ldots, y_n\}$. Each value in Y corresponds to features in X, in respected order. The joint distribution between $\mathbf{x}$ and y, where $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ for a particular feature d, is defined such as follows:

$$P(y, \mathbf{x}) = P(y|\mathbf{x})P(\mathbf{x}) \tag{1}$$

The goal is to estimate $P(\mathbf{x})$, then shift it into $P_{new}(\mathbf{x})$, before generating new synthetic dataset from $P_{new}(\mathbf{x})$, regardless of Y. The generator must be modeled based on the following data properties:

(1) **Dependent variables**: Some $x \in \mathbf{x}$ are highly correlated to each other. If there are at least two x variables that are correlated to each other, we can express this relation as a joint probability distribution between two random variables, such as:

$$f_{X_1,X_2}(x_1, x_2) = f_{X_2|X_1}(x_2 \mid x_1) f_{X_1}(x_1) \tag{2}$$
$$= f_{X_1|X_2}(x_1 \mid x_2) f_{X_2}(x_2) \tag{3}$$

where $f_{X_2|X_1}(x_2|x_1)$ and $f_{X_1|X_2}(x_1|x_2)$ are the conditional distributions of X_2 given $X_1 = x_1$ and of X_1 given $X_2 = x_2$ respectively, and $f_{X_1}(x_1)$ and $f_{X_2}(x_2)$ are the marginal distributions for X_1 and X_2 respectively.

(2) **Independent variables**: There is also a possibility of some random variables that do not depend on each other. Given two random variables, each variable is independent of each other if it can be expressed as follows:

$$f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1) \cdot f_{X_2}(x_2) \tag{4}$$

Here f_{X_1,X_2} is the joint probability distribution function between the dataset features. This means that obtaining some specific information about the value of one or more random variables leads to the conditional distribution of any other variable that is similar to its unconditional distribution, such that there is no variable exists to provide any information about any other variable.

In order to satisfy these properties (loosely), we used feature selection to determine the most important features $x \in \mathbf{x}$, and group them together into new $\mathbf{x}'$. We considered $\mathbf{x}'$ as dependent variables. Hence, each $x \notin \mathbf{x}'$ is considered as an independent variable.

Case 1: $m \notin \mathbf{M}$, that is, m is independent of each other. Let $m_1, m_2, \ldots, m_i$ be a set of features with i independent features, in which each m is independent of each other, and the features are drawn from a common distribution characteristics represented by the probability density function f. The kernel density estimate (KDE) model is defined as follows:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big) \tag{5}$$

where

- x is a single data point defined by each m;
- **h** is the bandwidth, also known as smoothing parameter;
- K is the kernel function;

In this paper, we opt for Gaussian kernel, defined as follows:

$$K(x;h) \propto \exp\Big(\frac{x^2}{2h^2}\Big) \tag{6}$$

The approximately deduced distribution function is then shifted to another distribution function:

$$f_{new}(\mathbf{x}) = z\hat{f}_h(\mathbf{x}) \tag{7}$$

where z is a user-defined weight.

Case 2: $m \in \mathbf{M}$, that is, each x is dependent of each other. Let $m_1, m_2, \ldots, m_i$ be a definition of dataset $\mathbf{x} \in \mathbf{X}$ with i dependent features in which each $m \in \mathbf{M}$ is dependent of each other, and are drawn from a common distribution described by the density function f.

In **Case 1**, the inferred distribution is shifted to another distribution as in Equation 7 where z is the user-defined weight. For **Case 2**, estimating more than 2 jointly random variables is memory and time consuming [21]. Hence, based on dataset $\mathbf{X}$, we calculate the correlation matrix for each feature $m \in \mathbf{M}$. Each highly correlated pair is grouped together, such that:

$$G = \{g_1, g_2, g_3, \ldots, g_n\}, \bigcap_{i=1}^{n} g_i = \{\varnothing\} \tag{8}$$

G is a set of group g_i of highly correlated features $\{m_1, m_2, \ldots, m_j\}$, such that $g_i = \{m_1, m_2, \ldots, m_j\}$. Here we only consider pairs that have absolute highest correlation to prevent overlapping set of features.

The population of correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as in Equation 9:

$$\rho_{X,Y} = \mathrm{corr}(X,Y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{9}$$

In this case, X and Y are random variables representing $m_i(X)$ and $m_j(Y)$.

Based on **Case 1** and **Case 2**, our estimation model can be defined as a set of several $f_{new}(\mathbf{x})$, such that:

$$F = \{f_{new_1}(\mathbf{x}), f_{new_2}(\mathbf{x}), f_{new_3}(\mathbf{x}), \ldots, f_{new_m}(\mathbf{x})\} \tag{10}$$

where m is the total number of distinct sets consisting of dependent variables and total number of independent variables.

3.3 Distribution Shifting

We implement z in Equation 7 as a user-defined weight (facilitated by random number generator function) of each sample, embed accordingly to the feature of that sample through addition operation, such that z is added together with each value in vector $\mathbf{x}$ of specified random variable. Hence, prior to distribution estimation process, we would slightly change value for each dependent and independent random variable, to ensure the probability distribution function is derived from different data. Algorithm 1 describes the distribution shifting process, starting from the feature selection.

Algorithm 1 Distribution Shifting

procedure SHIFT($\mathbf{X}$)
Require:
Dataset $\mathbf{X}$ with label $\mathbf{Y}$.
$\mathbf{M}_{id} \leftarrow CFS(\mathbf{X})$
$\mathbf{M}_{nid} \leftarrow \mathbf{M} \setminus \mathbf{M}_{id}$
$z \leftarrow RNG()$ ▷ Random number
for $m \in \mathbf{M}_{id}$ **do**
$F \leftarrow KDE(z(m(\mathbf{X})))$ ▷ Append into set F
end for
end procedure

In Algorithm 1, M_{id} is a set of independent features and M_{nid} is a set of dependent features. Random number z is a weight for feature m. Finally, the probability density function for shifted dataset X with feature m is computed through $KDE(z(m(\mathbf{X})))$.

3.4 Data Synthesization

Once the probability distribution function has been approximately inferred, new data $\mathbf{x}$ will be generated from the set of derived kernel density estimation models in F, such as in Equation 10. As a result, the new synthetic dataset is statistically different compared to the original dataset. Algorithm 2 describes the process of generating the new synthetic dataset.

