IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 6, 2019, accepted September 30, 2019, date of publication October 14, 2019, date of current version October 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947295

Differential Privacy for Data and Model Publishing of Medical Data

ZONGKUN SUN[1], YINGLONG WANG[2], MINGLEI SHU[2], RUIXIA LIU[2], AND HUIQI ZHAO[1]

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China

Corresponding author: Yinglong Wang (wangylscsc@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61603224, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2017MF029.

ABSTRACT Combining medical data and machine learning has fully utilized the value of medical data. However, medical data contain a large amount of sensitive information, and the inappropriate handling of data can lead to the leakage of personal privacy. Thus, both publishing data and training data in machine learning may reveal the privacy of patients. To address the above issue, we propose two effective approaches. One combines a differential privacy and decision tree (DPDT) approach to provide strong privacy guarantees for publishing data, which establishes a weight calculation system based on the classification and regression tree (CART) method and takes weights as a new element of differential privacy to participate in privacy protection and reduce the negative impact of differential privacy on data availability. Another uses the differentially private mini-batch gradient descent algorithm (DPMB) to provide strong protection for training data; it tracks the privacy loss and allows the model to satisfy differential privacy in the process of gradient descent to prevent attackers from invading personal privacy with the training data. It is worth mentioning that, in this paper, we adopt the data processed by DPDT as the training data of DPMB to further strengthen the privacy of data.

INDEX TERMS Deep learning, data privacy, differential privacy, data publishing.

I. INTRODUCTION

Recent progress in deep learning has led to impressive successes in a wide range of applications, such as the combination of deep learning with medical data. This approach enables the machine to learn some basic diagnoses, which not only can help patients better understand their physical condition with less frequent medical visits but can also help doctors to reduce work pressure and improve work efficiency [2]. However, with the rapid development of machine learning and deep learning, the acquisition of medical data has become a major problem. Because medical data contain a large amount of sensitive information, it may reveal the privacy of patients both when publishing data and training data in machine learning [3], [4]. Therefore, hospitals are reluctant to provide data sources. In addition, if privacy issues cannot be resolved, no technology can be used publicly, no matter how developed it is.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

An increasing number of people are paying attention to protecting data privacy while applying data. On the one hand, for publishing data, K-anonymity [5], L-diversity [6] and T-closeness [7] protect sensitive information against attacks, such as linking attacks, skewness attacks and background knowledge attacks. However, they do not have high resistance to the background knowledge attacks because they lack a strong attack model. However, differential privacy has a better ability to resist all of the above attacks with good privacy protection and has been widely used by scholars.

On the other hand, for a published model, recent attacks have used information hidden in the model to recover some sensitive training data. Such attacks can proceed directly by analyzing internal model parameters and indirectly by repeatedly querying models to gather data for the attack analysis, such as the model inversion (MI) attack [8], membership inference attack [9], etc. These attacks can be effectively resisted by applying a differential privacy algorithm to machine learning [10]–[12].

Although differential privacy provides strong privacy guarantees for publishing data and training data, it will greatly reduce the availability of data and models, which is embodied specifically in the fact that using the data processed by differential privacy as training data will decrease model accuracy, and applying a differential privacy algorithm to machine learning will also reduce accuracy sharply.

In this paper, we are committed to finding a balance between usability and privacy and to preprocessing data to solve the problems faced by most discrete medical data. Specific contributions are as follows:

1) By normalizing data, we make them more closely distributed and reduce their variance, and thereby data quality can be improved. In addition, differential privacy is positively affected by normalization through step simplification and sensitivity reduction, which decreases the negative impact of differential privacy on data availability and facilitates the application of differential privacy to data.
2) The combination of differential privacy and decision tree (DPDT) is proposed for publishing data to balance the protection strength and data availability. Different from the traditional method of adding noise, we formulate rules to calculate attribute weights via the decision tree. Then, we use weights to influence the degree of noise added to attributes. Thereupon, data availability and privacy can obtain a better balance, which further reduces the negative impact of differential privacy on data availability.
3) The differentially private mini-batch gradient descent algorithm (DPMB) is established to prevent attackers from invading personal privacy via training data, thereby providing strong privacy guarantees for training data of the publishing model. Moreover, the moments accountant is adopted, which is advanced and can obtain much tighter estimates on the overall privacy loss than the traditional strong composition theorem. We record the training accuracy for each privacy parameter, thus obtaining direct guidance for the selection of appropriate privacy parameters. In addition, to improve the fitting speed and the quality of the model, we let the learning rate decrease with the increase of epoch.

The rest of the paper is organized as follows. The next section reviews the background of differential privacy, the moments accountant, deep learning, and classification and regression tree (CART). III explains our design methodology. IV describes our experimental model and experimental results. V discusses related work, and the conclusion will be presented in VI.

II. BACKGROUND

This section introduces differential privacy and the moments accountant and briefly presents the definition of CART and an overview of deep learning.

A. DIFFERENTIAL PRIVACY

Differential privacy [10], [13], [14] possesses a strong capacity in privacy protection. In our experiments, we apply the original definition of ε-differential privacy [15] with a Laplace mechanism to data for dataset privacy protection and adopt the variant definition of (ε, δ)-differential privacy [16] with a Gaussian mechanism, which adds δ to indicate the possibility that ε-differential privacy might be broken, for training data protection in the model.

Original definition 1: Two input datasets d and d' whose difference is at most one record are called adjacent databases. For a randomized function $\mathcal{K} : \mathcal{D} \to \mathcal{R}$ with two adjacent databases as input, if you want it to satisfy ε-differential privacy, any subcollection of outputs $S \in \mathcal{R}$ needs to satisfy:

$$Pr[\mathcal{K}(d) \in S] \leq e^{\varepsilon} Pr[\mathcal{K}(d') \in S]. \tag{1}$$

The random function $\mathcal{K}$ protects privacy by adding noise to a real-valued function $f : \mathcal{D} \to \mathbb{R}$. When we adopt Laplace noise, the formula is as follows:

$$\mathcal{K}(d) \triangleq f(d) + Lap(0, \frac{\Delta f}{\varepsilon}). \tag{2}$$

In the above formulas, $Lap(0, \frac{\Delta f}{\varepsilon})$ is the Laplace distribution whose standard deviation is $\sqrt{2}\Delta f/\varepsilon$ and mean is 0, and Δf is f's *sensitivity* calculated from the formula $\max |f(d) - f(d')|$.

Variant definition 1: Two input datasets d and d' whose difference is at most one record are called adjacent databases. For a randomized function $\mathcal{K} : \mathcal{D} \to \mathcal{R}$ with two adjacent databases as input, if you want it to satisfy (ε, δ)-differential privacy, any subcollection of outputs $S \in \mathcal{R}$ needs to satisfy:

$$Pr[\mathcal{K}(d) \in S] \leq e^{\varepsilon} Pr[\mathcal{K}(d') \in S] + \delta. \tag{3}$$

When we adopt Gaussian noise, the formula is as follows:

$$\mathcal{K}(d) \triangleq f(d) + \mathcal{N}(0, \Delta f^2 \cdot \sigma^2). \tag{4}$$

In the above formulas, $\mathcal{N}$ represents the normal distribution.

Differential privacy has many useful properties, such as composability, group privacy and robustness. In this paper, we mainly use the basic composition theorem [16], [17] and advanced composition theorems [18]–[21] to repeatedly apply additive-noise mechanisms. To keep track of privacy loss, McSherry put forward the *privacy accountant* [22] via a composite mechanism, and Marín Abadi proposed the *moments accountant* [23] which can provide a tighter bound on the privacy loss.

