IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 June 2022, accepted 20 September 2022, date of publication 22 September 2022, date of current version 30 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208715

RESEARCH ARTICLE

Machine Learning Model Generation With Copula-Based Synthetic Dataset for Local Differentially Private Numerical Data

YUICHI SEI[1,2], (Member, IEEE), J. ANDREW ONESIMU[3], AND AKIHIKO OHSUGA[1], (Member, IEEE)

[1]Department of Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Tokyo 182-8585, Japan
[2]JST, PRESTO, Kawaguchi, Saitama 102-0076, Japan
[3]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

Corresponding author: Yuichi Sei (seiuny@uec.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP18H03229, Grant JP18H03340, Grant JP18K19835, Grant JP19K12107, Grant JP19H04113, and Grant JP21H03496; and in part by the Japan Science and Technology Agency (JST), Precursory Research for Embryonic Science and Technology (PRESTO), under Grant JPMJPR1934.

ABSTRACT With the development of IoT technology, personal data are being collected in many places. These data can be used to create new services, but consideration must be given to the individual's privacy. We can safely collect personal data while adding noise by applying differential privacy. However, because such data are very noisy, the accuracy of machine learning trained by the data greatly decreased. In this study, our objective is to build a highly accurate machine learning model using these data. We focus on the decision tree machine learning algorithm, and, instead of applying it as is, we use a preprocessing technique wherein pseudodata are generated using a copula while removing the effect of noise added by differential privacy. In detail, the proposed novel protocol consists of three steps: generating a covariance matrix from the differentially private numerical data, generating a discrete cumulative distribution function from differentially private numerical data, and generating copula-based numerical samples. Simulation results using synthetic and real datasets verify the utility of the proposed method not only for the decision tree algorithm but also for other machine learning algorithms such as deep neural networks. This method will help create machine learning models, such as recommendation systems, using differential privacy data.

INDEX TERMS Copula, data mining, decision trees, local differential privacy, machine learning, privacy-preserving data collection.

I. INTRODUCTION

Personal data can be collected to create machine learning models that can be employed by law enforcement agencies to profile suspects, by companies to predict performance once a job seeker is hired, and so on [1], [2]. However, we need to consider privacy and fairness when creating such machine learning models [3], [4].

In this study, we assume a scenario wherein a relatively small amount of data (e.g., data of less than 10,000 people) that has been collected under the application of local differential privacy [5], the de facto standard privacy metric [6], [7] (Fig. 1), already exists. In Fig.1, each person sends their personal attribute data, such as age, location, and medical data, to a model generator's server. All data is protected using local differential privacy techniques. Therefore, the model generator cannot know the true value of each data attribute. From the stored differentially private data, the model generator trains a machine learning model.

Local differential privacy is a specialized concept of differential privacy, especially for data collection from each person. Many companies, such as Apple and Google, have used local differential privacy, which can provide a strict privacy guarantee against adversaries with arbitrary background

The associate editor coordinating the review of this manuscript and approving it for publication was Wei-Yen Hsu.

knowledge [8]. Laplace noise is added to numerical data [9] as a general method to realize local differential privacy. The amount of noise is controlled by the privacy budget ϵ. In the model of local differential privacy, each person sends their obfuscated data to the data collector. The data collector can only see the obfuscated data and cannot know the true value for each individual. Laplace noise is added to numerical data [9] as a general method for realizing local differential privacy.

We focus on creating a decision tree, which is a well-known machine learning algorithm. Although we are in the era of deep learning, decision trees are still widely used and studied [10], [11], [12], [13], [14]. In terms of accuracy, decision trees achieve inferior results compared to those of deep learning; however, there are many advantages to using decision trees, including high human interpretability and non-parametric design [15], [16], [17].

Recently, differentially private decision tree generation algorithms have been widely proposed [17], [18], [19], [20]. Existing studies target differentially private decision tree generation from original (non-privatized) data. In this study, we aim to generate decision trees not from the original data but from locally stored differentially private data.

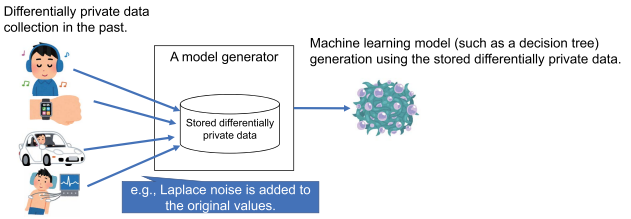


FIGURE 1. Scenario of this study: A model generator, which stores differentially private data, generates a high-quality machine learning model using the data.

In this study, we use a copula employed in economic and financial applications [21]. However, general copula algorithms do not consider differentially private data. Although differential privacy can protect each person's privacy, it adds extensive noise to the data. Therefore, the accuracy of the general copula model created from differentially private data becomes low, as the experimental results of Copula+DT in Section VI show. We propose a novel preprocessing technique wherein a synthetic dataset is generated using a copula while removing the effect of noise added by differential privacy. In detail, the proposed novel protocol consists of three steps: generating a covariance matrix from the differentially private numerical data, generating a cumulative distribution function of a discrete probability distribution from differentially private numerical data, and generating copula-based numerical samples from the covariance matrix and the discrete cumulative distribution function. The generated dataset is used to train machine-learning models. The results of experiments conducted on synthetic and real datasets demonstrate that our method significantly increases model accuracy. We focus on the decision tree algorithm in this study; however, we also conducted simulations for deep neural network (DNN), k-nearest neighbors (kNN), and support vector machine (SVM) using our proposed method. The results indicate that it increases the model accuracy of DNN and kNN with relatively few attributes but does not do so for SVM.

Table 1 presents the main notations used in this paper.

TABLE 1. Notations.

n	Number of data samples
Q_i	ith attribute of data
X_i	Random variable representing Q_i
g	Number of attributes
Δ_i	Difference between the maximum and minimum values of Q_i
ϵ_i	Privacy budget for differential privacy for Q_i
b	Number of bins of an input domain for calculating a cumulative distribution function
e	Base for natural logarithm.

Our contributions in this paper are summarized as follows. First, we introduced the relationship between the variance and the covariance of differentially private numerical data and those of original data. Second, we developed an algorithm to generate a copula model based on the estimated variance and covariance. Third, we developed an algorithm to convert the discrete cumulative distribution function into a continuous cumulative distribution function in the copula space to generate a high-quality machine learning model. Finally, we evaluated the performance of the proposed method using synthetic and real datasets.

The remainder of this paper is organized as follows. The assumptions of this study are described in Section II. Differential privacy, decision trees, and related research are discussed in Section III. Section IV analyzes the effect of differential privacy on decision trees. The proposed solution is introduced in Section V. The evaluation conducted is presented in Section VI, and the evaluation results are discussed in Section VII. Finally, we conclude this paper in Section VIII.

II. ASSUMPTION

A. TARGET SCENARIO

We call the organization generating the machine learning model the *model generator*. Many techniques can be employed for differentially private machine learning model generation. These techniques can be divided into three categories. In the first category, the model generator is assumed to store the original (i.e., non-privatized) personal data. The model generator is a trusted entity, and the generated models are shared with untrusted third parties. Many studies on differentially private decision trees fall into this category [17], [18], [19], [20]. We assume that the model generator is a semi-honest entity in this study; therefore, it cannot have direct access to the original personal data. The second and third categories also make this assumption. The difference between the second and third categories hinges on whether

or not the model generator has indirect access to the original personal data when generating machine learning models.

The second category is technology that does not access the original personal data, but instead uses the stored differential private data. Our study takes this assumption as its basis. There are several possible reasons why the model generator would store such data.

- The organization stores the data now for future use: Several organizations collect and store data while protecting the privacy for future use [22], [23], [24]. When they decide to generate machine learning models, they use the stored data.
- The organization stores the data for efficient machine learning model generation: Training data are very important for model debugging and analyzing the performance of trained model [25], [26]. Even if hyperparameters and a model structure are determined, training data are necessary to ensure efficient model generation.
- The organization stores the data for fairness auditing: The problem of the biased output of machine learning models for sensitive personal attributes such as race and gender has been widely recognized as a fairness problem in machine learning. An analysis of training data is required [27], [28], [29] to audit fairness or generate fair machine learning models. Moreover, what constitutes a bias depends on the attitudes of people and, therefore, it may change in the future. Therefore, it is necessary to store training data to cope with future changes.

In the third category, it is assumed that original personal data can be accessed indirectly when machine learning models are generated. In this category, the model generator trains machine learning models by collaborating with many personal data holders. Each personal data holder sends the information to train the machine learning models while protecting the privacy of their personal data. This approach has several merits; however, the model generator needs to obtain the information necessary to update the parameters of the machine learning model from other organizations that store personal data when the model generator trains machine learning models; it cannot ensure future access to data. Federated learning techniques fall into this category; there are many federated learning techniques for deep neural networks, and there are few techniques for decision trees. Although such techniques have been studied widely, we focus on the second category in this research for the reasons described above.

B. TARGET MACHINE LEARNING MODEL

In the past decade, deep neural networks have been extensively studied. However, it is difficult to understand the reason for the output of deep neural networks. Although there are several techniques for understanding the black box of deep neural networks, many issues still need to be resolved [30].

In contrast, the decision tree algorithm, which is one of the most popular machine learning algorithms, has high human interpretability. The main drawbacks of this algorithm are its tendency to overfit the data and its instability when small changes occur in the data; however, they can be minimized by limiting the depth of the tree, pruning unreliable leaf nodes, building ensembles instead of a single tree, etc. [17].

Although our proposed method can be applied to any machine learning algorithm, it is most effective for the decision tree algorithm. This is because all differential privacy data have a large noise, and decision trees overfit such noises. However, we show the results of applying the proposed method to DNN, kNN, and SVM in Section VI to demonstrate the adaptability of the proposed method to other machine learning algorithms.