Algorithm 2 Data Synthetization

procedure GENERATE(x)
Require:
Record $x \in \mathbf{X}$.
$\{x'_1, x'_2, \ldots, x'_n\} \leftarrow generatePoints(f \in \mathbf{F}) \neq x$
$x' \leftarrow pickOne(\{x'_1, x'_2, \ldots, x'_n\})$ ▷ Randomly
$x \leftarrow m(x')$
end procedure

However, Algorithm 2 is only suitable for generating independent variable, where the probability density function is derived from one dimension. Since there are cases where two random variables are involved in deriving probability density function, we want to ensure the generated synthetic data adhere to the multivariate probability distribution function. Hence, the data generation model is transformed into a linear regression modeling task through multivariate probability distribution function, such as defined in Equation 11.

$$\hat{y}(w,x) = w_0 + w_1 x_1 + \ldots + w_p x_p \tag{11}$$

where p is the length of vector x, and w is the coefficient vector of vector x. $\hat{y}$ is the predicted value given w and x.

However, there is one caveat for this linear regression modeling task to work. The conditional variable, $P(y|x)$ must be predefined first. Here, for each highly correlated pair of features, we manually chose the independent and dependent features as shown in Table 3. From this x, we build a linear regression model as in Equation 11, to compute the $\hat{y}$ value, which is the new synthetic value that form new synthetic dataset $\mathbf{X}'$.

Thus, $\mathbf{X}'$ is the new synthetic data. $\mathbf{X}'$ can now be used in variety of data science and analytic applications, such as classification, clustering or regression tasks. However, our work only focus on classification task. The verification of the proposed algorithm is then done through the classification task.

For clarification, we closed this section by visualizing our proposed generative model as a flowchart, illustrated in Figure 2.

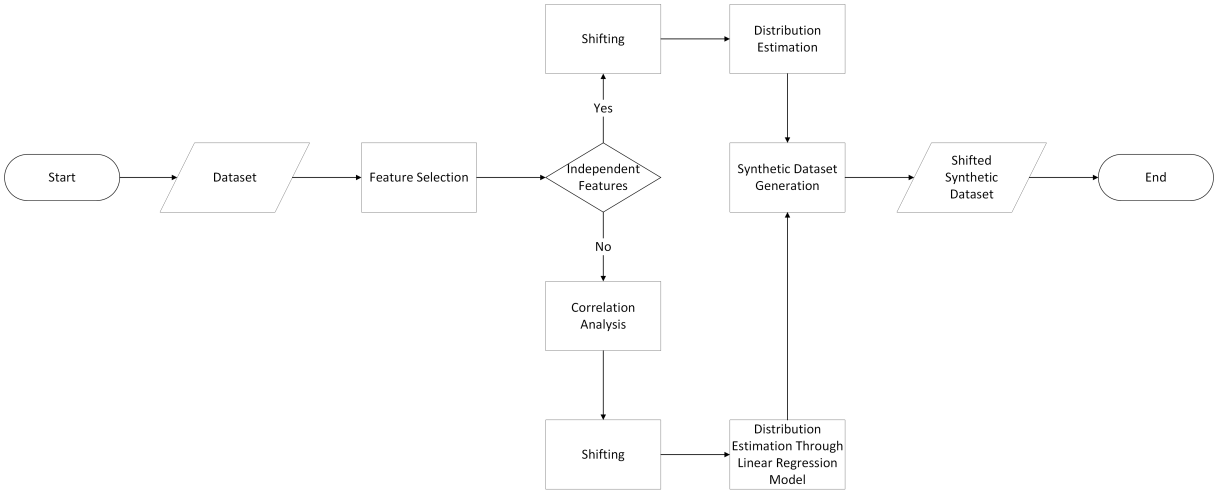

Figure 2: The flowchart for our shifted synthetic dataset generative model.

Figure 2 shows the main processes of our generative model: feature selection for determining independent features, correlation analysis to determine dependent features, dataset shifting where the shifting is performed on each independent and dependent features, distribution estimation to determine the new probability density function of the shifted dataset and finally, the synthetic dataset generation for generating new shifted synthetic dataset.

4 Experimentation and Analysis

The utility of original datasets and the new synthetic datasets are evaluated through classification task. Table 1 outlines each dataset used in the experiment. Based on datasets obtained from UCI machine learning repository [2], the datasets that are used in this experiment are as follows: **Fertility** Diagnosis [10], **Iris**, **Haberman**, **Liver** Disorder, **Breast** Cancer, **Pima** Indians Diabetes, **Thyroid** and **Bank** Marketing [19]. Note that, each dataset was preprocessed such that all non-numeric features transformed into numeric values, through a standardized unique mapping process, starting from 0 to n unique non-numeric values.

Table 1: Datasets used in this experimentation design.

Dataset Name	Total Features	Total Records	Total Classes
Fertility	10	100	2
Iris	4	150	3
Haberman	3	306	2
Liver	7	345	2
Breast	10	699	2
Pima	8	768	2
Thyroid	21	3,772	2
Bank	17	45,211	2

4.1 Experimental Design

The result of classification performance is evaluated using 5-fold cross-validation technique, on two standard machine learning classifiers: Decision Tree [26], and Support Vector Machine [1]. The parameters used in the experiments are as follows:

(1) Decision Tree parameters:

- Confidence factor: 0.25
- Number of folds: 3

(2) Support Vector Machines parameters:

- RBF Kernel = $K(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$
- Kernel parameter, $\sigma = 0.01$
- Regularization parameter $C = 64$

Each classifier is based on WEKA implementation [11]. Table 2 and Table 3 show the experimentation result, based on original dataset and shifted synthetic dataset, across all classifiers, respectively (the tables will be described soon).

4.2 Correlation Analysis

We first measured the correlations between any two features for every original and shifted synthetic dataset. Fig. 3 and Fig. 4 describe the distribution of two selected features in each dataset, allowing to

compare between the original dataset (on the left side of the figure) and the shifted synthetic dataset (on the right side of the figure). Every synthetic dataset has different correlation compared to the original dataset.