B. THE MOMENTS ACCOUNTANT

Abadi et al. [23] recommended the moments accountant based on former foundations [20], [21], [24]. It keeps track of a bound on the moments of the privacy loss random variable, which can provide a tighter bound compared to the strong composition theorem. The value of the privacy loss is associated with the level of noise added to the algorithm. It defines a bound for the privacy loss of function $\mathcal{K}$ to satisfy differential privacy, which is called the tail bound. Moreover,

it computes the log moments of the privacy loss and then gets the tail bound via the moments bound and the standard Markovin equality.

Definition 1: Let $\mathbf{a}$ represent an auxiliary input and c represent the privacy loss. The privacy loss at the output o is calculated by

$$c(o; \mathcal{K}, \mathbf{a}, d, d') \triangleq \log \frac{\mathcal{K}(\mathbf{a}, d) = o}{\mathcal{K}(\mathbf{a}, d') = o}. \tag{5}$$

The following is the definition of the moments accountant:

Definition 2: Let $\mathbf{a}$ represent an auxiliary input. The moments accountant is calculated by

$$\alpha_{\mathcal{K}}(\lambda) \triangleq \max_{\mathbf{a}, d, d'} \alpha_{\mathcal{K}}(\lambda; \mathbf{a}, d, d'). \tag{6}$$

In the above formula, $\alpha_{\mathcal{K}}(\lambda; \mathbf{a}, d, d')$ represents the log moment at the value of λ, and it can be calculated by $\alpha_{\mathcal{K}}(\lambda; \mathbf{a}, d, d') \triangleq \log \mathbb{E}[\exp(\lambda c(\mathcal{K}(d); \mathcal{K}, \mathbf{a}, d, d'))]$.

The properties of the moments accountant are as follows:

Theorem 1: 1.[**Composability**] Assume that a function $\mathcal{K}$ is composed of a series of adaptive functions $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_k$ where $\mathcal{K}_i : \prod_{j=1}^{i-1} \mathcal{R} \times \mathcal{D} \to \mathcal{R}_i$. Therefore, for any λ

$$\alpha_{\mathcal{K}}(\lambda) \leq \sum_{i=1}^{k} \alpha_{\mathcal{K}_i}(\lambda). \tag{7}$$

2.[**Tail bound**] For any $\varepsilon > 0$, the function $\mathcal{K}$ is (ε, δ)-differentially private for

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{K}}(\lambda) - \lambda\varepsilon). \tag{8}$$

The moments accountant can obtain much tighter estimates on the overall privacy loss and clearly show the relationship between model accuracy and privacy intensity. Though this concept is very new, its effectiveness and potential cannot be denied. The details can be found in the full version of the paper [23], [25].

C. CLASSIFICATION AND REGRESSION TREE

CART [26] uses the Gini index of each attribute to determine partitioning properties. The purity of dataset (D) can be measured by the Gini value:

$$\begin{aligned} Gini(D) &= \Sigma_{k=1}^{|y|} \Sigma_{k' \neq k} p_k p_{k'} \\ &= 1 - \Sigma_{k=1}^{|y|} p_k^2. \end{aligned} \tag{9}$$

Intuitively, $Gini(D)$ represents the probability of randomly extracting two different sample types from D. Therefore, the smaller the value of $Gini(D)$ is, the higher the purity of D is. The Gini index of attribute a is defined as:

$$Gini\_index(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v). \tag{10}$$

In the candidate attribute set A, we choose the property with the smallest Gini index as the optimal partition attribute:

$$a_* = \arg\min_{a \in A} Gini\_index(D, a). \tag{11}$$

That is, in this node of CART, according to this attribute division, the best classification effect can be obtained. Then, the data set is divided according to this attribute and the new optimal partition attribute corresponding to the next node of CART can obtained by using (9), (10) and (11) repeatedly.

From the above establishment of CART, we can think that the more times an attribute appears on nodes of CART, the more important it is for classification. Moreover, because the Gini index of each node attribute is calculated according to its parent-node, we have reason to believe that the closer an attribute is to the root node, the more important this attribute is to the classification.

Based on this discovery, we developed a method to express the importance of each attribute in CART as an attribute weight. Then, the weights are used to influence the process of adding noise in differential privacy in Section III.

D. DEEP LEARNING

Deep learning, which has been widely used in various fields, adopts back propagation (BP) algorithms to indicate how a machine adjusts its internal parameters to discover complex structures in large data sets. A complete deep neural network consists of an input layer, hidden layer and output layer. In deep neural networks, the selection of the activation function, the definition of the loss function, the selection of the gradient descent method and the value of the hyperparameter are closely related to the experimental results.

More precisely, we adopt rectified linear units (ReLUs) as our activation functions which have faster convergence rate and lower computational complexity and can solve the problem of gradient vanishing in sigmoids. Moreover, we define a loss function $\mathcal{L}$ that represents the penalty for mismatching the training data and calculate it by mean-square error (MSE), which means that loss $\mathcal{L}(\theta)$ on parameters θ is $\frac{1}{N}\Sigma_i \mathcal{L}(\theta, x_i)$, where x_i represents the training examples $\{x_1, \ldots, x_n\}$. The purpose of training is to find the smallest loss.

The loss function $\mathcal{L}$ is difficult to minimize due to complex networks. In practice, the mini-batch gradient descent (MBGD) algorithm is often used to find the smallest loss. It computes the gradient $\nabla_\theta \mathcal{L}(\theta)$ based on random examples whose quantity is b, and the formula is $g_b = 1/|b| \sum_{x \in b} \nabla_\theta \mathcal{L}(\theta, x)$. Then, we update θ based on the gradient and learning rate to find a local minimum.

There have been several systems with excellent performance to support neural networks [27], [28]. Moreover, our work is performed based on TensorFlow and PyTorch. The former is an open-source dataflow engine released by Google, while the latter is an open-source dataflow engine released by Facebook.

III. DESIGN METHODOLOGY

This section describes the main components of our approach toward differential privacy for data publishing and differentially private training of neural networks: Normalization, DPDT and DPMB.

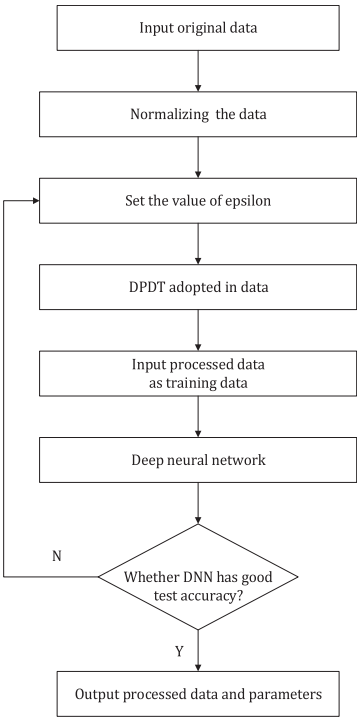


FIGURE 1. The process of protecting the publishing data.