C. TARGET DATA TYPE

We focus on numerical data in this paper because numerical data can be easily converted into categorical data; that is, numerical data are more useful than categorical data. For time series data, the proposed method can be implemented at each point in time. We can treat image data by applying the proposed method to each pixel of each image. However, in that case, its utility would significantly decrease because one image is composed of numerous pixels. Applying and evaluating other data types is a future task.

III. RELATED WORK

A. DIFFERENTIAL PRIVACY

Differential privacy is considered the most important privacy metric [31]. In machine learning algorithms such as deep neural networks and decision trees, differential privacy has been studied extensively in the past decade [9], [17]. Differential privacy is used for the central model, i.e., the anonymizer holds all original data (the first category was introduced in Section II). In contrast, local differential privacy assumes a local model, i.e., each person privatizes their values locally. In this paper, ''differential privacy'' refers to ''local differential privacy.''

Let X, Y, and M represent a domain of personal data, set of privatized data, and a privacy mechanism that takes $x \in X$ and outputs $y \in Y$, respectively.

Definition 1 (ϵ-Local Differential Privacy): Let $\epsilon > 0$. M satisfies ϵ-local differential privacy if, for every $x, x' \in X$ and $y \in Y$,

$$Pr(M(x) = y) \leq e^{\epsilon} Pr(M(x') = y). \tag{1}$$

Many differential privacy methods use the Laplace mechanism for numerical attributes [9]. Here, X represents numerical values. Let Δ represent the difference between the maximum and minimum values of X. The Laplace mechanism adds noise drawn from the Laplace distribution with a mean of zero and scale Δ/ϵ.

Much of the work on local differential privacy, such as [32], [33], [34], is primarily aimed at generating histograms of attribute values. If the generated histogram can achieve a sufficiently high accuracy, then it can predict the output value from the input value as well as the machine learning model. In Section VI, we compare the proposed method with the state-of-the-art methods [32].

B. GENERATING DIFFERENTIALLY PRIVATE MACHINE LEARNING MODELS FROM ORIGINAL DATA

Many researchers assume that the model generator has original (non-privatized) data, and they propose algorithms that generate differentially private machine learning models using these original data [35], [36], [37], [38], [39], [40], [41]. In this study, we assume that the model generator is not an honest entity, and that other methods are required.

C. GENERATING MACHINE LEARNING MODELS AFTER COLLECTION OF LOCAL DIFFERENTIALLY PRIVATE DATA

The proposed method is categorized into this approach. The generation of a machine learning model after collecting local differentially private data is considered a baseline approach in existing studies [42]. This is because it is difficult to increase model accuracy after the collection of local differentially private data. They generate deep neural network models using a set of local differentially private data; that is, they do not attempt to propose better algorithms. There are scenarios wherein the model generator has a set of local differentially private data but cannot access the original data directly or indirectly. Therefore, it is important to propose an algorithm that generates a high-accuracy machine learning model using a set of locally differentially private data that are already stored.

D. GENERATING MACHINE LEARNING MODELS WITH INDIRECT ACCESS TO ORIGINAL DATA

In this decade, federated learning techniques have been widely studied [43], [44], [45], especially for deep neural networks. Distributed data owners exist in federated learning, and the model generator generates a machine learning model without direct access to the data. The privacy of all data can be protected to some extent because the model generator does not directly access the personal data. Each data owner does not send the original data but sends the model gradient information or some other information to the model generator. However, recent studies have argued that there is a risk of privacy leakage in the gradient information of the model [46]. Hence, differentially private federated learning algorithms have been proposed [47], [48], [49], [50], [51], [52], [53]. These techniques can achieve high utility and privacy simultaneously.

In recent years, the shuffle model for differential privacy has attracted considerable attention [54]. The shuffle model can reduce noise added by local differential privacy; we should assume that a perfectly secure primitive exists [55]. Moreover, the model generator cannot store local differentially private data.

E. DIFFERENTIALLY PRIVATE SYNTHETIC DATA GENERATION

There are many methods for generating a differentially private dataset from original (non-privatized) data samples [56], [57], [58], [59]. These methods assume that the server has original data samples, and the goal is to generate and share a synthetic dataset that is similar to the data samples. In contrast, we assume that the server does not have original data samples, and we aim to generate a machine learning model from differentially private data samples.

We summarize the various perspectives of each type of generated machine-learning model with differential privacy and differentially private synthetic data generation in Table 2. Our target is the second category, where the model generator does not have original data but owns differentially private data.

F. COPULA

A covariance matrix Σ of all attributes and a cumulative distribution function F_j of each attribute Q_j are used to generate a copula model.

Samples based on a normal distribution with covariance matrix Σ are generated to generate random samples from the copula model. Let $s_{i,j}$ represent the jth attribute value of person i, and let the set of samples be $\{s_1, \ldots, s_n\}$, where $s_i = \{s_{i,1}, \ldots, s_{i,g}\}$, n represents the number of data samples and g represents the number of attributes.

Then, we divide each value $s_{i,j}$ of the samples by the standard deviation of each attribute σ_j. That is, for all i, j,

$$s_{i,j} \leftarrow s_{i,j}/\sigma_j. \tag{2}$$

The values of the cumulative distribution function of a standard normal distribution for each value of the samples were calculated;

$$t_{i,j} = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{s_{i,j}}{\sqrt{2}}\right)\right]. \tag{3}$$

Then, we obtain the corresponding value of each attribute from $t_{i,j}$. More specifically, for all i, j, we calculate

$$u_{i,j} = F_j^{-1}(t_{i,j}) \tag{4}$$

where F_j^{-1} is an inverse function of F_j.

The resulting $u_i = \{u_{i,1}, \ldots, u_{i,g}\}$ is a generated sample.

Rocher *et al.* [60] proposed a method to predict the number of people with a certain combination of attribute values in a population from a small sample using copula. They use mutual information to compute the copula; however, they do not use differential privacy or other privacy measures, i.e., they assume that they have original (non-privatized) data. Moreover, they did not predict the value of an attribute. Copula-based data synthesis has been studied to generate perturbed data while preserving the surrounding distribution of the data, which can be used to train machine learning models [61], [62], [63]. However, they do not consider differentially private data. Recent studies generate Copula models based on differentially private data [64], [65]. However, they do not target numerical data. Moreover, their aims are not to train machine learning models, but to re-identify individuals from incomplete datasets and generate histograms.

TABLE 2. Various perspectives of related work.

Category	Available data	Owner of the available data
Generating differentially private machine learning models from original data	Original data	Model generator
Generating machine learning models after collection of local differentially private data	Differentially private data	Model generator
Generating machine learning models with indirect access to original data	Differentially private information of original data	Other parties
Differentially private synthetic data generation	Original data	Model generator

IV. ANALYSIS OF DECISION TREE WITH LOCAL DIFFERENTIALLY PRIVATE DATA

A decision tree is a method for analyzing data using a tree structure. Each internal node represents a rule for data splitting.

There are several algorithms for generating decision trees, such as classification and regression trees (CART) [66], iterative dichotomiser 3 (ID3) [67], and C4.5 [68].

For regression decision tree algorithms, the mean squared error (MSE) is used to find the optimal splitting point of each attribute. The goal is to find the attribute and its splitting point that reduces the weighted average of the MSE of the child nodes to its lowest value. The MSE of node X_i is calculated as

$$MSE_i = \sum_j \frac{(X_{i,j} - \overline{X_i})^2}{k_i} \tag{5}$$

where $X_{i,j}$ represents the jth value at X_i, $\overline{X_i}$ represents the mean value of node X_i, and k_i represents the number of values of node X_i. The weighted average of the MSE of two child nodes X_i and $X_{i'}$ is calculated as

$$\text{Weighted average of MSE} = \frac{k_i MSE_i}{k_i + k_{i'}} + \frac{k_{i'} MSE_{i'}}{k_i + k_{i'}}. \tag{6}$$

It is difficult for the decision tree algorithm to split the tree properly when Laplace noise is added to each data sample because the MSE cannot be calculated correctly because of the noise. Figs. 2a–2c show an example where the split does not work. Here, a Boston dataset [69], [70] was used. This dataset comprises 13 feature attributes, e.g., per capita crime rate by town, and an objective attribute (median value of owner-occupied homes). Fig. 2a depicts the relationship between the split point of the per capita crime rate by town and the corresponding weighted average of the MSE. It is in the shape of a convex downward, and the weighted average of the MSE is minimized when the split point is set to seven. Figs. 2b and 2c depict cases where each data sample is collected under differential privacy. We set ϵ to five. In Fig. 2b, the weighted average of the MSE is minimized when the split point is set to 37, and in Fig. 2c, the weighted average of the MSE is minimized when the split point is set to 59. Each run yields completely different results because the amount of noise is stochastic under differential privacy, as shown in these figures. Further, regardless of the split point, the overall value of the weighted average of the MSE is considerably larger than that of the original value in Fig. 2a. Thus, it becomes very difficult to determine the correct split point when the noise of differential privacy is added to all data samples. This leads to difficulties in generating an accurate decision tree model under differential privacy.

On the other hand, Figs. 2d and 2e shows the results for the pseudo data generated by the proposed method. Because our proposed method generates pseudo data that preserve the statistical trend of each attribute, the shapes in Figs. 2d and 2e are similar to the original shape in Fig. 2a. The splitting points are eight and three, respectively, which are close to the optimal splitting point of five.

Moreover, the correlation information of attributes is destroyed in the differentially private data. Therefore, when creating a decision tree from differentially private data, the deeper the node is, the more significant the effect of the error becomes. In contrast, the pseudo dataset based on the proposed method reconstructs the correlation information of attributes. Therefore, even when the nodes are deeper, the deterioration of the accuracy of the decision tree can be suppressed.