Except for the shifted synthetic fertility data, it can be seen that the distribution for shifted synthetic datasets features formed multiple noticeable lines. For shifted synthetic fertility dataset, the data distribution could not be easily fit into simple mathematical function. This is because each value in the feature is randomly selected from precomputed KDE model, defined in Equation 5. For other features in other datasets, this is because, most of those features in the shifted synthetic datasets followed one-dimensional linear regression model, based on $P(y_i|x_i)$, as defined in Equation 11.

Some of the dependent features composition are changed from original dataset to synthetic shifted dataset. This can been seen and compared at *Dependent Features* column for original dataset in Table 2 and at *Dependent Features* column for shifted synthetic dataset in Table 3. Such features are *child-diseases* in **Fertility**, *petal-length* in **Iris**, *drinks* in **Haberman** and *age* in **Pima**. Therefore, we managed to mask the relationship of some of the dependent features. If an adversary finds a relationship between the two features in the shifted synthetic dataset, the relationship is simply untrue, hence preserving the privacy of the data owner.

In addition to that, there are also features that changed from independent features to dependent features and otherwise. From original dataset to shifted synthetic dataset, in terms of independent features to dependent features, such features are *age* and *season* in **Fertility**, *petal-width* in **Iris**, *age-at-operation* in **Haberman**, *pedigree* in **Pima**, *sick*, *TT3* in **Thyroid**, *duration*, *housing*, *poutcome* in **Bank** and most of independent features in **Breast**. In this case, we managed to artificially hide the dependency of some features, and adding artificial relation to some independent features with other features. Having to mask some of the independent features prevent adversary to identify the most relevant features, which could be used to extract some sensitive information from the data. This is because, independent features always bring the most quality information to the dataset.

The correlation value of dependent features also changed significantly from the original datasets to shifted synthetic datasets. Again, the high absolute correlation values result from linear regression model that produces highly correlated data between x and y in $P(y|x)$ relation. As a result, these correlation values can be used to exaggerate the relation of the dependent features, hence masking the real correlation between these dependent features.

4.3 Classification Analysis

The measurement we use is based on accuracy metric defined in Equation 12.

$$accuracy(y,\hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1} 1(\hat{y}_i = y_i) \times 100 \tag{12}$$

In Equation 12, y is the original target value, $\hat{y}$ is the predicted target value and n is the number of samples. We count the number of $\hat{y}_i = y_i$, and divide the obtained count value by the number of samples. The division result is then multiplied by 100 to get the percentage value.

Based on Table 3, it can been seen that the classification accuracy on synthetic data is improved significantly over the original data as shown in Table 2. Here, there is a possibility that the shifting process might introduce some unintentional bias inside the synthetic dataset, allowing some of the classes to be properly discriminated and distinguished.

The composition of the features also helps in improving the classification accuracy. This is especially noticeable on **Breast** dataset, where most of independent features in Table 2 became dependent features in Table 3. In this case, we manage to get 100% classification rate on **Breast** dataset.

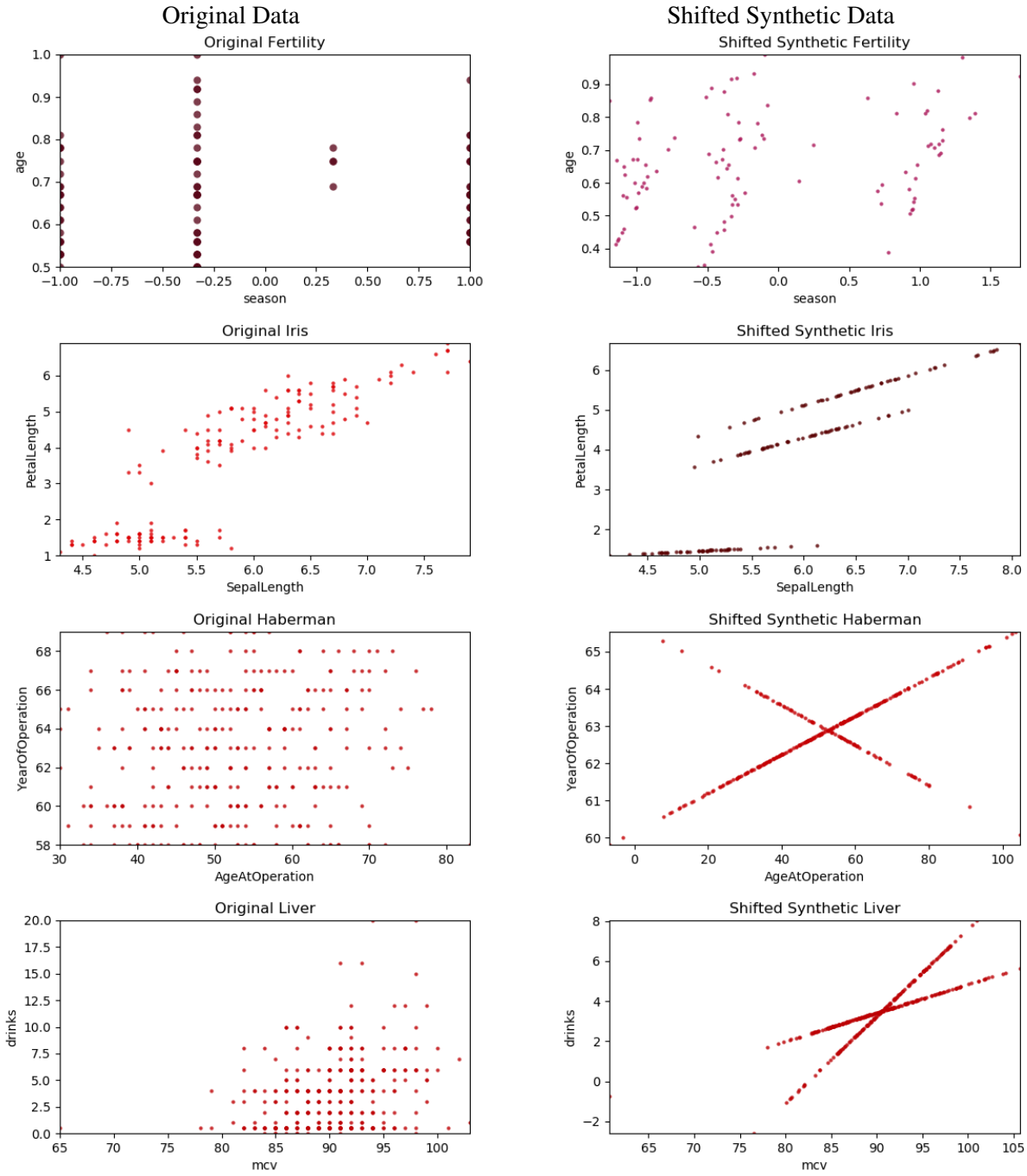

Figure 3: Comparison of shape distribution of two correlated features, between the original **Fertility**, **Iris**, **Haberman** and **Liver** datasets; the and shifted synthetic **Fertility**, **Iris**, **Haberman** and **Liver** datasets, respectively.

