IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 September 2022, accepted 30 September 2022, date of publication 4 October 2022,
date of current version 12 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212031

RESEARCH ARTICLE

AdaBias: An Optimization Method With Bias Correction for Differential Privacy Protection

XUANYU ZHAO[1], TAO HU[1,2,3], (Member, IEEE), JUN LI[1], AND CHUNXIA MAO[1]
[1]College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi 445000, China
[2]Hubei Engineering Research Center of Selenium Food Nutrition and Health Intelligent Technology, Enshi 445000, China
[3]Key Laboratory of Performing Art Equipment and System Technology, Ministry of Culture and Tourism, Beijing 100007, China

Corresponding author: Tao Hu (hutao_es@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61962019 and Grant 61961017, in part by the Science and Technology Bureau of Enshi Prefecture under Grant D20190014 and Grant XYJ2021000056, in part by the National Cultural and Tourism Science and Technology Innovation Project under Grant 2021064, in part by the Hubei Engineering Research Center of Selenium Food Nutrition and Health Intelligent Technology under Grant PT082004 and Grant PT082105, in part by the Training Program of High Level Scientific Research Achievements of Hubei Minzu University under Grant PY22011, and in part by the 2022 Graduate Education Innovation Project of Hubei University for Nationalities under Grant MYK2022006.

ABSTRACT A continuous increase in privacy attacks has caused the research and application of differential privacy (DP) to gradually increase. We can improve the efficiency of the DP model by Optimizing its parameters significantly. Inspired by the performance of various optimization methods for differential privacy, this paper proposes an improved RDP-AdaBound optimization method with bias correction, which is called ''AdaBias'', to increase the performance of Rényi differential privacy (RDP). The bias correction is used to realize the learning rate and speed up the convergence by upper and lower bound functions. We evaluate our method on the three datasets by training two different privacy model. We further compare three traditional optimization algorithms, namely, RDP-SGD, RDP-Adagrad, and RDP-Adam. And we use AdaBias to verify the performance of privacy protection on the COVID-19 dataset. Experimental results show that the new variant better implements learning rate adjustment to accommodate updates of noisy gradients. As a result, it can achieve higher accuracy and lower losses with a lower privacy budget, thereby better protecting data privacy.

INDEX TERMS Differential privacy, deep learning, optimization algorithm.

I. INTRODUCTION

The development of the Internet, cloud computing, and big data continues to cause severe privacy crises. The differential privacy (DP) mechanism [1] adds noise to the data to distort sensitive data to maintain specific statistical characteristics. It uses strong privacy mathematical guarantees against the largest background knowledge attacks. DP uses parameters to quantify the degree of privacy protection, which can make up for the shortcomings of traditional privacy protection models. It is now used for privacy protection based on support vector machines, regression, decision trees, and deep learning models and has been widely used in industries such as healthcare and financial services.

Rényi differential privacy (RDP) [2], [3] provides a unified definition for some privacy concepts, such as pure DP (ϵ-DP), approximate DP ((ϵ, δ)-DP) and CDP [4]. RDP is a relaxation of pure DP, which always indicates approximate DP. (α, ϵ)-RDP has a budget curve parameterized by the order α, which combines the concept of privacy budget, the theorem's application, and the Rényi divergence to better measure the change in the loss. It provides a convenient, quantitative, and more accurate way to monitor the accumulated privacy budget loss in real time for a single differential privacy stage.

The DP learning mechanism [5], [6] usually needs to solve the privacy violation of the model itself, training and testing data. When applied to deep learning models, the training process consists of multiple modules, such as networks, evaluation functions, algorithms, and datasets. The performance of the final model also varies with the different noise addition

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

mechanisms, gradient clipping algorithms, loss functions, and optimization strategies used. An optimization problem in deep learning usually refers to finding a set of parameters on a neural network that can significantly reduce the loss function. The learning rate greatly influences the final convergence effect of the neural network in the optimization algorithm. It determines the step size of the parameter space search. A step size that is too large will result in non-convergence, but a step size that is too small will also result in slower convergence. Both articles [7] and [8] found possible privacy breaching and reconstruction attacks from published models. Therefore, an optimizer that protects model privacy while training a model becomes increasingly important. We study the similarities and differences of existing optimization algorithms and consider modifying the gradient algorithm after combining it with the RDP privacy model, aiming to provide provable privacy guarantees in training.