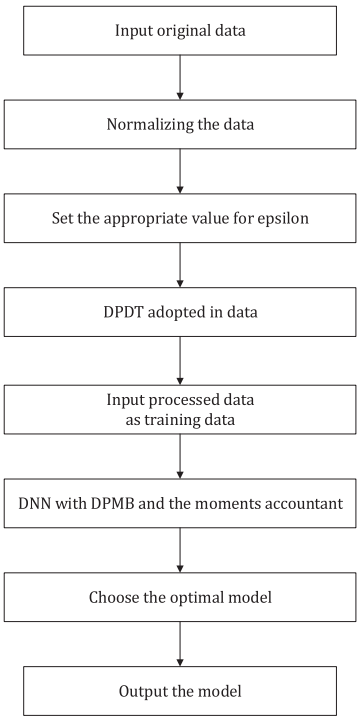


FIGURE 2. The process of protecting the publishing model.

Fig. 1 and Fig. 2 roughly show the application flow of our method. The specific process is detailed below.

A. NORMALIZATION

Many medical data have the characteristic that the variance of numerical distribution is too large, which has a great influence on the accuracy of machine learning. At the same time, it makes the local sensitivity of the data properties so large that data availability becomes extremely poor because we need to add larger noise to the data while applying differential privacy.

Through normalization, we make the data distribution more concentrated and make the local sensitivity of the data the same as the overall sensitivity, which facilitates the application of differential privacy.

We transform data into a matrix of m ∗ n, using $x_1, \ldots, x_n$ to represent each sample, $a_1, \ldots, a_n$ to represent each attribute, and a_{ij} to represent the value of attribute a_j of sample x_i.

$$D = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \tag{12}$$

We process data by zero-centering. First, calculate the average of each attribute:

$$\overline{a_j} = \frac{\sum_{i=1}^{m} a_{ij}}{m} \qquad (j = 1, 2, \ldots, n). \tag{13}$$

Then, use $a_{ij} - a_j$ to get zero-centered data:

$$D = \begin{bmatrix} a_{11} - \overline{a_1} & a_{12} - \overline{a_2} & \cdots & a_{1n} - \overline{a_n} \\ a_{21} - \overline{a_1} & a_{22} - \overline{a_2} & \cdots & a_{2n} - \overline{a_n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} - \overline{a_1} & a_{m2} - \overline{a_2} & \cdots & a_{mn} - \overline{a_n} \end{bmatrix}. \tag{14}$$

After zero-centering, we normalize the data. First, we define the range variable r and then calculate the maximum difference corresponding to each attribute as $d_1, d_2, \ldots, d_n$:

$$d_j = \max(a_{ij} - \overline{a_j}) - \min(a_{ij} - \overline{a_j}) \quad (j = 1, 2, \ldots, n). \tag{15}$$

Finally, compress the data to the range$(-\frac{r}{2}, \frac{r}{2})$:

$$D = \begin{bmatrix} \frac{(a_{11} - \overline{a_1}) \cdot r}{2 \cdot d_1 \cdot \max|a_{i1} - \overline{a_1}|} & \cdots & \frac{(a_{1n} - \overline{a_n}) \cdot r}{2 \cdot d_n \cdot \max|a_{in} - \overline{a_1}|} \\ \frac{(a_{21} - \overline{a_1}) \cdot r}{2 \cdot d_1 \cdot \max|a_{i1} - \overline{a_1}|} & \cdots & \frac{(a_{2n} - \overline{a_n}) \cdot r}{2 \cdot d_n \cdot \max|a_{in} - \overline{a_1}|} \\ \vdots & \ddots & \vdots \\ \frac{(a_{m1} - \overline{a_1}) \cdot r}{2 \cdot d_1 \cdot \max|a_{i1} - \overline{a_1}|} & \cdots & \frac{(a_{mn} - \overline{a_n}) \cdot r}{2 \cdot d_n \cdot \max|a_{in} - \overline{a_1}|} \end{bmatrix}. \tag{16}$$

It is noteworthy that r is closely related to the sensitivity in differential privacy. We can limit the sensitivity by limiting r, thus reducing the negative impact of differential privacy on data availability. In addition, r is flexible. We can find the optimal solution of r through iterative experiments.

B. THE COMBINATION OF DIFFERENTIAL PRIVACY AND DECISION TREE

As we all know, regardless of whether it is medical data or some other form of data, there are always some

attributes that have a great impact on the classification results, while others have little impact. One might attempt to protect the publishing data via differential privacy against attacks, such as linking attacks, skewness attacks, and background knowledge attacks. However, if you apply excessive noise to a vital attribute, data availability will be greatly reduced (this can be reflected by the test accuracy of the machine learning model). Therefore, if we can identify the factors with a large impact on classification results and then add mitigatory noise to these factors in the process of differential privacy, the data availability will be greatly increased.

Both feature extraction and feature selection can extract important attributes. Feature extraction methods, such as PCA, LDA, and SVD, will perform data transformation and dimension reduction such that the processed data cannot be published and provide statistical analysis and other functions. Similarly, with the feature selection methods, we also need to exclude the methods that will perform data transformation and dimension reduction. Fortunately, there are still many methods in feature selection that can obtain important features and meet the requirements of data publishing without conducting data transformation and dimension reduction, such as Relief (Relevant Features) and information entropy. These methods can reflect which attributes are important, but they cannot provide us with a specific weight value.

CART can statistically demonstrate which factors are particularly important in a model or relationship in terms of explanatory power and variance [26]. Moreover, CART is constructed by feature selection, and each node is based on the Gini index to select the most important partition attributes. By building CART, we can clearly analyze which attributes are important by observing the attributes of nodes. Then, by initializing the weight of the decision tree, we can obtain the specific weight value corresponding to each attribute and combine it with differential privacy (e.g. (19)) to realize the process of adding different levels of noise to different important attributes to reduce the negative impact of differential privacy on data availability. It is worth mentioning that this method is simple and fast.

1) THE PROCESS OF INITIALIZING ATTRIBUTE WEIGHTS

First, we use the data that have been normalized as the training data of CART to get one decision tree and compute the depth of the tree as d. Then, we assign a weight value of $d-1$ to the attribute that is on the first layer of the tree, assign a weight value of $d-2$ to attributes that are on the second layer of the tree, and so on, with the weight decreasing progressively with each layer. Next, we calculate the total weight of each attribute as $W_1, W_2, \ldots, W_n$. Finally, we normalize each weight. For example:

From Fig. 3, we give each layer a weight and calculate the weight of each attribute. We can calculate that the weight of a_1 is 5 $(3+1+1)$ and then assign this value to W_1.

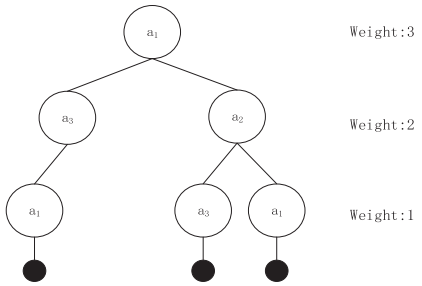


FIGURE 3. Initializing weights based on CART.