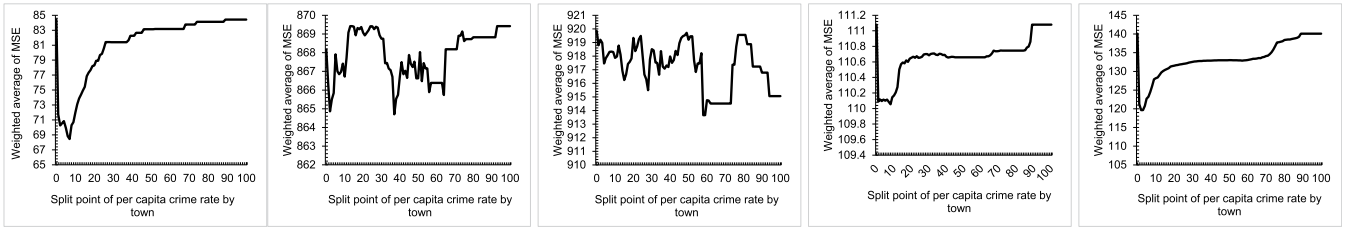
V. PROPOSED METHOD

Let $\mathcal{L}(x; \mu, s)$ represent the Laplace probability density function with mean μ, scale s, and a random variable $x \in X$. When the mean μ is zero, we use $\mathcal{L}(x; s)$.

A. OUTLINE

Copula-based data synthesis has been researched to produce perturbed data and incorporate rich statistical information in the perturbed data. The proposed protocol is developed in three steps: 1) generate a covariance matrix from the differentially private data (Section V-B), 2) generate a cumulative distribution function (Section V-C), and 3) generate copula samples (Section V-D). The generated copula samples are used to train the machine learning model. The algorithms used in all the steps were developed in this study. Overview of the proposed method is shown in Fig. 3.

In the first step, we introduce the relationship between the variance and the covariance of differentially private data and those of original data. The model generator cannot access the original data; however, the proposed method can estimate the variance and the covariance of the original data from the differentially private data.



(a) Original data (b) Differential privacy (1) (c) Differential privacy (2) (d) Proposed method (1) (e) Proposed method (2)

FIGURE 2. Relationship between split point and the weighted average of the mean squared error (MSE) when generating decision trees.

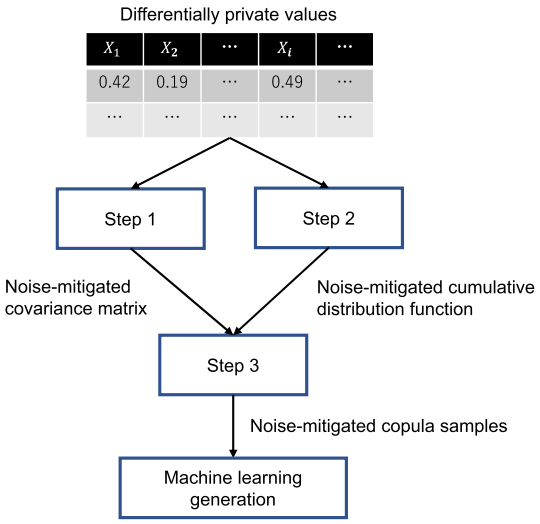


FIGURE 3. Overview of the proposed method.

In the second step, the cumulative distribution function is estimated. There are many methods for estimating cumulative distribution functions from categorical differentially private data (e.g., [8], [32], [71], [72]). In general, to treat continuous values, they first discretize each value into categories [33]. Therefore, some of the continuous value information is lost from the beginning. Our proposed method treats continuous values as is and derives the cumulative distribution function based precisely on the probability distribution of the Laplace distribution owing to differential privacy. Naturally, the random variable of the estimated cumulative distribution function is defined as a discrete random variable. However, the accuracy of the estimated cumulative distribution function is better when treating the data as discrete values from the beginning.

In the final step, copula samples are generated using the results of the first and second steps. To obtain precise samples, our proposed method converts the discrete cumulative distribution function into a continuous cumulative distribution function in the copula space.

B. GENERATION OF NOISE-MITIGATED COVARIANCE MATRIX FROM DIFFERENTIALLY PRIVATE DATA

From the observation of differentially private values, the variance and covariance of the original values need to be calculated. Without loss of generality, the ranges of all attribute values are considered to be $[0, 1]$. In this case, the Laplace noise is drawn from $\mathcal{L}(x; 1/\epsilon)$ if each attribute needs to be protected with ϵ-differential privacy.

Let X_i denote the random variable of the ith attribute of personal data, Z_i denote the random variable with a Laplace distribution $\mathcal{L}(x; 1/\epsilon_i)$, and $\widehat{X}_i$ denote the summation of X_i and Z_i, i.e.,

$$\widehat{X}_i = X_i + Z_i. \tag{7}$$

Let $E[\cdot]$ denote the expected value of a random variable $\cdot$. From the property of the linearity of expectation,

$$E[\widehat{X}_i] = E[X_i + Z_i] = E[X_i] + E[Z_i] = E[X_i] \tag{8}$$

because the mean of Z is zero.

Let $\sigma^2_{X_i}$ represent the variance of X_i. The value of $\sigma^2_{\widehat{X}_i}$ can be calculated by

$$\begin{aligned}\sigma^2_{\widehat{X}_i} &= E[(\widehat{X}_i - E[\widehat{X}_i])^2] = E[(X_i + Z_i - E[X_i])^2] \\ &= E[(X_i - E[X_i])^2] + 2E[X_iZ_i] - 2E[X_i]E[Z_i] + E[Z_i^2] \\ &= \sigma^2_{X_i} + 2E[X_iZ_i] - 2E[X_i]E[Z_i] + E[Z_i^2]. \end{aligned} \tag{9}$$

We have $E[X_iZ_i] = E[Z_i] = 0$ and

$$E[Z_i^2] = \int_{x=-\infty}^{\infty} x^2 \mathcal{L}(x; 1/\epsilon_i)dx = \frac{2}{\epsilon_i^2}. \tag{10}$$

Thus,

$$\sigma^2_{X_i} = \max\left(\sigma^2_{\widehat{X}_i} - \frac{2}{\epsilon_i^2}, 0\right) \tag{11}$$

where we ensure the variance is greater than or equal to zero.

Let σ_{X_i,X_j} represent the covariance of X_i and X_j. The covariance of $\sigma_{\widehat{X}_i,\widehat{X}_j}$ is represented by

$$\begin{aligned}\sigma_{\widehat{X}_i,\widehat{X}_j} &= E[(\widehat{X}_i - E[\widehat{X}_i])(\widehat{X}_j - E[\widehat{X}_j])], \\ &= E[(X_i + Z_i - E[X_i])(X_j + Z_j - E[X_j])] \\ &= E[(X_i - E[X_i])(X_j - E[X_j])] + E[Z_iZ_j] \\ &\quad + E[(X_j - E[X_j])Z_i] + E[(X_i - E[X_i])Z_j]. \end{aligned} \tag{12}$$

The following equation is obtained because Z_i and Z_j are independent of other random variables and $E[Z_i] = E[Z_j] = 0$.

$$\sigma_{X_i,X_j} = \sigma_{\widehat{X}_i,\widehat{X}_j}. \tag{13}$$

Let Σ be a covariance matrix calculated based on Equations 11 and 13. It may be invalid for a normal distribution because the generated covariance matrix may contain some errors; that is, it may not be a positive definite matrix. We use the eigenvalue decomposition technique to create a positive definite matrix.

Let q_i and λ_i be the ith eigenvalues and ith eigenvectors of matrix Σ. Sort q_i for all i in order of magnitude and create a diagonal matrix D that makes them diagonal values. Further, the corresponding λ_i are arranged in the same order to form matrix Λ.

The eigenvalue decomposition of a matrix can be performed as

$$\Sigma = D\Lambda D^{-1}. \tag{14}$$

The fact that a matrix is positive definite is equivalent to the fact that all eigenvalues of the matrix are positive [73]. Hence, we obtain the positive definite matrix version of Σ by replacing the negative values of Λ with small positive values to obtain the matrix Λ' and using the equation

$$\Sigma' = D\Lambda' D^{-1}. \tag{15}$$

C. GENERATION OF NOISE-MITIGATED CUMULATIVE DISTRIBUTION FUNCTION FROM DIFFERENTIALLY PRIVATE DATA

We assume that the model generator already has differentially private data privatized by the Laplace mechanism because it is the most fundamental mechanism. Let $\tilde{v_i}$ represent the privatized value of the true value v_i of person i; that is, $\tilde{v_i}$ is drawn from $\mathcal{L}(x; v_i, s)$, where $s = \Delta/\epsilon$. We use two hyperparameters: b, which represents the number of bins of an input domain for calculating a cumulative distribution function, and r, which determines the output domain. These hyperparameters do not affect privacy; however, they affect the accuracy of machine learning models. The output domain can be $[-\infty, \infty]$ in theory because we assume a Laplace mechanism for realizing differential privacy. However, the accuracy decreases when we set the output domain too wide. If the true value takes a minimum or maximum value, the hyperparameter r specifies the ratio of the time it will fall within that range of the output domain. Let min and max represent the minimum and maximum true values and let min^{pri} and max^{pri} represent the minimum and maximum values of the output domain. The minimum value of an output domain is calculated by solving the equation with regard to min^{pri}.

$$\int_{x=-\infty}^{min_{pri}} \mathcal{L}(x; (max - min)/\epsilon))dx = 1 - r. \tag{16}$$