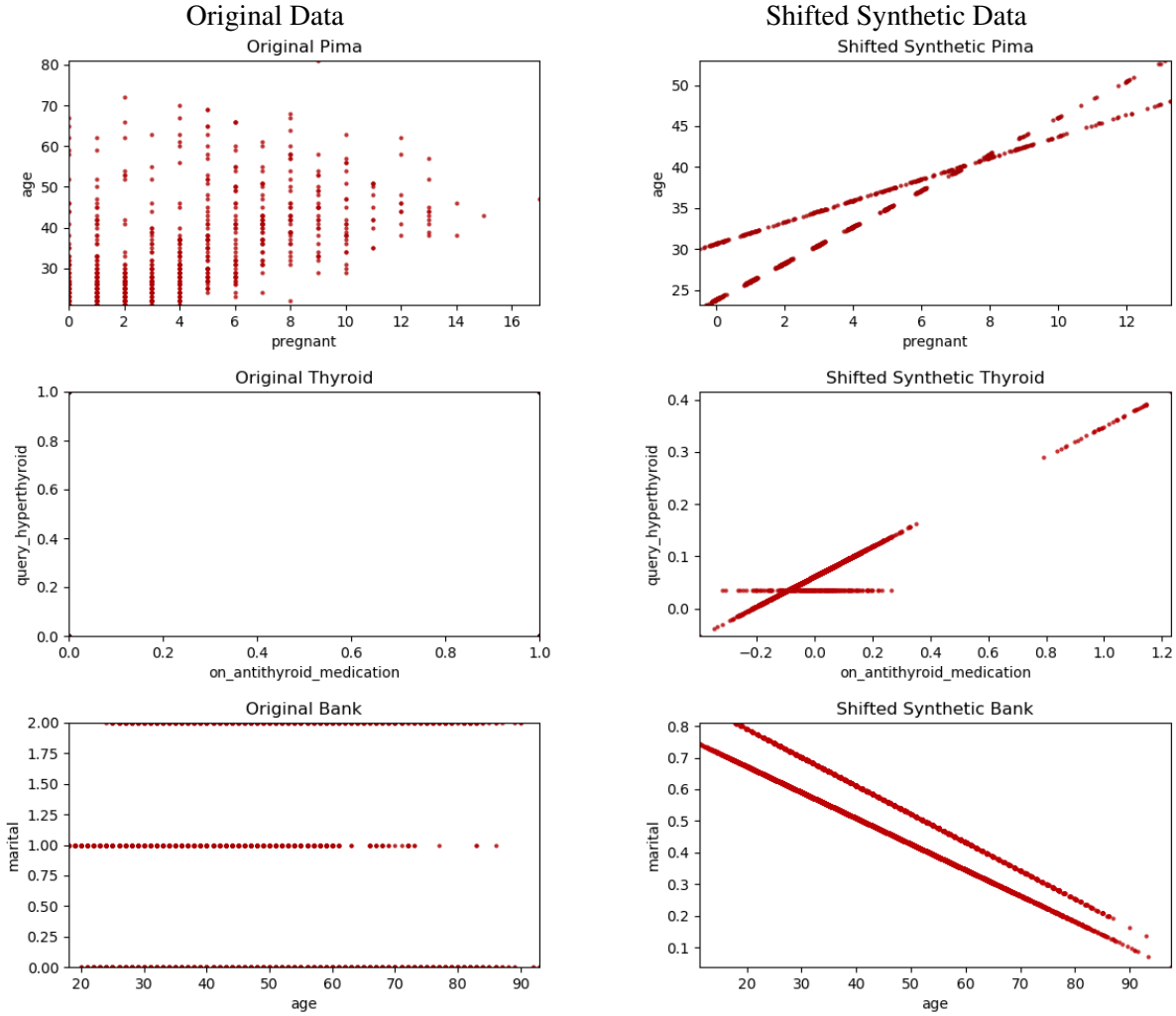

Figure 4: Comparison of shape distribution of two correlated features between the original **Pima**, **Thyroid** and **Bank** datasets; and the shifted synthetic **Pima**, **Thyroid** and **Bank** datasets, respectively.

4.4 Privacy Analysis

In order to analyze the privacy of our generative model, we used Definition 1.2 in the following way:

(1) Since we are sharing the whole dataset, then $P(A(D_1) \in S) = 1$ and D_1 is the whole S.

(2) If D_1 is the original dataset and D_2 is the synthetic dataset, then, it should be $P(A(D_2) \in S) \leq e^{\varepsilon} P(A(D_1) \in S)$. In this case, $e^{\varepsilon} = 1$.

Besides that, we could see that features composition in original dataset, such as in Table 2, are changed in the shifted synthetic dataset, as in Table 3. As a result, the shifted synthetic dataset is largely different compared to the original dataset as well as the statistical composition of independent and dependent features are also significantly changed. For example, if one can infer that there is some correlation between *marital* and *housing* in shifted synthetic **Bank** dataset, hence, it is not entirely true, since in original **Bank** dataset, *housing* is considered as independent variable while *marital* is correlated with *age*. Comparing each dependent variable y in $P(y|x)$, from Table 2 with Table 3, we can see features undergoing change such as *petal-length* in **Iris**, *year-of-operation* in **Haberman**, *drinks* in **Liver**, *age* in

Table 2: Feature Correlation and Classification Results on Original Data Based on Accuracy Percentage (%).