Normalize each weight according to the following formula:

$$W_i = \frac{W_i}{\sum_{j=1}^{n} W_j} \quad (i = 1, 2, \ldots, n). \tag{17}$$

2) USING THE WEIGHT OF EACH PROPERTY TO INFLUENCE DIFFERENTIAL PRIVACY

Although traditional differential privacy can provide enough privacy guarantees for publishing data, the data availability will drop rapidly. As seen from the differential privacy formula (2), when ε is fixed, the magnitude of Δf is proportional to the magnitude of the final added noise. f contains many functions, such as finding the sum, average, and maximum. In security analysis, we usually regard f as a summation function. Before the data are processed by normalization, the local sensitivity is different. However, after normalization processing, the values of all local and global sensitivities are unified to r. For two datasets with only one record differential, only when the different records correspond to the maximum and minimum, respectively, will Δf be maximized and its value is r $(\Delta f = r)$. Then, when r in normalization is fixed, that is, when Δf is fixed, we use the attribute weights W_i calculated in the previous step to impose different noise on different attributes a_i according to formula (20). This is different from the traditional method in which all attributes impose the same level of noise regardless of the importance of attributes to classification. Therefore, in our method, if an attribute has a greater impact on the classification results, it will get more moderate noise, thus reducing the negative impact of traditional differential privacy and balancing data availability with privacy security.

The original differential privacy formula is:

$$Pr[\mathcal{K}(d) \in S] \leq e^{\varepsilon} Pr[\mathcal{K}(d') \in S]. \tag{18}$$

Its noise adding formula is:

$$\mathcal{K}(d) \triangleq f(d) + Lap(0, \frac{\Delta f}{\varepsilon}). \tag{19}$$

We implement DPDT by improving the traditional noise adding formula:

$$\mathcal{K} \triangleq f(a_i) + Lap(\frac{\Delta f \cdot (1 - \sqrt{W_i})}{\varepsilon}) \quad (i = 1, 2, \ldots, n). \tag{20}$$

It is noteworthy that the differential privacy with the Laplace mechanism is used for data publishing, while the

Gaussian mechanism is adopted in the DPMB algorithm for model publishing in the process of gradient descent in the training model. Mixed use of the two mechanisms can complicate data and models and enhance their security.

C. DIFFERENTIALLY PRIVATE MINI-BATCH GRADIENT DESCENT ALGORITHM

In recent years, many attacks on models have been put forward such as the MI attack [8] and membership inference attack [9]. Through these attacks, an adversary may be able to extract parts of the training data. Thus, many people attempt to protect the privacy of training data by conducting research only on the final parameters of the training process. However, the same as for the protection of published data, our problem is also that the availability of the model will be destroyed if we add overly conservative noise to parameters on the basis of the worst-case analysis.

Therefore, we use the DPMB algorithm to solve this problem and, in particular, combine the moments accountant with it. In this part, we use a variant differential with the Gaussian mechanism.

The variant differential privacy formula is:

$$Pr[\mathcal{K}(d) \in S] \leq e^{\varepsilon} Pr[\mathcal{K}(d') \in S] + \delta. \tag{21}$$

Its noise adding formula is:

$$\mathcal{K}(d) \triangleq f(d) + \mathcal{N}(0, S^2(f) \cdot \sigma^2). \tag{22}$$

Algorithm 1 simply introduces the process of finding the optimal θ by reducing the loss function $\mathcal{L}(\theta)$. Unlike traditional MBGD, we add noise to gradients for privacy protection. At the same time, the l_2 norm is used to avoid model overfitting. Eventually, we calculate the privacy loss based on the moments accountant. In Algorithm 1, an epoch contains M/B iterations, where one iteration requires B samples, B samples constitute a Big-batch, and a Big-batch contains B/b batches. Moreover, we use q to represent the sampling probability, T to represent the total number of iterations in the training process, and E to represent the number of epochs. The algebraic relationships of the parameters are as follows: $q = B/M$ (each epoch consists of $1/q$ Big-batches i.e. B/qb batches) and $T = E/q$. Next, we will describe the composition of the algorithm in detail.

Norm clipping: To ensure that Algorithm 1 meets the (ε, δ)-differential privacy, we need to limit the impact of each individual example by replacing g with $\mathrm{g}/\max(1, \frac{||\mathrm{g}||_2}{R})$. From the above formula, it can be seen that g will be saved in the case of $||\mathrm{g}||_2 \leq R$ while g will be reduced to R in the case of $||\mathrm{g}||_2 > R$.

Big-batches: In Algorithm 1, like the general MBGD algorithm, we calculate the gradient based on a batch of examples. However, the difference is that we also define a group called Big-batch. One Big-batch consists of B samples randomly selected from all samples with sampling probability q ($q = B/M$). Moreover, Big-batch and batch are two parameters with a clear division of labor. We calculate the gradient in batches, and then B/b batches form a Big-batch for the addition of noise. It is worth mentioning that an epoch is composed of M/B Big-batches in the experiment.

Algorithm 1 DPMB (Outline)

Require: A data set consisting of M samples $\{x_1, \ldots, x_M\}$, loss function $\mathcal{L}(\theta) = \frac{1}{M}\Sigma_i \mathcal{L}(\theta, x_i)$. Batch size b, Big-batch size B, gradient norm bound R, initial learning rate η and total number of iterations T. **Initialize** θ_0 randomly

In one epoch:

for $t = 1, 2, \ldots, M/B$ **do**

  Sampling B_t samples from all samples

  for $i = 1, 2, \ldots, B_t/b$ **do**

    Calculate gradient

    for $j = 1, 2, \ldots, b$ **do**

      $\mathrm{g}_{b_i}(x_j) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_j)$

      Normalize gradient

      $\overline{\mathrm{g}}_{b_i}(x_j) \leftarrow \mathrm{g}_{b_i}(x_j)/\max(1, \frac{||\mathrm{g}_{b_i}(x_j)||_2}{R})$

    end for

    $\overline{\mathrm{g}}_{b_i} \leftarrow \frac{1}{b}\Sigma_j \overline{\mathrm{g}}_{b_i}(x_j)$

  end for

  Add noise

  $\widetilde{\mathrm{g}}_t \leftarrow \frac{b}{B}\Sigma_i(\overline{\mathrm{g}}_{b_i} + \mathcal{N}(0, \sigma^2 R^2 I))$

  Reduce learning rate

  $\eta_t = \frac{1}{2}(1 + \cos(\frac{t\pi}{T}))\eta$

  Descent

  $\theta_{t+1} \leftarrow \eta_t \widetilde{\mathrm{g}}_t$

end for

Ensure: θ_T

Dynamic Change of Learning Rate: Generally, the learning rate is fixed. However, too high a learning rate may skip the global optimum, and too low a learning rate will make the model fall into local optima. We introduce a dynamic learning rate, which changes with the number of iterations in the training process. In this manner, we can not only accelerate the training speed but also find the global optimum more effectively. The formula for dynamically changing the learning rate is as follows:

$$\eta_t = \frac{1}{2}(1 + \cos(\frac{t\pi}{T}))\eta. \tag{23}$$

In the above formula, η_t represents the learning rate corresponding to the tth iteration and η represents the initial learning rate. For example, we set η as 1 and T as 300. The change of learning rate during training is shown in Fig. 4.