By solving this equation,

$$min^{pri} = min + \frac{(max - min)\log 2(1 - r)}{\epsilon}. \tag{17}$$

In the same way,

$$max^{pri} = max + \frac{(max - min)\log 2r}{\epsilon}. \tag{18}$$

Let w represent the width of each bin, i.e.,

$$w = \frac{max - min}{b}. \tag{19}$$

Let b^{pri} represent the number of bins in the output domain. This value is calculated as

$$b^{pri} = \frac{max^{pri} - min^{pri}}{w}. \tag{20}$$

Let $\mathscr{L}(x; \mu, s)$ represent the cumulative distribution function of the Laplace distribution with mean μ and scale s. The probability that a true value is categorized in b_i, and it is privatized to another bin b_j^{pri} is calculated by

$$P_{i,j} = \begin{cases} S_{|i-j|+1} & (j \neq 1,\ j \neq b^{pri}) \\ R_0 + S_1 & (i = 1,\ j = 1) \\ R_0 - \sum_{k=2}^{i} P_{i,k} & (i \neq 1,\ j = 1) \\ 1 - \sum_{k=1}^{b^{pri}-1} P_{i,k} & (j = b^{pri}) \end{cases} \tag{21}$$

where for arbitrary m,

$$R_0 = \int_{t=m}^{m+w} \frac{\mathscr{L}(m; t, s)dt}{w} = \frac{s - e^{-w/s}s}{2w} \tag{22}$$

and for arbitrary m and $i \in \{1, \ldots, b^{pri}\}$

$$\begin{aligned} S_i &= \int_{t=m}^{m+i*w} \frac{\mathscr{L}(m; t, s)dt}{w} - \int_{t=m}^{m+(i-1)*w} \frac{\mathscr{L}(m; t, s)dt}{w} \\ &= \begin{cases} \dfrac{e^{-i*w/s}\left(-1 + e^{w/s}\right)^2 s}{2w} & (i \geq 2) \\ 1 + \dfrac{-1 + e^{-w/s}s}{w} & (i = 1.) \end{cases} \end{aligned} \tag{23}$$

From $P_{i,j}$ for all i, j and differentially private data samples, a cumulative distribution function of the true values can be estimated. We can use expectation–maximization based algorithms such as [8].

D. GENERATION OF COPULA SAMPLES FROM NOISE-MITIGATED STATISTICS

A copula model is created from the noise-mitigated covariance matrix Σ (Section V-B) and the noise-mitigated cumulative distribution function F_j $(j = 1, \ldots, g)$ (Section V-C). Then, copula samples can be generated using the copula model based on Section III-F. However, Section III-F assumes that the random variable of a cumulative distribution function is continuous whereas the random variable of the cumulative distribution function obtained in Section V-C is discrete.

Let $F_j(k)$ represent the probability that the random variable of the jth attribute is less than or equal to k, where $k = \{0, \ldots, b-1\}$. The values of $t_{i,j}$ are obtained by Equation 3 for all i and j. Let min_k represent the minimum value of k in $\{0, \ldots, b-1\}$ that satisfies $F_j(k) \geq t_{i,j}$. Then, we calculate

$$\begin{cases} u'_{i,j} = \dfrac{t_{i,j}}{F_j(0) \times b} & (min_k = 0) \\ u'_{i,j} = \dfrac{min_k + t_{i,j} - F_j(min_k - 1)}{(F_j(min_k) - F_j(min_k - 1)) \times b} & (otherwise.) \end{cases} \tag{24}$$

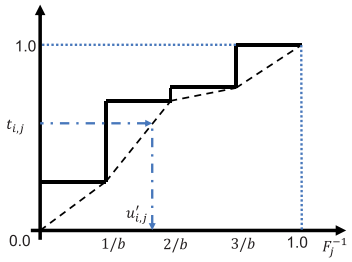


FIGURE 4. An example of calculating $u'_{i,j}$ from $t_{i,j}$ in Equation (24) where $b = 4$.

An example of calculating Equation 24 is illustrated in Figure 4. The figure represents the case where $b = 4$. The resulting $u'_i = \{u'_{i,1}, \ldots, u'_{i,g}\}$ is a generated sample of the proposed method.

The overall procedure of the proposed method is shown in Algorithm 1.

VI. EVALUATION

We evaluated the effectiveness of our proposed method using both artificially created and real datasets. We compared the case where decision trees are created as is with the case where decision trees are created using training data generated by our proposed method because this research targets the decision tree algorithm as a machine learning model. Further, we compared the method of creating a decision tree by generating training data using a normal copula [60]. In addition, we compared our method with the data augmentation method, which is widely used to increase the training data samples.

There are several methods for generating histograms from local differentially private data. Although such methods primarily aim to generate a histogram of attribute values and do not target the prediction of an attribute value from other attribute values, we show the results of state-of-the art methods [32] for the comparison. Gu *et al.* [32] proposed IDUE based on Google's RAPPOR [71] and IDUE based on OUE [72] for generating a histogram of attribute values. The value with the highest frequency among all the histogram's bins that match the attribute values to be predicted can be considered to be the predicted value. Because these methods assume that the data samples are categorized, the input data samples are divided into predefined categories. Here, we consider that the number of categories for the entire attribute is 10^7. Henceforth, IDUE based on RAPPOR is denoted as IDUE(R) and IDUE based on OUE is denoted as IDUE(O).

The privacy budget ϵ was varied over the range 0.01–10 for each attribute. The hyperparameters of each machine learning algorithm are common among the methods being compared. All results are the average of 10 simulations of 5-fold cross validation repeated with the same settings. We used the MSE between the true and predicted values.

The model hyperparameters of the proposed method are b, r, and the target number of samples. We set 100, 0.05, and 100,000, respectively, in the experiments.

Algorithm 1 Overall Procedure of the Proposed Method

Input: Differentially private data $v_{i,j}$ $(i = 1, \ldots, n;\ j = 1, \ldots, g)$, privacy parameter ϵ_i $(i = 1, \ldots, g)$, hyperparameters b, r, and the target number of samples
Output: Machine learning model

for $i = 1, \ldots, g$ **do**
  $Q_i \leftarrow \{v_{j,i} | j = 1, \ldots, n\}$
  $\sigma_{\widehat{X}_i} \leftarrow$ standard deviation of Q_i
  $\sigma^2_{X_i} \leftarrow \max\left(\sigma^2_{\widehat{X}_i} - \frac{2}{\epsilon_i^2}, 0\right)$
  for $j = 1, \ldots, g$ **do**
    $\sigma_{X_i,X_j} \leftarrow$ covariance of Q_i and Q_j
  end for
end for
Generate covariance matrix Σ from $\sigma^2_{X_i}$ and σ_{X_i,X_j} $(i, j = 1, \ldots, g)$
for $i = 1, \ldots, g$ **do**
  $q_i \leftarrow i$th eigenvalue of Σ
  $\lambda_i \leftarrow i$th eigenvector of Σ
end for
Generate matrix D from q_i $(i = 1, \ldots, g)$
Generate matrix Λ from r_i $(i = 1, \ldots, g)$
$\Lambda' \leftarrow \Lambda$ with replacement of negative values with small positive values
$\Sigma' \leftarrow D\Lambda' D^{-1}$
$b^{pri} \leftarrow$ Equation (20) based on Equations (17)-(19), b and r
for $i = 1, \ldots, b^{pri}$ **do**
  for $j = 1, \ldots, b^{pri}$ **do**
    $P_{i,j} \leftarrow$ Equation (21) based on Equation (22)-(23)
  end for
end for
for $j = 1, \ldots, g$ **do**
  $F_j \leftarrow$ estimation results of expectation-maximization using $P_{k,l}$ $(k, l = 1, \ldots, b^{pri})$ and Q_j
end for
$num \leftarrow$ the target number of samples
$\mathbf{S} \leftarrow$ samples generated based on g-dimensional multivariate normal distribution with Σ'
for $i = 1, \ldots, num$ **do**
  for $j = 1, \ldots, g$ **do**
    $t_{i,j} \leftarrow$ Equation (3) using $s_{i,j}$ in $\mathbf{S}$
    $u'_{i,j} \leftarrow$ Equation (24) using F_j
  end for
  $u'_i \leftarrow t'_{i,j} (j = 1, \ldots, g)$
end for
Generate a machine learning model using u'_i $(i = 1, \ldots, num)$

A. EXPERIMENT WITH SYNTHETIC DATASETS

We used three probability distributions to generate synthetic datasets: multivariate normal distribution, multivariate t distribution, and negative multinomial distribution. We used two

parameters for generating the datasets: number of attributes (g) and number of people (n).

For the multivariate normal distribution, all values of the mean vector were set to zero, and the covariance matrix was randomly generated such that it was a symmetric positive definite matrix of real numbers. For the multivariate t-distribution, the scale matrix was randomly generated such that it was a symmetric positive definite matrix of real numbers, and the degrees of freedom parameter was randomly generated such that it was a positive real number. For a negative multinomial distribution, the number of failures until the experiment was stopped was set to the number of samples, and the success probability was randomly generated in $(0, 1/g)$. Each dataset contained g attributes. One attribute was randomly selected and set to the desired output value.

This study focuses on relatively low-dimensional data (e.g., fewer than 30 attributes) based on the fact that many studies on differentially private decision tree generation target personal data with fewer than 30 attributes. For example, Zhao *et al.* used three real datasets and the numbers of attributes were 11, 15, and 19, respectively [18]. The number of attributes of the dataset used in [19] was 20. Wang *et al.* [20] used the real census dataset with 10 attributes and synthetic datasets with 20 attributes. Moreover, many other machine learning models have been proposed that use personal data with fewer than 30 attributes, such as [74], [75]. Of course, there are also many machine learning models that use a larger number of attributes; however, because most research on differentially private decision trees is conducted on datasets with relatively small dimensions, we conducted our experiments on datasets with fewer than 30 attributes.