Dataset	Independent Features	Dependent Features $(y\|x)$ y	Dependent Features $(y\|x)$ x	Absolute Correlation Score	Decision Tree	Support Vector Machine
Fertility	age	alcohol-frequency recent-fevers child-diseases num-hours-sitting-pd	trauma surgical-intervention season smoking-habit	0.2427 0.2316 0.1765 0.1061	84.00	88.00
Iris	petal-width	petal-length	sepal-length	0.8718	96.00	96.67
Haberman	positive-nodes	year-of-operation	age-at-operation	0.0895	70.92	70.58
Liver	gammagt	sgot drinks	sgpt mcv	0.7396 0.3126	68.70	64.35
Breast	cell-size bare-nuclei epithelial-cell-size normal-nucleoli bland-chromatin marginal-adhesion clump-thickness mitoses	-	-	-	93.99	94.27
Pima	pedigree	age insulin bmi	pregnant skin-thicknes blood-pressure	0.5443 0.4367 0.2818	71.22	62.23
Thyroid	sick TT3	T4U on-thyroxine hypopituitary query-hyperthyroid TT4 TBG-measured query-hypothyroid psych TSH-measured TT4-measured FTI TBG FTI-measured	age sex query-on-thyroxine antithyroid-medication pregnant thyroid-surgery I131-treatment lithium goitre tumor TSH T3-measured T4U-measured	0.1541 0.0882 0.1405 0.1266 0.1513 - 0.0472 0.0379 0.0630 0.0499 0.1922 - 0.9971	98.36	92.95
Bank	duration housing poutcome	previous month education campaign marital loan	pdays contact job day age balance	0.4548 0.4387 0.2595 0.1625 0.1264 0.0844	89.82	87.95

Pima, and most of the features in **Breast** and **Thyroid**. Especially this is noticeable in **Breast** dataset, where the correlation in shifted synthetic dataset does not exist in the original dataset.

5 Discussion

Regardless of the mathematical derivations for producing the shifted synthetic datasets, our method could simply be considered as a kind of a cryptography task, encryption. In the end, we changed original values to other new values, so that one cannot easily deduce the original value. Still, our method is quite different compared to traditional encryption, as the latter is based on a fixed sequence of steps. For example, a two way encryption functions such as Advanced Encryption Standard (AES) [20] or one way encryption function such as Secure Hash Algorithm (SHA) [8] consist of their own sequences of steps. One can consider our proposed method as producing a new data as a framework based on statistical approach that the entire or some of the processes in the encryption task can be easily manipulated to cater for many use case scenarios.

Another thing is that the proposed method makes it more difficult to decipher the original data when the input data has increased significantly. There are many possible values that can be used to mask a single original value that is a part of large dataset. However, in encryption, as the data is increased in magnitude order, the chance of encryption algorithm to collide also increases. For example, a hash function has no awareness of other values in the set of inputs. It just executes some mathematics or logic

Table 3: Feature Correlation and Classification Results on Shifted Synthetic Data Based on Accuracy Percentage (%).

Dataset	Independent Features	Dependent Features ($y\|x$)		Absolute Correlation Score	Decision Tree	Support Vector Machine
		y	x			
Fertility	season	alcohol-frequency	trauma	0.7735	98.00	92.00
		recent-fevers	surgical-intervention	0.9027		
		child-diseases	age	0.1455		
		num-hours-sitting-pd	smoking-habit	0.9332		
Iris	sepal-length	petal-length	petal-width	0.9589	97.33	100.00
Haberman	age-at-operation	year-of-operation	positive-nodes	0.1454	70.92	93.13
Liver	gammagt	sgot	sgpt	0.9715	88.98	78.55
		drinks	gammagt	0.2840		
Breast	cell-shape	mitoses	clump-thickness	0.3776	98.86	100.00
		epithelial-cell-size	cell-size	0.7523		
		bare-nuclei	marginal-adhesion	0.6620		
		normal-nucleoli	bland-chromatin	0.6253		
Pima	pregnant	insulin	skin-thickness	0.9688	98.43	88.02
		bmi	blood-pressure	0.6718		
		age	plasma	0.2041		
Thyroid	hypopituitary query-hyperthyroid	T4U	age	0.7755	99.97	99.92
		on-thyroxine	sex	0.8218		
		lithium	query-on-thyroxine	0.5565		
		psych	antithyroid-medication	0.5396		
		goitre	sick	0.3171		
		TT4	pregnant	0.9647		
		I131-treatment	thyroid-surgery	0.3885		
		TBG-measured	query-hypothyroid	-		
		TT4-measured	tumor	0.6788		
		TBG	TSH-measured	-		
		FTI	TSH	0.9835		
		T4U-measured	T3-measured	0.2952		
		referral-source	T3	0.0634		
Bank	age	previous	pdays	0.9949	99.95	88.31
		month	contact	0.9706		
		education	job	0.9704		
		campaign	day	0.8961		
		housing	marital	0.1845		
		loan	balance	0.7807		
		poutcome	duration	0.0335		

operations on the input values that are passed to it. Thus, there's always a chance that two or more different inputs will generate the same hash value, creating a pattern that can be used to deduce the encryption key. Once the key has been determined, extracting values from encrypted data is a straightforward process.

The hash collision probability for encryption is visualized in Figure 5. The graph produces an S-curve shape. This does not happen in our case, because, the shifted synthetic dataset is largely different in contrast to the original dataset, as discussed in Section 4.4. Our method also has no need for securing the key in order to protect the data. In Table 2 and Table 3, it can also be observed that the features' relations have undergone significant changes. For example, if one deduced that *marital* status can be determined based on the *housing* value in shifted synthetic **Bank** dataset, then, it is not entirely correct, as in the original dataset, *marital* is highly dependent on the *age* value.

However, there are some drawbacks, at least during the execution and experimentation results. During the computation, the process to estimate each independent variable distribution function is very fast. However, the process becomes very slow due to computing-intensive operation and large memory requirement when estimating the probability distribution function of a set of several dependent variables. This is why we resort to using simple linear regression model to automatically generate y value in $P(y|x)$ relation.