D. THE MOMENTS ACCOUNTANT IN DNN

To analyze data privacy, numerous studies have begun to focus on privacy losses. For the Gaussian noise, according to the standard arguments [10], the privacy amplification theorem [30] and the strong composition theorem [18], we can get the best overall bound. Nevertheless, without considering the particular noise distribution, the strong composition theorem can be loose. The moments accountant proposed by Abadi et al. in 2016 [23] is beneficial in both theory and practice,

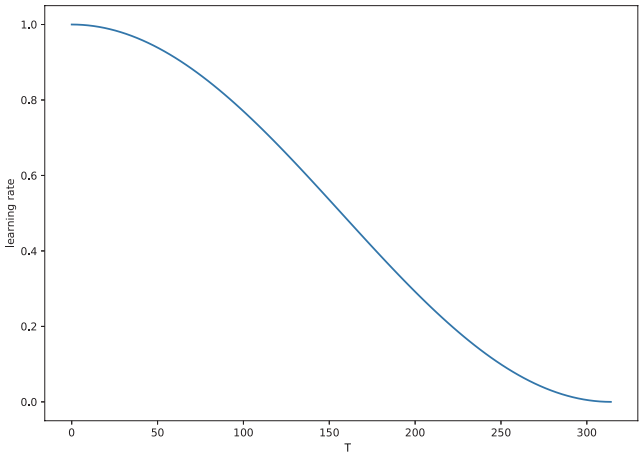


FIGURE 4. Dynamic change of learning rate.

so we use it to keep track of a bound on the moments of the privacy loss in DNN.

The most difficult task is to calculate $\alpha_{\mathcal{K}_t}(\lambda)$ for each step. For the Gaussian mechanism, $\alpha(\lambda)$ is defined as $\alpha(\lambda) = \log \max(E_1, E_2)$. In the above formula,

$$E_1 = \mathbb{E}_{z\sim\mu_0}[(\mu_0(z)/\mu(z))^{\lambda}], \tag{24}$$

$$E_2 = \mathbb{E}_{z\sim\mu}[(\mu(z)/\mu_0(z))^{\lambda}]. \tag{25}$$

μ_0 denotes the probability density function (pdf) of $\mathcal{N}(0, \sigma^2)$, and μ_1 indicates the pdf of $\mathcal{N}(1, \sigma^2)$. Moreover, we define μ and let $\mu = (1-q)\mu_0 + q\mu_1$.

Then, we can compute $\alpha(\lambda)$ by numerical integration and get the asymptotic bound

$$\alpha(\lambda) \leq O(q^3/\sigma^3) + q^2\lambda(\lambda+1)/(1-q)\sigma^2. \tag{26}$$

Next, by (7) mentioned in Section II, we can bound $\alpha_{\mathcal{K}}(\lambda)$ at each step and overall and we can use the tail bound to convert the moments bound to the (ε, δ)-differential privacy guarantee through (8).

Moreover, the details of comparing the strong composition theorem and the moments accounts can be found in the paper [23].

IV. EXPERIMENTAL RESULTS

This section evaluates the normalization, DPDT and DPMB algorithms. We choose one popular medical data set: Diabetes [1].

A. EXPERIMENTAL MODEL

This subsection can be divided into two parts. The first part is the experimental model of DPDT for publishing data. The second part is the experimental model of DPMB for model publishing.

1) THE MODEL OF DPDT

We use the weight of each property to influence differential privacy, and construct CART in scikit-learn with normalized data as the training data to calculate weights. Then, after calculating the weight of each attribute based on CART, we use the method mentioned in Section III to impose differential privacy on the data with weighted influence. Next, we construct a deep neural network (named *Deep_A*) in TensorFlow and PyTorch with $\mathcal{L}(w) = \frac{1}{2m}\sum_{i=1}^{m}||y_i' - y_i||_2^2 + \frac{\lambda}{2m}\sum_{l=2}^{L}||w||_2^2$ (mean square error and l_2 norm) as the loss function and with the general MBGD algorithm. Finally we use differentially private data as training data to observe data availability through the deep neural network we just constructed.

2) THE MODEL OF DPMB

We have implemented DPMB algorithms mentioned in Subsection III-C in a new neural network named *Deep_B* by TensorFlow and PyTorch. We construct a *sanitizer*, which safeguards privacy via gradient pretreatment, and *privacy_accountant* [23], which follows the tracks of the privacy loss over the procedure of training. Other model information regarding *Deep_B* has been provided in Subsection II-D.

We utilize *DPMB_Optimizer*, which finds the minimum point of the loss function via DPMB, and *DPTrain*, which iteratively calls *DPMB_Optimizer* via the moments accountant to limit the total privacy loss. The code of *DPMB_Optimizer* and *DPTrain* is detailed in [23].

B. NORMALIZING THE DATA

To facilitate the use of differential privacy and reduce data variance for publishing data, we have normalized the data. We compare the availability of raw data with that of processed data for machine learning through *Deep_A*, with ReLU as the activation function, setting the batch size $b = 20$, range variable $r = 4$, $dropout = 0.4$, and learning rate $\eta = 0.01$.

From Fig. 5, it can be distinctly seen that both the testing accuracy of processed data and the fitting speed are much better than those of the original data. It can be observed that the availability of medical data for machine learning may be truly poor because of the large variance of the data or the problem of data quality. Normalizing the medical data exactly can improve both model accuracy and convergence rate.

C. DPDT WITH THE LAPLACE MECHANISM FOR DATA PUBLISHING

To ensure balance between the availability and privacy of medical data, we add mitigated noise that is influenced by the weight coefficient of each property to the medical data during the process of differential privacy based on the detailed methods in Section III.

We divide privacy security into three scales, which are called small ($\varepsilon = 8$), medium ($\varepsilon = 2$) and large ($\varepsilon = 0.5$). Here ε represents the privacy parameters of differential privacy, reflecting the magnitude of the noise applied to the data. Then we use *Deep_A*, whose initial learning rate η is set to 0.01 and batch size b is set to 20 to observe the testing accuracy of the noisy data to reflect data availability and set the sensitivity of differential privacy at r ($\Delta f = r$)

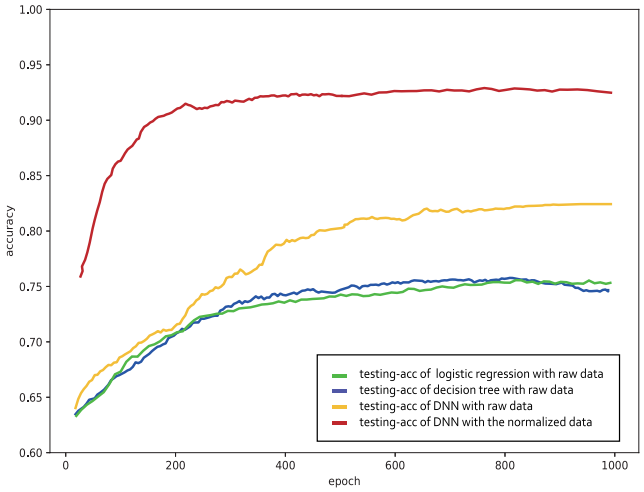


FIGURE 5. Raw medical data and normalized medical data.

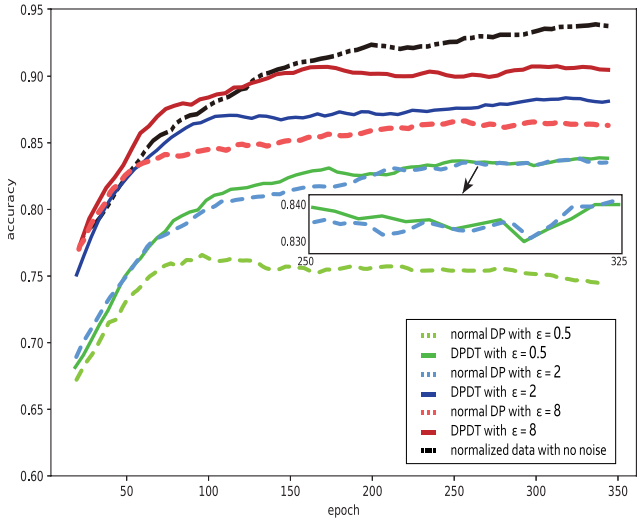


FIGURE 6. Comparison of model testing accuracy between DP and DPDT. In this experiment, we use the normalized data as training data of *Deep_A*. What's more, the ε level is set at 0.5, 2 and 8 in turn.

which is imbued with a specific value in the process of normalizing. What's more, the training data of *Deep_A* is the normalized data. It is worth mentioning that the learning rate will decrease with the increase of epoch as in Fig. 4.