A. RELATED WORKS

The stochastic gradient descent algorithm (SGD) [9] is one of the most commonly used methods to solve deep learning optimization problems. Bu et al. [10] studied the scalable framework of privacy protection SGD. Song et al. [11] introduced DP-SGD and proposed a differential privacy method of single point and small-batch SGD, but how to track the privacy of the whole training process was not studied. DP-SGD adds random noise to the gradients during the optimization process [12]. Abadi et al. [6] improved the computational efficiency of DP-SGD and proposed a ''moment estimation'' method to track the cumulative privacy loss. The specific improvement of its privacy parameters and the demonstration of the practicability of its neural training model make DP-SGD one of the up-and-coming privacy machine learning methods. However, it faces two challenges: the gap between its accuracy and privacy-free methods may be significant and the considerable training time overhead (cost of gradient clipping).

Lin et al. [13] examined and quantitatively analyzed the relative effects of different factors on the performance of the model, including the optimization algorithm, noise addition order, and gradient clipping threshold. Liu et al. [14] designed an adaptive cubic quasi-Newton optimizer that can help eliminate suboptimal solutions and improve the performance of deep neural networks on medical image analysis tasks. Zhou et al. [15] introduced new second-order momentum and dynamic learning rate bounds in the proposed adaptive momentum online algorithm LightAdam, which improved the model generalization ability. FracM [16] can partially solve the trap problem of local minima and speed up the training process. Jie et al. [17] proposes a novel adaptive learning rate strategy with different layers based on the hypergradient descent framework.

The privacy budget (ϵ, δ) and verification loss are highly dependent on the selected values of the noise scale σ and learning rate η, respectively. Frisk et al. [18] focused on three optimization methods of these two super parameters in the DP-SGD model, which provides a basis for finding the balance point between privacy and utility. Zhou et al. [19] provided the proof of convergence for DP-SGD and DP-Adam and provided an empirical risk analysis for the DP variables of the adaptive gradient method. Wu et al. [20] proposed a differentially private random block coordinate adaptive gradient algorithm that randomly selects a block coordinate to update model parameters and adds Laplacian noise at each iteration. Koskela et al. [21] proposed ADADP with automatic learning rate optimization, and the performance is comparable to that of DP-SGD. DP-LSSGD [22] utilizes a Gaussian mechanism to make the trained non-convex models more stable. Chen et al. [23] proposed an RDP-SGD algorithm for convex empirical risk minimization. Anil et al. [24] proposed an improved DP-SGD algorithm and added RDP that can be converted to DP. Improving the privacy analysis part of the RDP can improve the performance of the model, Wang et al. [25] proved that the results of RDP can be amplified by downsampling, which provided the basis for the research in this paper.

B. CONTRIBUTIONS

This paper proposes a rigorous adaptive method to find a better learning rate and apply it to the RDP learning setting. We find that AdaBound achieves a gradual transition of the learning rate in the process. However, since the upper and lower bound functions are manually designed and fixed, it will affect the value of the final learning rate. Therefore, we improve the RDP-AdaBound algorithm and set a dynamic upper and lower bound function for the adaptive learning rate to prevent the calculated learning rate from upgrading quickly. Our main contributions are as follows:

- We introduce several optimization algorithms in deep learning in the Rényi DP model and improve the AdaBound to improve the level of privacy protection.
- We propose a novel method called AdaBias to imporve the performance of RDP model. We use a bias correction to the upper and lower bound functions for learning rate clipping during training, which can better realize the soft cutting of the learning rate, speeding up the model convergence process.
- Extensive experiments on three benchmark datasets demonstrate the effectiveness of the proposed AdaBias. It can improve the accuracy of the privacy model and reduce the model loss while reducing the privacy budget. And we use AdaBias to verify the performance of privacy protection on the COVID-19 dataset [26].

II. PRELIMINARIES

Fig. 1 is the train process of the RDP optimization model. First, the data are transformed and cropped by a gradient, and the differential privacy mechanism is applied to add noise and optimize the gradient. Finally, the model result is output after model training.