Through classification task, we also managed to prevent any data utility degradation on the new shifted synthetic dataset. However, unnecessary bias has been introduced to the dataset, making the classification accuracy significantly improved over the original datasets. This effect is probably due to the linear regression model that is used in generating the y value in $P(y|x)$ relation. As it can been seen

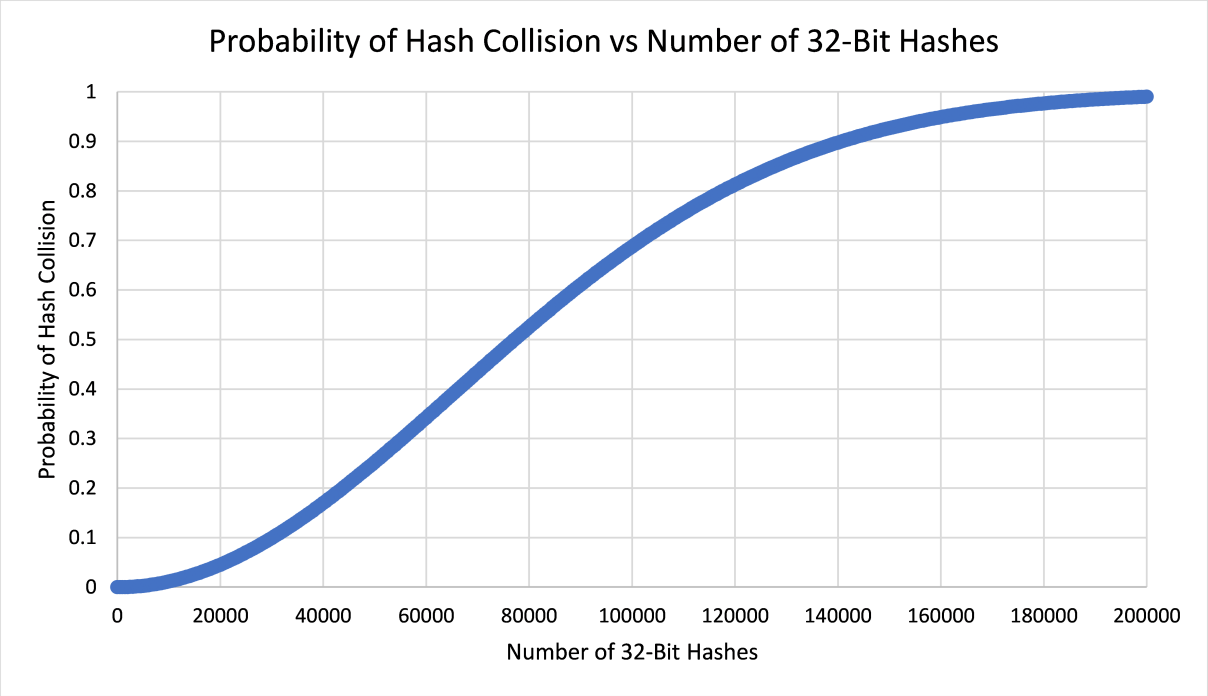

Figure 5: Illustrates the probability of collision when using 32-bit hash values.

in Figure 3 and Figure 4, an obvious and consistent pattern can been observed for $P(y|x)$ relation. This could be the reason why the classification performance in shifted synthetic dataset is way more higher compared to original dataset.

Despite of the drawbacks, we can conclude that our privacy-preserving approach did achieve the privacy properties defined in Section 2. The shifted synthetic dataset provides anonymity, unlinkability, pseudonymity and deniability such as follows:

(1) It provides *anonymity* through new synthetic dataset, as it differs completely from original dataset.

(2) It provides *unlinkability* by random value generation of each field in each record, hence reducing the probability of linking several records to each single entity.

(3) It provides *deniability*, since the original and shifted synthetic dataset is totally different, hence allowing user to deny any claim for the synthetic record to be linked to that user.

(4) And, it also provides *pseudonymity*, in the sense that only data owner, for example, an Internet Service Provider, can define how the data was generated, while data user can only see the shifted synthetic data.

It is important to note that the privacy properties can only be achieved if the assumptions defined in Assumption 1 and Assumption 2 are properly adhered to. If either of this assumption is disregarded, then, the task of breaching data owner privacy became a trivial process. Our method relies on the obscurity of the generative model, but with many ways in customizing each individual task in the model, such as in feature selection task, shifting task, and distribution estimation task.

6 Related Works

Privacy-preserving task is usually based on syntactic privacy protection. The easiest albeit computationally expensive process is through encryption process on the whole dataset, transforming it into complicated-to-understand values. It is straightforward, however, it provides almost zero data utility value, especially for data analytics task.

Another method is to hide certain value in the dataset from being accessed by others. Certain field values such as personal identification data like *name*, *age* and *address* can be suppressed from the data user. However, simply hiding these data is not sufficient. A combination of other fields or other records can be used to infer the hidden data. For example, if there is a field that is highly associated with age value, such as *age requirement*, then one could simply deduce the approximate value of *age*. Regardless, there will be often correlation that can be inferred, across fields and records with thorough statistical analysis. In order to solve this data suppression limitation, data suppression and generalization is usually performed based on k-anonymity and l-diversity methods [18]. In the k-anonymity method, the granularity of data representation is reduced with the use of techniques such as generalization and suppression, where, l-diversity is used to solve some k-anonymity such as prior knowledge attack and homogeneity attack. Nevertheless, the data suppression will slightly, if not significantly, reduce the data utility.

Data perturbation is another method that can be implemented to satisfy the privacy constraint. Here, the data is distorted in order to mask the original distribution, which is accomplished by the alteration of an attribute value by a new value [9]. For example, a value can be changed from 0 to 1 in order to deceive data user, thus, preventing further re-identification process. Despite of that, data perturbation process is a sensible process if the distortion function is properly defined. This is because, blindly distorting the data will affect the statistical structure of the data itself, hence resulting in uncertain relation being learned by data modeler. This will greatly reduce the value of data utility, if for example, a valid record is transformed into new statistically incomprehensible relation within the new dataset. Data modeler for example will treat this record as outlier or noise that should be removed from the dataset.

Data randomization is also one of the privacy preserving methods. It comprises records swapping or fields swapping or combination of both [27]. However, not only the statistical structure of the data is notably broken, but, it is also a computationally demanding task, depending on the selection of data randomization algorithm in the first place. In addition to that, misrepresentation of data due to randomize privacy preserving process will only increase the data modelling computing time in order to come out with sensible model on that data. Furthermore, tracking how the randomization process is done is a very tedious task, so that the relation between original dataset and new synthetic dataset is properly clarified and preserved.