Comparing the solid and dotted lines in Fig. 6, we might clearly find that direct application of differential privacy to data does provide strong privacy guarantees for data, but at the same time, it does greatly reduces data availability and exhibits the problem that high data availability leads to poor security and high security almost leads to data unavailability. However, our approach has achieved remarkable results in the effective combination of security and data availability.

In terms of security, DPDT does add less noise than traditional differential privacy when setting the same ε. In other words, our approach has a slight loss in model security compared with traditional differential privacy. We try to convert ε in DPDT into *true* ε corresponding in differential privacy

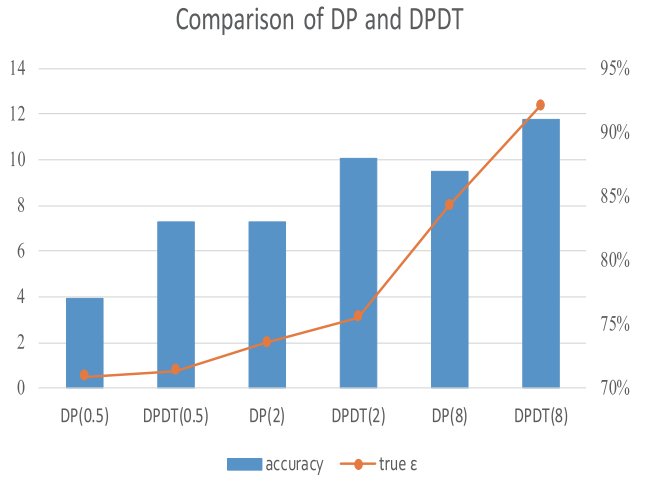


FIGURE 7. Comprehensive analysis of the security and accuracy of DP and DPDT.

while keeping the noise at the same level. First, we transform $Lap(\frac{\Delta f \cdot (1-\sqrt{W_i})}{\varepsilon})$ (20) into $Lap(\frac{\Delta f}{\varepsilon/(1-\sqrt{W_i})})$. Then, the value of Δf is fixed through normalization. Thus, if we set the value of ε and calculate the value of W_i, we can get the noise we need to add. In our experiment, the attribute of the diabetes pedigree function (a_7) has the highest weight (W_7) among all attributes. We take this attribute as an example to calculate the safety loss of DPDT compared with traditional differential privacy. Through CART, we calculate $W_7 = 58/357$. If we set $\varepsilon = 0.5$ in DPDT, then the *true* ε of a_7 corresponding to traditional differential privacy is 0.84 (calculated by $\varepsilon/(1-\sqrt{W_7})$). That is, when ε is 0.5, the noise added by DPDT to the attribute of a_7 is similar to that applied by traditional differential privacy when ε is 0.84. After that, we calculate the values of *true* ε of the other attributes. Finally, we average the *true* ε of all attributes as the final *true* ε corresponding to differential privacy. In this manner, the values of *true* ε corresponding to differential privacy are 0.77, 3.09 and 12.37 when ε in DPDT is set to 0.5, 2 and 8 respectively. Therefore, in terms of security and accuracy, we make a comprehensive analysis of DPDT and traditional DP by Fig. 7. From Fig. 7, we can see that data availability (represented by classification accuracy) has been greatly improved with a slight loss of security. At the same time, comparing DPDT (0.5) and DP (2) in Fig. 7, we can find that DPDT can provide more powerful privacy protection under the same testing accuracy.

After verifying the availability of differentially private data that is influenced by weights (calculated by CART), we perform the inverse operation of normalizing to restore the data and publish the restored data, while publishing $(\overline{a_1}, \ldots, \overline{a_n})$, maximum difference $(d_1, \ldots, d_n)$, range variable (r) and one black-box that can apply DPDT to data as well. In this manner, for new data (previously not in a published dataset), we can normalize it according to published parameters and apply the black-box to making it consistent with published data.

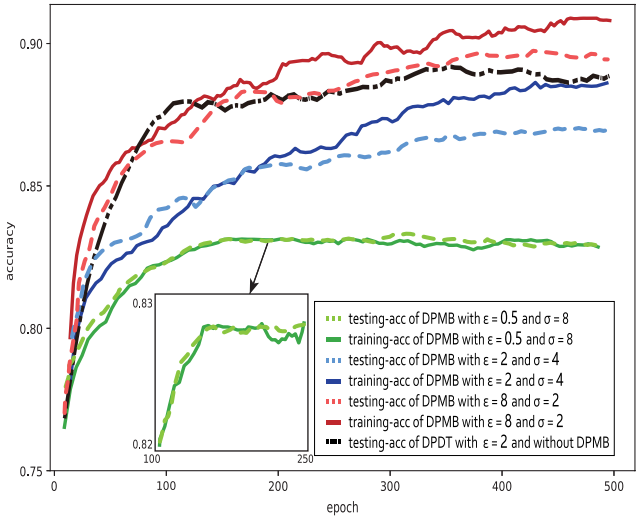


FIGURE 8. Results on the accuracy for DPMB in *Deep_B*. We use the data, which have been applied to DPDT with $\varepsilon = 2$, as training data. In this experiments, the ε level is set at 0.5, 2 and 8 in turn.

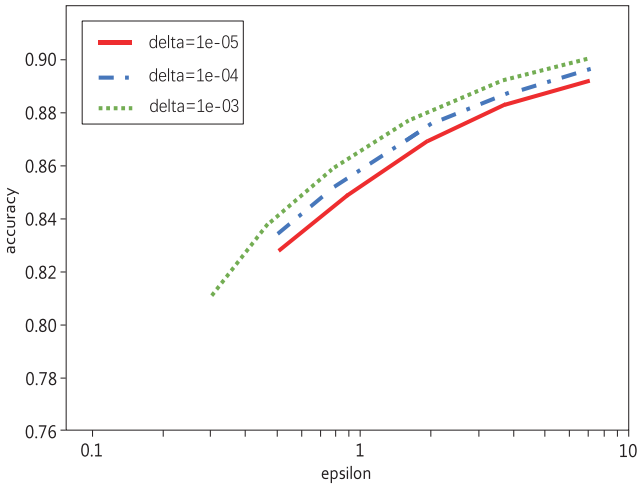


FIGURE 9. Accuracy of various (ε, δ) privacy values on the Diabetes dataset.

TABLE 1. Summary of results.

	Raw data	Normalization	Normal DP	DPDT	DPMB		
					S	MED	L
Acc	82%	93%	S 87%	S 91%	92%	89%	84%
			MED 83%	MED 88%	89%	87%	83%
			L 77%	L 83%	82%	79%	75%

D. DPMB ALGORITHM WITH A GAUSSIAN MECHANISM FOR MODEL PUBLISHING

The DPMB algorithm is proposed to prevent various model attacks while the moments accountant provides a tighter bound on the privacy loss. We use the data that have been applied to DPDT with $\varepsilon = 2$ as training data in *Deep_B* to compare with the normal MBGD algorithm in *Deep_A*. Moreover, we show the dynamic changes between privacy parameters and testing accuracy via the moments accountant.