Fig. 5 shows the simulation results, where n is fixed at 1000 and ϵ is varied from 0.01 to 10 for each attribute. The number of attributes (g) was set to 30. The trend of the results obtained is similar for all probability distributions. The smaller the value of ϵ, the larger the MSE is, and, even in scenarios where the value of ϵ is sufficiently large, the MSE does not go to zero because of the performance limitations of the machine learning model. There is almost no difference in the results between the data augmentation method (Aug.+DT) and the method using the decision tree as is (DT). The method using a copula [60] (Copula+DT) produced similar accuracy. This means that simply applying the copula model to differential privacy data does not lead to improved accuracy.

Our proposed method (Proposal+DT) achieved a higher accuracy (note that a low MSE indicates high accuracy). The estimation accuracy of IDUE(R) and IDUE(O) is relatively low. Note that these methods can construct a histogram of all combinations of attribute values, that is, predicting one attribute value is not the main objective of these methods.

Next, the value of ϵ was fixed at 1.0, and the experiment was conducted by varying n from 1,000 to 10,000. Fig. 6 depicts the results. The accuracy of the proposed method improves as the value of n increases. This is because the larger the value of n is, the better the prediction accuracy of the covariance matrix and the reconstruction accuracy of the cumulative distribution function are. The accuracy of IDUE(R) and IDUE(O) also improves as the value of n increases. In general, methods that generate histograms from differentially private data require a large amount of data. It is expected that the accuracy of these methods will be much better when large datasets are available. In contrast, the accuracy of the other methods did not improve as the value of n increased. The accuracy of the machine learning model is not expected to improve because of the large influence of noise in differential privacy, even if there is a large amount of data with large errors.

To evaluate the variability of the MSE of the proposed method, the results are shown in Fig. 7, where the standard deviation is represented as an error bar. When the size of a dataset is small, the value of the standard deviation is relatively large, but the value of the standard deviation decreases as the size of the dataset increases. Overall, it can also be seen that the standard deviation is not very large compared to the value of MSE. In addition, all of the training accuracies (and their standard deviations) were almost 0.0.

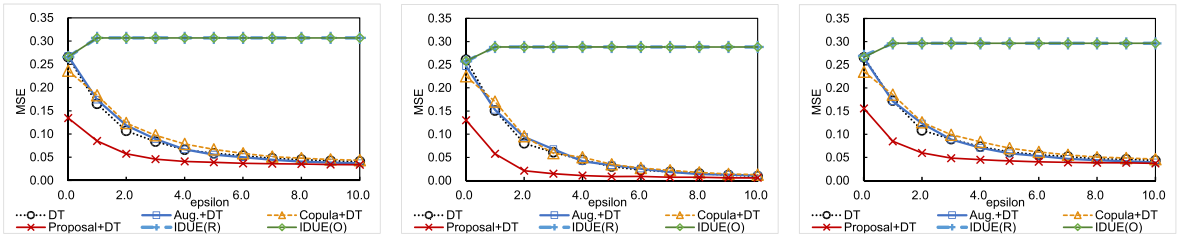
B. EXPERIMENT WITH REAL DATASETS

We used four real datasets for the evaluation. A description of each dataset is provided below.

In the real datasets of Boston, !Kung, Diabetes, and Adult, the number of attributes is 14, 4, 11, and 7, respectively. These datasets are accessible to all. Moreover, our research targets the area of the convergence of privacy and machine learning technologies; therefore, we selected famous datasets for the privacy and machine learning areas, respectively. The most important reason for using the Adult dataset is that it is often used as a benchmark in the field of privacy protection data analysis. The !Kung dataset is also often used to evaluate differential privacy techniques. Boston and Diabetes datasets are famous for machine learning because they are included in the scikit-learn framework, which is the foremost machine learning framework. Each dataset is detailed below.

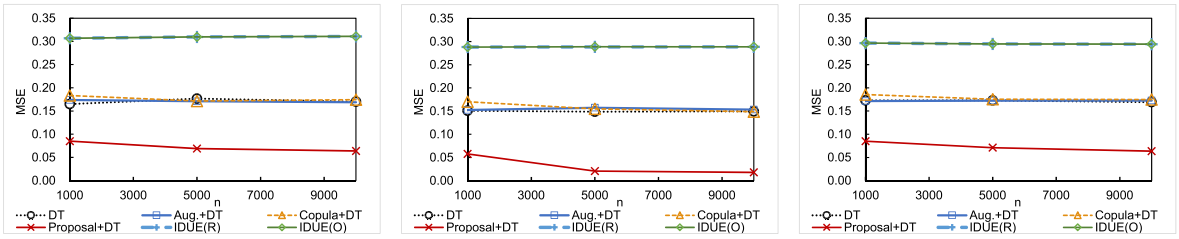
- Boston dataset
  The Boston dataset is considered the baseline dataset for machine learning algorithms [69], [70]. A famous scikit-learn framework[1] contains these data. The Boston dataset comprises data on housing in Boston in the late 1970s. It contains 506 sets of data with attributes such as the crime rate of each city and the percentage of the low-income population. Further, this dataset has been used in many studies on privacy-preserving data mining [76], [77].
- !Kung dataset
  The !Kung dataset [78], [79] is a small census dataset that is widely used for experiments on data mining for differential privacy, such as in [37] and [80]. The !Kung dataset contains 287 records. Following [37], we set

[1] https://scikit-learn.org/



(a) Multivariate normal distribution (b) Multivariate t-distribution (c) Negative multinomial distribution

FIGURE 5. MSE results of synthetic datasets ($g = 30$, $n = 1000$.)



(a) Multivariate normal distribution (b) Multivariate t-distribution (c) Negative multinomial distribution

FIGURE 6. MSE results of synthetic datasets ($g = 30$, $\epsilon = 1.0$.)

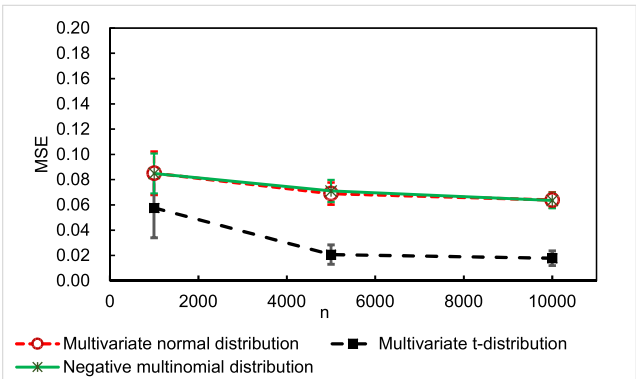


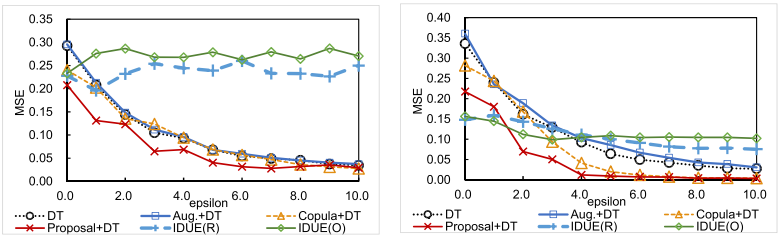
FIGURE 7. MSE results with error bars of the proposed method with changes in the size of dataset.

a task for predicting the height from other attributes (weight, age, and sex).

- Diabetes dataset
  The diabetes dataset [81] is contained in scikit-learn. This dataset was designed to predict the progression of the disease after one year based on the test results of diabetic patients. It contains 442 records with 11 attributes. Many studies have used this dataset to evaluate data mining techniques [82], [83].
- Adult dataset
  The Adult dataset [84] has been used in many studies on privacy-preserving data mining, such as [85], [86]. This dataset is the census data from the USA and has 30,162 records. It contains a flag indicating whether the salary of each person is greater than 50,000 dollars, six numerical attributes such as age, and eight categorical attributes such as race. We used the salary attribute and six numerical attributes.

Fig. 8 shows the experimental results. The accuracy of the proposed method is the best in the experimental results on the real dataset. For the Adult dataset, the accuracy of the proposed method (Proposal+DT), IDUE(R), and IDUE(O) are similar. The adult dataset has more than 30,000 records, which is a relatively large dataset for personal data containing privacy information. Although there is an error of differential privacy, if the value of ϵ is large and sufficient data are collected, IDUE(R) and IDUE(O) can achieve high accuracy as well as the proposed method. However, the proposed method achieved the best accuracy for most settings, especially when ϵ is in [0.01, 8.0] for all datasets.

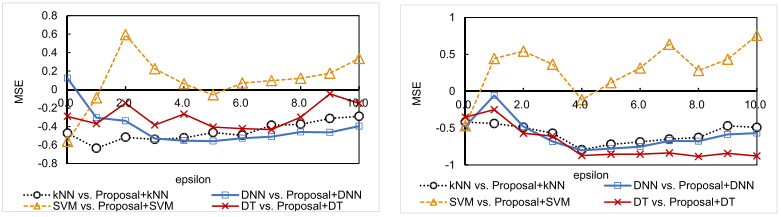
Finally, we conducted experiments on DNN, SVM, and kNN to determine if the proposed method can be applied to other machine learning algorithms besides decision trees. The results are depicted in Fig. 9, which shows the increase ratio of the MSE of each machine learning algorithm. For a decision tree, let α be the MSE of Proposal+DT, and let β be the MSE of DT. In this case, the increase ratio is calculated by $(\alpha - \beta)/\beta$. Therefore, the increase ratio becomes negative if the MSE of Proposal+DT is less than that of DT. Thus, we calculated the increase ratio for the other algorithms as well. For kNN and DT, the proposed method is clearly effective; for DNN, the proposed method can improve the accuracy of the Boston, !Kung, and Diabetes datasets, except for the Adult dataset, which has a large amount of data. For the Adult dataset, the proposed method does not deteriorate the accuracy, and the accuracy is almost the same as that of the DNN. However, the proposed method is not effective for SVM.