Synthetic generator mechanism is another approach that has start to gain traction in recent literature for privacy-preserving task [36, 29, 4, 25]. The synthetic generator mechanism is used with respect to its input data as a basis in generating new sensible privacy-preserving synthetic dataset. While it increases the difficulty for data modeler when modelling the dataset, performing the task on the dataset that are generated by synthetic generator mechanism is a feasible task, compared to syntactic privacy protection approach.

With respect to that, there are two different models in synthetic generator mechanism, which is either a generative based model or evaluative based model. Generative based model is a model that emphasize on how to systematically generate a new synthetic data [36, 25], while evaluative based model is designed to test whether the new synthetic data pass or fail the privacy test [29, 4]. In evaluative based model, the mechanism of generating new synthetic data is loosely defined. Table 4 describes the existing synthetic generator mechanism approaches in the literature.

Our method is in parallel with generative based model. This is because, our model is derived through statistical framework to generate new synthetic data that is totally different from original data, while

taking care of the relation of each feature in the dataset through individual feature analysis.

Table 4: Descriptive Comparison on Existing Synthetic Generator Mechanism Model.

Approach	Model	Description
A method to release differential privacy based dataset using deep learning based generative architecture [36]	Generative	Through deep generative model, known as dp-GAN, a new semantically preserved privacy-preserving synthetic data is generated a for arbitrary analysis tasks, based on semantic rich data. The utility of the data is preserved through scalable multi-fold optimization strategies. Therefore, since the process is scalable (due to the nature of deep learning architecture itself), deriving new synthetic dataset from very large dataset is a very fast process.
A method to generated new synthetic dataset based on estimated shifted dataset distribution function [25]	Generative	Within a supervised learning framework (classification task), this approach estimate the probability conditional distribution between covariates and class, modify the characteristics of the estimated probability distribution and generate new synthetic data based on the changed (shifted) distribution. Our work simplify this approach by removing the covariate-class $p(x,y)$ relationship, and demonstrated the usability of this approach through various datasets through simulated experiment on classification task.
Bayesian estimation of disclosure risks for multiply imputed, synthetic data [29]	Evaluative	A framework to estimate disclosure risks in multiply-imputed synthetic data based on Bayesian model is designed. This framework is used to measure how good the privacy property on multiply-imputed synthetic data, record by record. If one record has high risk of disclosure, one can treat that record again, so the risk of disclosure is reduced.
A definition of plausible deniability for evaluating newly generated privacy-preserving synthetic data [4]	Evaluative	A framework that introduces privacy parameters to describe several formal privacy guarantee are proposed before any privacy-preserving synthetic dataset is released to third parties. The framework is known as plausible deniability measure.

Nevertheless, despite of synthetic generator mechanism capability in providing approximately similar level of data utility as the original dataset, the mechanism of producing those synthetic data itself is vaguely understood. Moreover, syntactic privacy protection provides clear and tangible privacy preservation effect compared to synthetic generator mechanism, although at the cost of data utility in a significant way. Hence, it is understandable on why syntactic privacy protection is widely used to satisfy the privacy constraint.

7 Conclusion and Future Works

In this paper, we have proposed a method to preserve data privacy through synthetic dataset shift for preserving data mining task without degrading the performance of data utility. The proposed method

involves several statistical processes to perform two main tasks, feature selection, and distribution estimation of shifted data.

Experimentation on 8 standard UCI datasets shows that this approach can be used for preserving the privacy by producing new synthetic data that is totally different from the original data, as well as changing the composition of independent (and the relation of dependent) features from original data to the new shifted synthetic data. Using the case of the classification task, we demonstrate that data utility is similar (not less) after the distribution shift. We have shown that, (1) the synthetic data is statistically significantly different when compared to the original data, hence the privacy of the data is preserved, and (2) the utility of the synthetic data is preserved.

Finally, this work requires that the middle man, which is the data keeper, can be trusted by data owner and data user, to store and preserve the privacy of data owner, while sharing the data with data user. Hence, in the future, we would like to explore on how to formally enforce the trust on data keeper so data owners trust their data is in good hands. In addition to that, we would like to decrease or remove the obscurity of our model, so it will become no issue if the generative model is leaked to the third party.

Furthermore, we also would like to maintain the performance of data utility without degrading the data utility but also without introducing unnecessary bias into the dataset. As it can been seen in this paper, one of the data utility metric, the classification accuracy, is significantly improved in shifted synthetic datasets, compared to original datasets. Hence, in the future, it is appropriate to control this bias by providing a mechanism for user to use and tune the bias introduced in the dataset.

Acknowledgement

This research article was funded by Ministry of Higher Education of Malaysia, under Fundamental Research Grant Scheme (FRGS/1/2019/ICT02/UUM/02/2) with S/O code 14358.

References

[1] M. M. Adankon and M. Cheriet. Support vector machine. In *Encyclopedia of biometrics*, pages 1303–1308. Springer-US, 2009.

[2] K. Bache and M. Lichman. Uci machine learning repository, 2013. `https://archive.ics.uci.edu/ml/index.php` [Online: Accessed on November 15, 2020], 2013.

[3] J. Ball. Us and uk struck secret deal to allow nsa to'unmask'britons' personal data. `https://www.theguardian.com/world/2013/nov/20/us-uk-secret-deal-surveillance-personal-data` [Online: Accessed on November 15, 2020], November 2013.

[4] V. Bindschaedler, R. Shokri, and C. A. Gunter. Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment*, 10(5):481–492, January 2017.

[5] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, et al. Orange: data mining toolbox in python. *The Journal of machine Learning Research*, 14(1):2349–2353, January 2013.

[6] C. Dwork. Differential privacy: A survey of results. In *Proc. of the International Conference on Theory and Applications of Models of Computation, Xi'an, China*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer-Heidelberg, April 2008.

[7] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of the Theory of Cryptography Conference, New York, NY, USA*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer-Heidelberg, March 2006.

[8] D. Eastlake and P. Jones. Us secure hash algorithm 1 (sha1). `https://www.hjp.at/(st_a)/doc/rfc/rfc3174.html` [Online: Accessed on November 15, 2020], September 2001.