In our experiment, we divide the data into the training set and test set with proportions of 500 : 268. In addition, we utilize $q = 0.1$, $\delta = 10^{-5}$, Big-batch size $B = 50$, batch size $b = 10$, initial learning rate $\eta = 0.01$, a 200-unit ReLU hidden layer, and a 20-unit ReLU hidden layer. To reduce sensitivity, we choose a value for C by taking the median of the norms of the unclipped gradients over the course of training. Like the upper part, we define three levels of noise scale, which are small ($\sigma = 2$), medium ($\sigma = 4$), and large ($\sigma = 8$).

Fig. 8 displays the results for different noise levels. In each plot, it shows how the training accuracy and testing accuracy change with epochs. In turn, we achieve 83%, 87%, and 89% testing accuracy for $(0.5, 10^{-5})$, $(2, 10^{-5})$, and $(8, 10^{-5})$-differential privacy, respectively.

We can discover that the difference between the model accuracy in the training and testing is so small by applying DPMB, which conforms with the theory that differentially private training generalizes well [31]. In contrast, the gap between training and testing accuracy is evidence of over fitting, and it will increase with the number of epochs. Moreover, we surprisingly find that the accuracy becomes higher when we apply a small noise, which is consistent with what Ian Goodfollow said: differential privacy is the friend of machine learning.

Moreover, through the moments accountant, we can get a δ value for any given ε and we record the accuracy for different (ε, δ) pairs in Fig. 9. This intuitive representation allows us to select appropriate privacy parameters while ensuring the model accuracy.

Finally, all experimental results, including those not mentioned in the text but achieved, are summarized in Table 1. In addition, S, MED and L represent small, medium and large noises corresponding to those above, respectively.

From all of the experiments and Table 1, we can observe the following conclusions.

1. The quality of medical data is often mediocre, and normalizing data can greatly improve data availability.

2. When applying differential privacy to data, DPDT can greatly increase data availability by considering the weight of attributes on the premise of guaranteeing data privacy.

3. The model can be protected at low accuracy loss via the DPMB algorithm and the moments accountant, and we can have direct guidance to expediently choose privacy parameters via a statistical chart.

V. RELATED WORK

Since the late 1990s, both privacy-preserving data publishing and machine learning have been areas of energetic work from several research groups.

Early research on privacy preservation studied the secure function evaluation (SFE) [32] and secure multiparty computation (MPC) [33], which focus on minimizing the information revealed. However, they cannot guarantee that the information will not leak at all, so we are more concerned

with how to protect personal privacy under the assumption that information is leaked.

With the development of technology and theory, K-anonymity was proposed by Sweeney et al. in 2002 [5]. Subsequently, L-diversity [6], T-closeness [7] and M-invariance [34] were proposed. They achieved good results under the assumption that the attacker has no or only slight of background knowledge of sensitive information. Generally, these early privacy models have two shortcomings: one is the lack of a strict mathematical definition of privacy protection, which makes it impossible to quantify the degree of privacy protection or disclosure; the other is the hypothetical restriction on the background knowledge of the attacker. However, we in fact cannot predict how much background knowledge the attacker has. Therefore, we expect a privacy model that can strictly define the degree of privacy protection mathematically without assuming the background knowledge of the attacker.

In 2006, Dwork et al. proposed the definition of differential privacy [13]. This method meets the needs of previous researchers and can provide strong privacy protection for data privacy. The research on differential privacy can generally be divided into two directions: one is data publishing under privacy protection [13], [18], [35]–[45]; the other is privacy protection for machine learning [29], [40], [46]–[54].

For data publishing, many methods for set-valued data publishing [42], [43] have been proposed for data mining, statistical query, data analysis, etc. rather than for classification. In addition, there are some secure data publishing methods that are effective but not applicable to medical data, such as [44]. Because the sensitive data contained in medical data is not only the final label but also various sensitive attributes, the method in [44] requires a great limitation on the attackerąŕs background knowledge, and the security level is insufficient. However, not much is found in papers that address the privacy preservation to achieve the goal of classification [45], [54]. The content of [45] is the closest to our work, and the data are also medical data. In [45], anonymity technology and differential privacy are combined to protect published medical data, which enhances the security of the published data. Then, the protected data is treated as the training data of the C4.5 model to observe the classification effect, and the classification accuracy is enhanced by increasing the taxonomy tree depth (TTD). The results in [45] show that the data quality of training data processed by differential privacy is damaged, and thus the classification accuracy is greatly reduced. To solve this problem, we adopt two methods, normalization and DPDT, to improve the quality of data and reduce the negative impact of differential privacy on data availability. Thereupon, data availability and privacy can achieve a better balance.

For machine learning, differential privacy has been combined with decision tree [51], [52] and with logistic regression by Chaudhuri and Monteleoni [53]. We attempted to use logistic regression and decision tree to train Diabetes [1], but we obtained a very low training accuracy of 75%. Therefore, we combine differential privacy and deep learning to give training data stronger guarantees in the case of slight accuracy loss via the DPMB algorithm, which achieves good results. Moreover, the combination of MPC and differential privacy is widely used [47]–[50]. They build a distributed aggregation classifier and divide the sensitive information into several parties as training data, thus avoiding prohibitive amounts of noise and ensuring model validation. However, in our experiment, the total number of our training data points is only 500. Because of the small amount of training data, the establishment of a distributed aggregation classifier will only cause the model to under-fit. In terms of noise limitation, we use the moments accountant [23] to get a much tighter estimation of the privacy loss to avoid excessive noise addition, which is more effective and convenient for a small amount of training data. In terms of security, even with the model generated by the differential private algorithm, it is still possible to perform training data disclosure [54]. Hence, we use DPDT protected data instead of sensitive data as training data, thus further enhancing the security of the data. We prefer that training data and models can be published for research without revealing personal privacy, while sensitive data cannot be released directly. In terms of model complexity, our model is simpler and equally effective.

VI. CONCLUSION

In this paper, we use normalization to improve data quality, achieving 93% testing accuracy which is 11% higher than that before without data processing. In addition, we achieve 83%, 88% and 91% testing accuracy by defining the variable r in normalization, by determining the method of calculating the weight of each attribute and by changing the traditional differential privacy. We can see that DPDT offers a more significant improvement in data availability than normal differential privacy. Moreover, we use the data processed with normalization and DPDT as training data, proving that the model can obtain much tighter estimates on the overall privacy loss and give training data stronger guarantees in the case of slight loss of accuracy via the DPMB algorithm and the moments accountant. More interestingly, the DPMB algorithm can effectively prevent over-fitting, and slightly improve the accuracy of the model in some uncertain situations. Through the above analyses, it can be seen that we have achieved a better balance between privacy and availability, and initially solved the problem of privacy disclosure in data and model publishing.

In addition, the model and training data can be further improved. In particular, we would like to explore other classes of deep networks such as the combination of LSTM and CNN, which may contribute to our experience. The combination of MPC, differential privacy and the moments accountant may better limit privacy loss and noise addition for a large dataset. Moreover, we would like to achieve additional improvements in training datasets. Many training datasets contain more data than Diabetes [1], and the accuracy of the model will be improved by the amount of data. However,

public medical datasets with such a large amount of data are unfortunately difficult to obtain.