(a) Boston (b) !Kung

(c) Diabetes (d) Adult

FIGURE 8. MSE results of real datasets.



(a) Boston (b) !Kung

(c) Diabetes (d) Adult

FIGURE 9. Increased ratio of the MSE of the real datasets of various machine learning models. Negative values indicate that our method decreased the MSEs.

In this experiment, we have measured not only MSE but also mean absolute error (MAE) to further analyze the performance of the proposed method. The results are shown in Fig. 10.

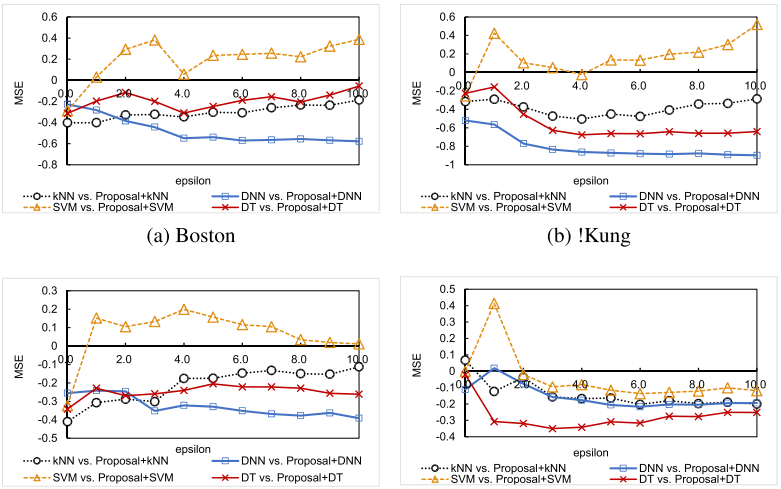
Because the MSE is calculated as the square of the difference between the true value and the predicted value, the MSE will increase significantly if there is a value that is significantly mis-predicted. Therefore, it is suitable for evaluating models that require robustness. On the other hand, because MAE calculates absolute value errors, it measures average ability without considering robustness. The results of Fig. 10 are similar to those of Fig. 9; therefore, for both MSE and MAE indicators, the proposed method is more useful than existing methods for kNN, DT, and DNN.

VII. DISCUSSION

A. ADVANTAGES AND DRAWBACKS

In the previous section, we compared the proposed method with the copula method, histogram generation methods (IDUE(R) and IDUE(O)), and data augmentation. Experimental results show that the proposed method has the highest accuracy. On the other hand, the computation complexity of the proposed method is higher than the copula method because the proposed method uses an expectation–maximization-based algorithm and a copula algorithm. On the contrary, data augmentation has a very small computational cost but also poor accuracy.

If histogram analysis rather than machine learning model generation is the goal, then histogram generation methods



(a) Boston (b) !Kung

(c) Diabetes (d) Adult

FIGURE 10. Increased ratio of the MAE of the real datasets of various machine learning models. Negative values indicate that our method decreased the MAEs.

TABLE 3. Comparison of methods for machine learning model generation from local differentially private data.

Method	Accuracy	Complexity
Proposed method	High	Middle
Copula method	Middle	Low–Middle
Histogram generation such as [32], [72]	Low	Middle
Data augmentation	Low	Low

have very good accuracy. The objective of this study, however, was machine learning model generation, and histogram generation methods did not work well for this purpose. Table 3 summarizes the accuracy of the machine learning models generated and the complexity of the methods.

B. VALUE OF EPSILON

We found that the proposed method is especially effective when ϵ is in the range 0.01–8. Here, we analyze the amount of noise imparted to confirm that it is within a range that can be applied in many practical scenarios. The noise added by differential privacy is generated from $\mathcal{L}(x; \Delta/\epsilon)$. Therefore, the expected absolute value is calculated as

$$E[noise] = \int_{x=-\infty}^{\infty} abs(x)\mathcal{L}(x; \Delta/\epsilon)dx = \Delta/\epsilon. \quad (25)$$

The expected absolute value of the Laplace noise is $1/\epsilon$ when the range of the value of a true personal attribute value is $[0, 1)$. The expected absolute value of the noise is in the range $[0.125, 100]$ when ϵ is in the range $[0.01, 8.0]$; i.e., the amount of noise relative to the range of the possible values of the true value ranges from 12.5% to 10000%.

To determine the influence of ϵ on the effectiveness of the reconstruction copula model from differentially private data, we conducted an additional experiment using the Boston dataset. In Fig. 11, Original represents the correlation value (-0.388) between per capita crime rate by town and an

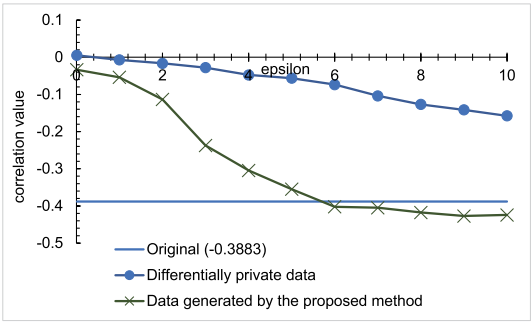


FIGURE 11. Calculated correlation value vs. ϵ.

objective attribute (median value of owner-occupied homes). Fig. 11 also shows the correlation values of differentially private data and data generated by the proposed method. When the value of ϵ is small, the correlation values of differentially private data and data generated by the proposed method approach zero. Because each data sample has a large noise, the information about correlation will be lost. However, for all values of ϵ, it can be seen that the proposed method works well and the correlation values are closer to the true values than the differentially private data.

C. APPLICATION TO CATEGORICAL DATA

We have primarily focused on numerical data in this paper. However, application to categorical data is not difficult; it is, in fact, straightforward. Mutual information is used for the characteristics of each attribute pair. The cumulative distribution function of each pair of attributes was estimated to reduce the impact of perturbation. Then, the mutual information of each pair of attributes was calculated.

If personal data have both numerical and categorical values, privatized numerical values are digitized into several categories when generating a copula model. Specifically,

assume that the first h attributes are numerical and the next $(g - h)$ attributes are categorical without loss of generality, i.e., $Q_1, \ldots, Q_h$ are numerical attributes and $Q_{h+1}, \ldots, Q_g$ are categorical attributes. When generating a cumulative distribution function for each attribute, the method described in Section V-C is used for Q_i $(i \leq h)$, and the method described above is used for Q_i $(i > h)$. In our proposed method, covariance is necessary for each pair of numerical attributes, and mutual information is necessary for each pair of categorical attributes. Assume that the pair of attributes are Q_i and Q_j. The covariance is calculated if $i \leq h$ and $j \leq h$. The mutual information is calculated if $i > h$ and $j > h$. Special processing is required if $i \leq h$ and $j > h$ or if $i > h$ and $l \leq h$. In this case, the privatized numerical values are digitized into several categories, and the mutual information of the two attributes is then calculated.

As has been mentioned, this paper is concerned with regression tasks. When applying our method to the classification task, it is a requirement to deal with class imbalance.

D. COMPARISON BETWEEN SEVERAL MACHINE LEARNING ALGORITHMS

The reason why the proposed method works well is as described in Section IV for decision trees. Because differentially private data have a large amount of noise, the accuracy of machine learning models trained on such data decreases. However, several machine learning algorithms are robust to such noise data.

In DNN, parameters are updated using stochastic gradient descent or its variants. If too much noise is added to this process, it will often be trained in the wrong direction. However, by increasing the batch size, the robustness to noisy data is increased. This is because, within a single batch, gradient updates from randomly sampled noisy data are nearly canceled out [87]. Nevertheless, there is a limit to the ability to cancel out noise. The experimental results show that the accuracy of DNN is better when using the proposed method.

The SVM for regression is also called support vector regression (SVR). SVR employs an e-insensitive loss function that penalizes predictions that are farther from the desired output than e. The e-insensitive region is less sensitive to noisy inputs and thus increases the robustness of the model [88]. This property of SVM may have worked well for noisy, differentially private data. More detailed validation for SVM is a future issue.

On the other hand, KNN is known to be very sensitive to noisy data [89]. Therefore, the proposed method works well also for KNN, as shown by the experimental results.

E. TREATING HIGH-DIMENSIONAL DATA

Because a copula model is suitable for low-dimensional data, handling high-dimensional data as it is with our method is difficult. To treat high-dimensional data, techniques of dimension reduction, such as principal component analysis (PCA), can be used. Several studies have shown that reducing the dimensions improved machine learning models' accuracy [90], [91]. To perform PCA with differentially private data, the algorithm Wang and Xu proposed [92] can be used.

For DNN, many models use high-dimensional data. However, several studies have generated highly accurate DNN models, using PCA or other dimension reduction techniques, such as [93]. This study is concerned with data with relatively few attributes. Therefore, for high-dimensional data, it has not been verified that the proposed method works effectively without dimensionality reduction. Verification of how the proposed method works with and without dimensionality reduction is a future issue.

One reason to focus on decision trees in this paper is high human interpretability. On the other hand, in many studies, resarchers have aimed to interpret DNNs' behavior. For example, Nascita *et al.* proposed an algorithm that provides global interpretation for DNNs [94]. Interpretation of model behavior when DNN models are constructed using our proposed method is also an issue to be addressed in the future.