[9] G. Giaconi, D. Gündüz, and H. V. Poor. Smart meter privacy with renewable energy and an energy storage device. *IEEE Transactions on Information Forensics and Security*, 13(1):129–142, January 2018.

[10] D. Gil, J. L. Girela, J. De Juan, M. J. Gomez-Torres, and M. Johnsson. Predicting seminal quality with artificial intelligence methods. *Expert Systems with Applications*, 39(16):12564–12573, November 2012.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, November 2009.

[12] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, April 1999.

[13] B. Heitmann, J. G. Kim, A. Passant, C. Hayes, and H.-G. Kim. An architecture for privacy-enabled user profile portability on the web of data. In *Proc. of the First International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec'10), Barcelona, Spain*, pages 16–23. ACM, September 2010.

[14] S.-C. Huang. Using gaussian process based kernel classifiers for credit rating forecasting. *Expert Systems with Applications*, 38(7):8607–8611, July 2011.

[15] S. P. Kasiviswanathan and A. Smith. A note on differential privacy: Defining resistance to arbitrary side information. *arXiv preprint arXiv:1801.01594*, March 2008.

[16] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *Proc. of the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD'11), Athens, Greece*, pages 193–204. ACM, June 2011.

[17] A. Lee. Towards informatic personhood: understanding contemporary subjects in a data-driven society. *Information, Communication & Society*, pages 1–16, July 2019.

[18] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey*, pages 106–115. IEEE, April 2007.

[19] S. Moro, R. Laureano, and P. Cortez. Using data mining for bank direct marketing: An application of the crisp-dm methodology. In *Proc. of the 25th European Simulation and Modelling Conference (ESM'11), Guimaraes, Portugal*, pages 117–121. Eurosis-ETI, October 2011.

[20] NIST-FIPS. Announcing the advanced encryption standard (aes). Technical report, National Institute of Standards and Technology (NIST), November 2001.

[21] T. A. O'Brien, K. Kashinath, N. R. Cavanaugh, W. D. Collins, and J. P. O'Brien. A fast and objective multidimensional kernel density estimation method: fastkde. *Computational Statistics & Data Analysis*, 101:148–160, September 2016.

[22] P. O'Day. Nsa surveillance: How it's happening and why you should care. *Interface: The Journal of Education, Community and Values*, 13(inter13):239–244, December 2013.

[23] G. Perrin, C. Soize, and N. Ouhbi. Data-driven kernel representations for sampling with an unknown block dependence structure under correlation constraints. *Computational Statistics & Data Analysis*, 119:139–154, March 2018.

[24] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. `http://www.maroki.de/pub/dphistory/2010_Anon_Terminology_v0.34.pdf` [Online: Accessed on November 15, 2020], August 2010.

[25] M. S. M. Pozi, A. A. Bakar, R. Ismail, S. Yussof, A. R. Fiza, and R. Ramli. Shifting dataset to preserve data privacy". In *Proc. of the 2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e'18), Langkawi Island, Malaysia*, pages 134–139. IEEE, November 2018.

[26] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, June 2014.

[27] M. A. Rahman, M. H. Manshaei, E. Al-Shaer, and M. Shehab. Secure and private data aggregation for energy consumption scheduling in smart grids. *IEEE Transactions on Dependable and Secure Computing*, 14(2):221–234, June 2017.

[28] M. Reis, A. Garcia, and R. J. Bessa. A scalable load forecasting system for low voltage grids. In *Proc. of the 2017 IEEE Manchester PowerTech, Manchester, UK*, pages 1–6. IEEE, July 2017.

[29] J. P. Reiter, Q. Wang, and B. Zhang. Bayesian estimation of disclosure risks for multiply imputed, synthetic

data. *Journal of Privacy and Confidentiality*, 6(1), June 2014.

[30] S. Scardapane, R. Altilio, V. Ciccarelli, A. Uncini, and M. Panella. Privacy-preserving data mining for distributed medical scenarios. In *Multidisciplinary Approaches to Neural Computing*, volume 69 of *Smart Innovation, System, and Technologies*, pages 119–128. Springer-Cham, August 2018.

[31] D. E. Seidl, G. Paulus, P. Jankowski, and M. Regenfelder. Spatial obfuscation methods for privacy protection of household-level data. *Applied Geography*, 63:253–263, September 2015.

[32] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proc. of the 21st Annual Conference on Neural Information Processing Systems (NIPS'07), Vancouver, BC, Canada*, pages 1433–1440. NeurIPS, December 2008.

[33] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, May 2002.

[34] D. Vatsalan, Z. Sehili, P. Christen, and E. Rahm. Privacy-preserving record linkage for big data: Current approaches and research challenges. In *Handbook of Big Data Technologies*, pages 851–895. Springer-Cham, February 2017.

[35] H. Wang, Q. Xu, and L. Zhou. Seminal quality prediction using clustering-based decision forests. *Algorithms*, 7(3):405–417, May 2014.

[36] X. Zhang, S. Ji, and T. Wang. Differentially private releasing via deep generative model. *arXiv preprint arXiv:1801.01594*, January 2018.

[37] Z. Zhang, Z. Qin, L. Zhu, J. Weng, and K. Ren. Cost-friendly differential privacy for smart meters: exploiting the dual roles of the noise. *IEEE Transactions on Smart Grid*, 8(2):619–626, June 2017.

Author Biography

Muhammad Syafiq Mohd Pozi is an academician and a machine learning researcher in School of Computing, Universiti Utara Malaysia. He received the Bachelor of Computer Science with Honours from Infrastructure University of Kuala Lumpur, Malaysia in 2012. He received his Doctor of Philosophy in Computer Science from Universiti Putra Malaysia in 2016. His research interest is on modelling uncertainty in machine learning, optimization, computer vision and natural language processing. His research collaboration spans across various research domains, such as network security, social science and lately, medicine. He is a member of ACM.

Mohd. Hasbullah Omar is currently an Associate Professor at the School of Computing, Universiti Utara Malaysia. He received the Bachelor of Engineering with Honours in Electronics, Telecommunication and Computer Engineering from University of Bradford, UK in 1999. Then, he received his Master and Doctor of Philosophy in Information Technology from Universiti Utara Malaysia, in 2002 and 2011 respectively. His research interest includes communication protocols, mobile network technology and sensor networks. He is a member of IEEE.