REFERENCES

[1] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[2] M. de Bruijne, "Machine learning approaches in medical image analysis: From detection to diagnosis," *Med. Image Anal.*, vol. 33, pp. 94–97, Oct. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841516301098

[3] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data," *J. Biomed. Inform.*, vol. 50, pp. 234–243, Aug. 2014.

[4] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE Access*, vol. 4, pp. 2751–2763, 2016.

[5] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[6] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, 2006, p. 24.

[7] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, 2007, pp. 106–115.

[8] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1322–1333.

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[10] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[11] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48901–48911, 2019.

[12] S. Chang and C. Li, "Privacy in neural network learning: Threats and countermeasures," *IEEE Netw.*, vol. 32, no. 4, pp. 61–67, Aug. 2018.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.* Springer, 2006, pp. 265–284.

[14] C. Dwork, "A firm foundation for private data analysis," *Commun. ACM*, vol. 54, no. 1, pp. 86–95, Jan. 2011.

[15] C. Dwork, "Differential privacy," in *Encyclopedia of Cryptography and Security*. Boston, MA, USA: Springer, 2011, pp. 338–340. doi: 10.1007/978-1-4419-5906-5_752.

[16] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Springer, 2006, pp. 486–503.

[17] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. STOC*, vol. 9, 2009, pp. 371–380.

[18] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Oct. 2010, pp. 51–60.

[19] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, Jun. 2017.

[20] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*. [Online]. Available: https://arxiv.org/abs/1603.01887

[21] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptogr. Conf.* Springer, 2016, pp. 635–658.

[22] F. D. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 19–30.

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

[24] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 263–275.

[25] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," 2016, *arXiv:1610.05755*. [Online]. Available: https://arxiv.org/abs/1610.05755

[26] R. J. Lewis, "An introduction to classification and regression tree (CART) analysis," in *Annu. Meeting Soc. Acad. Emergency Med.*, San Francisco, CA, USA, vol. 14, 2000, pp. 1–15.

[27] R. Ierusalimschy, L. Figueiredo, and W. Celes, "Lua—An extensible extension language," *Softw., Pract. Exper.*, vol. 26, no. 6, 1996, Art. no. 635652.

[28] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *Proc. BigLearn, NIPS Workshop*, 2011, pp. 1–6.

[29] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM J. Comput.*, vol. 40, no. 3, pp. 793–826, Jun. 2011.

[30] A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," *Mach. Learn.*, vol. 94, no. 3, pp. 401–437, Mar. 2014.

[31] R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman, "Algorithmic stability for adaptive data analysis," in *Proc. 48th Annu. ACM Symp. Theory Comput.*, 2016, pp. 1046–1059.

[32] S. Micali and P. Rogaway, "Secure computation," in *Proc. Annu. Int. Cryptol. Conf.* Springer, 1991, pp. 392–404.

[33] O. Goldreich, "Secure multi-party computation," Tech. Rep., 1998, vol. 78.

[34] X. Xiao and Y. Tao, "M-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 689–700.

[35] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," *J. JACM*, vol. 60, no. 2, pp. 12:1–12:25, Apr. 2013.

[36] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: Efficient algorithms and hardness results," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 381–390.

[37] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Oct. 2010, pp. 61–70.

[38] A. Roth and T. Roughgarden, "Interactive privacy via the median mechanism," in *Proc. 42nd ACM Symp. Theory Comput.*, 2010, pp. 765–774.

[39] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2339–2347.

[40] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *Proc. Conf. Learn. Theory*, 2012, pp. 1–24.

[41] D. D. Lee, P. Pham, Y. Largman, and A. Ng, "Advances in neural information processing systems 22," Tech. Rep., 2009.

[42] H. Zhang, Z. Zhou, L. Ye, and X. Du, "Towards privacy preserving publishing of set-valued data on hybrid cloud," *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 316–329, May 2015.

[43] S. Wang, L. Huang, Y. Nie, P. Wang, H. Xu, and W. Yang, "PrivSet: Set-valued data analyses with locale differential privacy," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 1088–1096.

[44] L. Chen, S. Zhong, L.-E. Wang, and X. Li, "A sensitivity-adaptive ρ-uncertainty model for set-valued data," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Springer, 2016, pp. 460–473.

[45] A. N. K. Zaman, C. Obimbo, and R. A. Dara, "An improved data sanitization algorithm for privacy preserving medical data publishing," in *Proc. Can. Conf. Artif. Intell.* Springer, 2017, pp. 64–70.

[46] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 989–997.

[47] M. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1876–1884.

[48] A. Rajkumar and S. Agarwal, "A differentially private stochastic gradient descent algorithm for multiparty classification," in *Artificial Intelligence and Statistics*. 2012, pp. 933–941.

[49] M. Heikkilä, E. Lagerspetz, S. Kaski, K. Shimizu, S. Tarkoma, and A. Honkela, "Differentially private Bayesian learning on distributed data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3226–3235.

[50] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6343–6354.

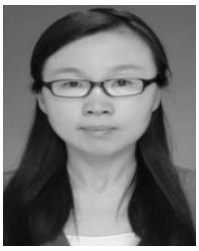
[51] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 114–121.

[52] M. Bojarski, A. Choromanska, K. Choromanski, and Y. LeCun, "Differentially- and non-differentially-private random decision trees," 2014, *arXiv:1410.6973*. [Online]. Available: https://arxiv.org/abs/1410.6973
[53] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. NIPS*, 2009, pp. 289–296.
[54] K. Boyd, E. Lantz, and D. Page, "Differential privacy for classifier evaluation," in *Proc. 8th ACM Workshop Artif. Intell. Secur.*, 2015, pp. 15–23.

ZONGKUN SUN received the B.E. degree in network engineering from the Shandong University of Science and Technology, Qingdao, China, where he is currently pursuing the M.E. degree with the College of Computer Science and Engineering. His research interests include data security, medical artificial intelligence, medical big data, and the medical Internet of Things.

YINGLONG WANG received the M.S. degree in industrial automation and the Ph.D. degree in communication and information systems from Shandong University, Jinan, China, in 1990 and 2005, respectively. He is currently a Research Fellow with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan. His current research interests include wireless networks, information security, and cloud computing.

MINGLEI SHU received the B.S. degree in automation, the M.S. degree in power electronics, and the Ph.D. degree in communication and information systems from Shandong University, China, in 2003, 2006, and 2017, respectively. Since 2006, he has been a Research Assistant with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include wireless sensor networks, wireless body area networks, and information security.



RUIXIA LIU received the M.S. degree from the Shaanxi University of Science and Technology, in 2004, and the Ph.D. degree in information science and engineering from the Shandong University of Science and Technology, Qingdao, China, in 2017. She has been an Associate Research with the Shandong Computer Science Center (National Supercomputer Center in Jinan). Her research interests include wireless body area networks for medical applications, the optimization of MAC protocol solutions, cross-layer optimization, and QoS features.

HUIQI ZHAO received the B.E. degree in computer science and technology from the Shandong University of Science and Technology, Qingdao, China, in 2003, the M.E. degree in control theory and control engineering from the Shandong University of Science and Technology, in 2009, and the Ph.D. degree from the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, in 2019.

He is currently with the Shandong University of Science and Technology. His main research interests are industrial information security, data privacy protection, cloud security, the Internet of Things security, and so on. He is also a member of the CCF, a member of the China Automation Society, a member of the China Automation Society Network Information Service Committee, and a China Information Security Certification Center CISAW Lecturer.