F. PREPROCESSING TECHNIQUES

General preprocessing techniques include data cleaning, dimension reduction, and so on [95]. They do not consider differentially private numerical data, which are very noisy but for which the probability distribution of the noise is the Laplace distribution. Our proposed method generates a copula-based synthetic dataset that reduces the noise due to differential privacy. Therefore, the techniques (e.g., data cleaning and dimension reduction) could be applied to the copula-based synthetic dataset generated by the proposed method. Data augmentation is another preprocessing technique used for increasing training data. In addition, this technique does not consider differentially private data; therefore, it makes little contribution to improving the accuracy of machine learning. In the experiment section, we showed that our method outperforms other techniques, including a data augmentation technique.

VIII. CONCLUSION

Personal data with noise caused by differential privacy is widely collected to protect privacy. In this paper, we proposed a method for generating highly accurate machine learning models, especially decision tree models, based on datasets with differential privacy noise. Experimental results show that the proposed method improves the accuracy of machine learning models, not only for the decision tree algorithm but also for kNN and DNN with relatively few attributes, for a range of practical ϵ values compared with the conventional copula method and state-of-the-art IDUE(R) and IDUE(O).

In future work, we plan to extend the proposed method to other types of datasets where differential privacy is applicable, such as time-series data, image data, and data with graph structures.

REFERENCES

[1] D. Mukherjee, M. Yurochkin, M. Banerjee, and Y. Sun, "Two simple ways to learn individual fairness metrics from data," in *Proc. ICML*, vol. 2020, pp. 7097–7107.

[2] I. Žliobaitė, "Measuring discrimination in algorithmic decision making," *Data Mining Knowl. Discovery*, vol. 31, no. 4, pp. 1060–1089, Jul. 2017.

[3] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. ACM CCS*, Sep. 2015, pp. 1310–1321.

[4] E. Toreini, M. Aitken, K. Coopamootoo, K. Elliott, C. G. Zelaya, and A. van Moorsel, "The relationship between trust in AI and trustworthy machine learning technologies," in *Proc. ACM FAccT*, Jan. 2020, pp. 272–283.

[5] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE FOCS*, Oct. 2013, pp. 429–438.

[6] T. Murakami and K. Takahashi, "Toward evaluating re-identification risks in the local privacy model," 2020, *arXiv:2010.08238*.

[7] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 746–789, 1st Quart., 2020.

[8] T. Murakami and Y. Kawamoto, "Utility-optimized local differential privacy mechanisms for distribution estimation," in *Proc. USENIX Secur. Symp.*, 2019, pp. 1877–1894.

[9] T. Zhu, D. Ye, W. Wang, W. Zhou, and P. Yu, "More than privacy: Applying differential privacy in key areas of artificial intelligence," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2824–2843, Jun. 2022.

[10] S. H. Yoo, H. Geng, T. L. Chiu, S. K. Yu, D. C. Cho, J. Heo, M. S. Choi, I. H. Choi, C. C. Van, N. V. Nhung, B. J. Min, and H. Lee, "Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging," *Frontiers Med.*, vol. 7, pp. 1–8, Jul. 2020.

[11] E. A. Toraih, R. M. Elshazli, M. H. Hussein, A. Elgaml, M. Amin, M. El-Mowafy, M. El-Mesery, A. Ellythy, J. Duchesne, M. T. Killackey, K. C. Ferdinand, E. Kandil, and M. S. Fawzy, "Association of cardiac biomarkers and comorbidities with increased mortality, severity, and cardiac injury in COVID-19 patients: A meta-regression and decision tree analysis," *J. Med. Virol.*, vol. 92, no. 11, pp. 2473–2488, Nov. 2020.

[12] J. Jiang, X. Zhu, G. Han, M. Guizani, and L. Shu, "A dynamic trust evaluation and update mechanism based on C4.5 decision tree in underwater wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9031–9040, Aug. 2020.

[13] Q. Hou, N. Zhang, D. S. Kirschen, E. Du, Y. Cheng, and C. Kang, "Sparse oblique decision tree for power system security rules extraction and embedding," *IEEE Trans. Power Syst.*, vol. 36, no. 2, pp. 1605–1615, Mar. 2021.

[14] H. Lu and X. Ma, "Hybrid decision tree-based machine learning models for short-term water quality prediction," *Chemosphere*, vol. 249, Jun. 2020, Art. no. 126169.

[15] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens, "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models," *Decis. Support Syst.*, vol. 51, no. 1, pp. 141–154, Apr. 2011.

[16] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[17] S. Fletcher and M. Z. Islam, "Decision tree classification with differential privacy: A survey," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 83:1–83:33, 2019.

[18] L. Zhao, L. Ni, S. Hu, Y. Chen, P. Zhou, F. Xiao, and L. Wu, "InPrivate digging: Enabling tree-based distributed data mining with differential privacy," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 2087–2095.

[19] Z. Sun, Y. Wang, M. Shu, R. Liu, and H. Zhao, "Differential privacy for data and model publishing of medical data," *IEEE Access*, vol. 7, pp. 152103–152114, 2019.

[20] S. Wang and J. M. Chang, "Privacy-preserving boosting in the local setting," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4451–4465, 2021.

[21] A. J. Patton, "A review of copula models for economic time series," *J. Multivariate Anal.*, vol. 110, pp. 4–18, Sep. 2012.

[22] N. A. Khan, M. A. Habib, and S. Jamal, "Effects of smartphone application usage on mobility choices," *Transp. Res. A, Policy Pract.*, vol. 132, pp. 932–947, Feb. 2020.

[23] P. Krieter and A. Breiter, "Analyzing mobile application usage: Generating log files from mobile screen recordings," in *Proc. MobileHCI*, Sep. 2018, pp. 1–10.

[24] Ü. G. Peköz, "Product usage data collection and challenges of data anonymization," in *Data-Centric Business and Applications* (Lecture Notes on Data Engineering and Communications Technologies). Cham, Switzerland: Springer, 2018, pp. 117–136.

[25] E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ML test score: A rubric for ML production readiness and technical debt reduction," in *Proc. IEEE BigData*, Dec. 2017, pp. 1123–1132.

[26] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, and J. Wernsing, "Machine teaching: A new paradigm for building machine learning systems," 2017, *arXiv:1707.06742*.

[27] K. Holstein, J. W. Vaughan, H. Daumé, M. Dudik, and H. Wallach, "Improving fairness in machine learning systems: What do industry practitioners need?" in *Proc. ACM CHI*, May 2019, pp. 600:1–600:16.

[28] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani, "Aequitas: A bias and fairness audit toolkit," 2018, *arXiv:1811.05577*.

[29] F. Tramer, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin, "FairTest: Discovering unwarranted associations in data-driven applications," in *Proc. IEEE European S&P*, Apr. 2017, pp. 401–416.

[30] Y. Liang, S. Li, C. Yan, M. Li, and C. Jiang, "Explaining the black-box model: A survey of local interpretation methods for deep neural networks," *Neurocomputing*, vol. 419, pp. 168–182, Jan. 2021.

[31] T. Wang, X. Zhang, J. Feng, and X. Yang, "A comprehensive survey on local differential privacy toward data statistics and analysis," *Sensors*, vol. 20, no. 24, pp. 1–48, 2020.

[32] X. Gu, M. Li, L. Xiong, and Y. Cao, "Providing input-discriminative protection for local differential privacy," in *Proc. ICDE*, Apr. 2020, pp. 505–516.

[33] T. Murakami, H. Hino, and J. Sakuma, "Toward distribution estimation under local differential privacy with small samples," *Proc. Privacy Enhancing Technol.*, vol. 2018, no. 3, pp. 84–104, Jun. 2018.

[34] Y. Sei and A. Ohsuga, "Differentially private mobile crowd sensing considering sensing errors," *Sensors*, vol. 20, no. 10, pp. 2785:1–2785:25, May 2020.

[35] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, no. 29, pp. 1069–1109, Mar. 2011.

[36] X. Fang, F. Yu, G. Yang, and Y. Qu, "Regression analysis with differential privacy preserving," *IEEE Access*, vol. 7, pp. 129353–129361, 2019.

[37] M. T. Smith, M. A. Álvarez, M. Zwiessele, and N. D. Lawrence, "Differentially private regression with Gaussian processes," in *Proc. AIStat*, 2018, pp. 1195–1203.

[38] A. F. Barrientos, J. P. Reiter, A. Machanavajjhala, and Y. Chen, "Differentially private significance tests for regression coefficients," *J. Comput. Graph. Statist.*, vol. 28, no. 2, pp. 440–453, Apr. 2019.

[39] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM CCS*, 2016, pp. 308–318.

[40] Y. Lin, L.-Y. Bao, Z.-M. Li, S.-Z. Si, and C.-H. Chu, "Differential privacy protection over deep learning: An investigation of its impacted factors," *Comput. Secur.*, vol. 99, Dec. 2020, Art. no. 102061.

[41] Z. Bu, J. Dong, Q. Long, and S. Weijie, "Deep learning with Gaussian differential privacy," *Harvard Data Sci. Rev.*, vol. 2, no. 3, pp. 1–31, Jul. 2020.

[42] Y. Sei, H. Okumura, and A. Ohsuga, "Privacy-preserving publication of deep neural networks," in *Proc. IEEE DSS*, Dec. 2016, pp. 1418–1425.

[43] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[44] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020.

[45] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated Learning—Challenges, methods, and future directions," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2020.
[46] M. Song, Z. Wang, Z. Zhang, Y. Song, Q. Wang, J. Ren, and H. Qi, "Analyzing user-level privacy attack against federated learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2430–2444, Oct. 2020.
[47] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthcare Informat. Res.*, vol. 5, no. 1, pp. 1–19, Mar. 2021.
[48] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, Q. S. T. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
[49] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "LDP-fed: Federated learning with local differential privacy," in *Proc. ACM EdgeSys*, Apr. 2020, pp. 61–66.
[50] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *Proc. IEEE SP*, May 2020, pp. 304–317.
[51] Z. Chuanxin, S. Yi, and W. Degang, "Federated learning with Gaussian differential privacy," in *Proc. RICAI*, Oct. 2020, pp. 296–301.
[52] D. Wang and J. Xu, "On sparse linear regression in the local differential privacy model," in *Proc. ICML*, 2019, pp. 6628–6637.
[53] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5827–5842, Jul. 2020.
[54] U. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms*, 2019, pp. 2468–2479.
[55] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *Proc. Annu. Int. Cryptol. Conf.*, in Lecture Notes in Computer Science, vol. 11693, 2019, pp. 638–667.
[56] Z. Zhang, T. Wang, N. Li, J. Honorio, M. Backes, S. He, J. Chen, and Y. Zhang, "PrivSyn: Differentially private data synthesis," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 1–18.
[57] G. Vietri, G. Tian, M. Bun, T. Steinke, and Z. S. Wu, "New oracle-efficient algorithms for private synthetic data release," in *Proc. ICML*, 2020, pp. 9707–9716.
[58] F. Harder, K. Adamczewski, and M. Park, "DP-MERF: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation," in *Proc. AISTAT*, vol. 130, 2021, pp. 1819–1827.
[59] K. Cai, X. Lei, J. Wei, and X. Xiao, "Data synthesis via differentially private Markov random fields," *Proc. VLDB Endowment*, vol. 14, no. 11, pp. 2190–2202, Jul. 2021.
[60] L. Rocher, J. M. Hendrickx, and Y.-A. de Montjoye, "Estimating the success of re-identifications in incomplete datasets using generative models," *Nature Commun.*, vol. 10, no. 1, pp. 1–9, Dec. 2019.
[61] Y. Chen, "A copula-based supervised learning classification for continuous and discrete data," *J. Data Sci.*, vol. 14, no. 4, pp. 769–790, Mar. 2021.
[62] A. R. Gonçalves, F. J. V. Zuben, A. Banerjee, U. Dogan, M. Kloft, F. Orabona, and T. Tommasi, "Multi-task sparse structure learning with Gaussian copula models," *J. Mach. Learn. Res.*, vol. 17, no. 33, pp. 1–30, 2016.
[63] J. A. Carrillo, M. Nieto, J. F. Velez, and D. Velez, "A new machine learning forecasting algorithm based on bivariate copula functions," *Forecasting*, vol. 3, no. 2, pp. 355–376, May 2021.
[64] Y. Sei, J. Andrew, H. Okumura, and A. Ohsuga, "Privacy-preserving collaborative data collection and analysis with many missing values," *IEEE Trans. Depend. Sec. Comput.*, early access, May 13, 2022, doi: 10.1109/TDSC.2022.3174887.
[65] Y. Sei, H. Okumura, and A. Ohsuga, "Re-identification in differentially private incomplete datasets," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 62–72, 2022.
[66] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *classification and Regression Trees*. Evanston, IL, USA: Routledge, Oct. 1984.
[67] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.
[68] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[69] J. M. Chen, "An introduction to machine learning for panel data," *Int. Adv. Econ. Res.*, vol. 27, no. 1, pp. 1–16, Feb. 2021.
[70] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[71] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM CCS*, Nov. 2014, pp. 1054–1067.
[72] T. Wang, J. Blocki, N. Li, T. Wang, J. Blocki, and N. Li, "Locally differentially private protocols for frequency estimation," in *Proc. USENIX Secur. Symp.*, 2017, pp. 729–745.
[73] H. Zhang and F. Ding, "A property of the eigenvalues of the symmetric positive definite matrix and the iterative algorithm for coupled Sylvester matrix equations," *J. Franklin Inst.*, vol. 351, no. 1, pp. 340–357, Jan. 2014.
[74] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms," *NPJ Digit. Med.*, vol. 4, no. 1, pp. 1–5, Dec. 2021.
[75] A. L. Booth, E. Abels, and P. McCaffrey, "Development of a prognostic model for mortality in COVID-19 infection using machine learning," *Mod. Pathol.*, vol. 34, no. 3, pp. 522–531, Mar. 2021.
[76] K. Mandal and G. Gong, "PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proc. ACM CCS*, 2019, pp. 57–68.
[77] P. Vepakomma, A. Singh, O. Gupta, and R. Raskar, "NoPeek: Information leakage reduction to share activations in distributed deep learning," in *Proc. IEEE ICDMW*, Nov. 2020, pp. 933–942.
[78] N. G. B. Jones, L. C. Smith, J. F. O'Connell, K. Hawkes, and C. L. Kamuzora, "Demography of the Hadza, an increasing and high density population of savanna foragers," *Amer. J. Phys. Anthropol.*, vol. 89, no. 2, pp. 159–181, Oct. 1992.
[79] N. Howell, *Demography of the Dobe !Kung*, 2nd ed. Evanston, IL, USA: Routledge, 2017.
[80] A. Dandekar, D. Basu, and S. Bressan, "Differentially private non-parametric machine learning as a service," in *Proc. DEXA*, 2019, pp. 189–204.
[81] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, Apr. 2004.
[82] L. Zhao, "Privacy-preserving distributed analytics in fog-enabled IoT systems," *Sensors*, vol. 20, no. 21, pp. 1–23, 2020.
[83] H. Choi, Y. Kim, and S. Kwon, "Sparse bridge estimation with a diverging number of parameters," *Statist. Interface*, vol. 6, no. 2, pp. 231–242, 2013.
[84] D. Dua and C. Graff. (2019). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[85] X. Xiao, Y. Tao, and M. Chen, "Optimal random perturbation at multiple privacy levels," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 814–825, 2009.
[86] G. Yang, X. Ye, X. Fang, R. Wu, and L. Wang, "Associated attribute-aware differentially private data publishing via microaggregation," *IEEE Access*, vol. 8, pp. 79158–79168, 2020.
[87] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," May 2017, *arXiv:1705.10694*.
[88] M. Awad and R. Khanna, "Support vector regression," in *Efficient Learning Machines*. Berkeley, CA, USA: Apress, 2015, pp. 67–80.
[89] J. A. Sáez, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognit.*, vol. 46, no. 1, pp. 355–364, 2013.
[90] A. Suleiman, M. R. Tight, and A. D. Quinn, "Hybrid neural networks and boosted regression tree models for predicting roadside particulate matter," *Environ. Model. Assessment*, vol. 21, no. 6, pp. 731–750, Dec. 2016.
[91] A. Caggiano, R. Angelone, F. Napolitano, L. Nele, and R. Teti, "Dimensionality reduction of sensorial features by principal component analysis for ANN machine learning in tool condition monitoring of CFRP drilling," *Proc. CIRP*, vol. 78, pp. 307–312, Jan. 2018.
[92] D. Wang and J. Xu, "Principal component analysis in the local differential privacy model," *Theor. Comput. Sci.*, vol. 809, pp. 296–312, Feb. 2020.
[93] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *J. Ambient Intell. Hum. Comput.*, vol. 1, pp. 1–14, Apr. 2020.
[94] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 4, pp. 4225–4246, Dec. 2021.
[95] H. Benhar, A. Idri, and J. L. Fernández-Alemán, "Data preprocessing for heart disease classification: A systematic literature review," *Comput. Methods Programs Biomed.*, vol. 195, pp. 1–30, Oct. 2020.

YUICHI SEI (Member, IEEE) received the Ph.D. degree in information science and technology from The University of Tokyo, in 2009. From 2009 to 2012, he was with the Mitsubishi Research Institute. He joined The University of Electro-Communications, in 2013, where he is currently an Associate Professor with the Graduate School of Informatics and Engineering. He is also a Visiting Researcher at the Mitsubishi Research Institute and an Adjunct Researcher at Waseda University. His current research interests include pervasive computing, privacy-preserving data mining, and software engineering. He is a member of the IEEE Computer Society (IEEE CS), the IEEE Signal Processing Society (IEEE SP), the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Japan Society for Software Science and Technology (JSSST). He was a recipient of the IPSJ Best Paper Award and the JSCE Hydraulic Engineering Best Paper Award, in 2017.

J. ANDREW ONESIMU received the Bachelor of Engineering (B.E.) degree in CSE, in 2011, the Master of Engineering (M.E.) degree from Anna University, Chennai, India, in 2013, and the Ph.D. degree from the Vellore Institute of Technology (VIT), Vellore, India, in 2021. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering (CSE), Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. He is an active researcher published scientific research articles in reputed journals and conferences. He also served as a speaker for prestigious conferences worldwide. He is having more than eight years of teaching experience at undergraduate (U.G.) and postgraduate (P.G.) levels. His research interests include privacy preserving data, healthcare data analysis, deep learning, machine learning, computer vision, and blockchain technologies.

AKIHIKO OHSUGA (Member, IEEE) received the Ph.D. degree in computer science from Waseda University, in 1995. From 1981 to 2007, he was with Toshiba Corporation. He joined The University of Electro-Communications, in 2007. He is currently a Professor with the Graduate School of Informatics and Engineering. He is also a Visiting Professor at the National Institute of Informatics. His research interests include agent technologies, web intelligence, and software engineering. He is a member of the IEEE Computer Society (IEEE CS), the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), the Japanese Society for Artificial Intelligence (JSAI), the Japan Society for Software Science and Technology (JSSST), and the Institute of Electrical Engineers of Japan (IEEJ). He has been a fellow of IPSJ, since 2017. He served as a member for the JSAI Board of Directors, a member for the JSSST Board of Directors, and a member for the JSSST Councilor. He received the IPSJ Best Paper Awards, in 1987 and 2017. He served as the Chair for the IEEE CS Japan Chapter.