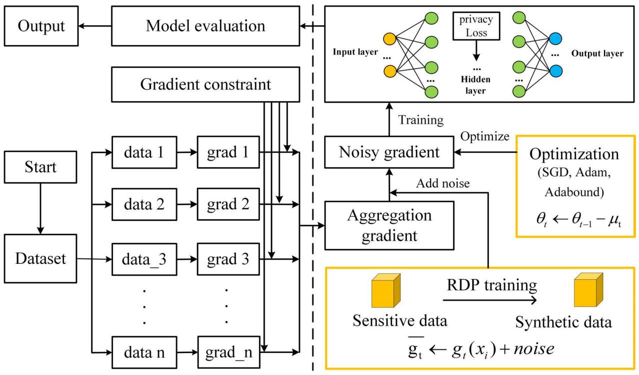


FIGURE 1. The training process of different optimization methods in the Rényi differential privacy model. Instead of calculating the gradient average of a batch of samples, we calculate the gradient of each sample and clip their ℓ_2 norm. Next, they are aggregated into a set of gradients, and then sampled Gaussian noise is added.

A. Rényi DIFFERENTIAL PRIVACY

Definition 1: (ϵ, δ)*-DP*. For neighboring databases $D, D' \in D^n$, a randomized mechanism M:D → R, and any O ∈ R satisfies the following inequality:

$$Pr[M(D) = O] \leq e^{\epsilon} Pr[M(D') = O] + \delta. \tag{1}$$

The boundary of privacy loss can also be called the privacy budget ϵ. The probability Pr[·] is controlled by the randomness of M. When $\delta = 0$, the guarantee is called pure DP, while $\delta > 0$ is a relaxation of pure DP called approximate DP. That is, the original definition of differential privacy does not include the additional term δ, and its variant allows the possibility of ϵ-DP [27] being destroyed by the probability δ. The smaller the value of ϵ, the higher the degree of privacy protection.

Definition 1: RDP. D_α is the α-Rényi divergence. A randomized mechanism M satisfies (α, ϵ)-RDP if:

$$D_\alpha(M(D') \| M(D)) \leq \epsilon. \tag{2}$$

If M obeys (α, ϵ)-RDP, then M obeys $(\epsilon + \frac{\log(1/\delta)}{(\alpha-1)}, \delta)$-DP for all $0 < \delta < 1$.

Definition 1: Sampled Gaussian Mechanism (SGM). Let f be a subset of function mapping from S to R^d, the sampling rate is 0< q < 1, the noise scale $\sigma > 0$, and each element of S is randomly and independently sampled with probability q. The SGM $SG_{q,\sigma}(S)$ of the function f in RDP is composed of subsampling and additive Gaussian noise, which is described as follows:

$$SG_{q,\sigma}(S) f(x : x \in S) + N(0, \sigma^2 I^d). \tag{3}$$

where the noise scale σ is the standard deviation of the additive Gaussian noise, which determines the privacy cost of each iteration. The privacy amplification of this mechanism is obtained through sampling. It extracts a random subset from a large dataset and then uses a function with an output space of R^d to add the variance of each coordinate of σ^2 to d-dimensional spherical Gaussian noise.

TABLE 1. Factors and types affecting the proposed method.

Abbreviation	Implicaion
T	iteration time
ρ_t	learning rate or step size at iteration t
C	gradient norm bound
σ	noise scale
(ϵ, δ)	(epsilon, delta) privacy parameters
β_1, β_2	momentum parameter
$\eta_l(t)$	lower bound function of learning rate at iteration t
$\eta_s(t)$	upper bound function of learning rate at iteration t
θ_0	initial parameter vector
$f(\cdot)$	loss function
g_t	gradient at iteration t

B. RDP OPTIMIZERS

DP-SGD has five primary hyperparameters that affect the results: several training iterations (more iterations lead to more significant privacy cost), batch size, learning rate, gradient clipping threshold, and noise scale.

Fig. 2 shows a selection of some algorithms for gradient descent analysis. SGD keeps going forward at a constant rate. The adaptive learning rate method Adagrad [28] and RMSProp [29] will immediately start in the right direction and converge at the same speed. However, Momentum [30] and its variant NAG [31] will deviate from the track, while NAG can quickly correct the route because it improves responsiveness by constantly adjusting the direction. Although the adaptive moment estimation algorithm Adam [32] that combines Momentum and RMSProp converges slowly in the early stage, it can quickly converge through a correction in the later stage. AdaBound [33] is the fastest to move in the relatively correct direction. The disadvantage is that it loses its early lead by the end of training.

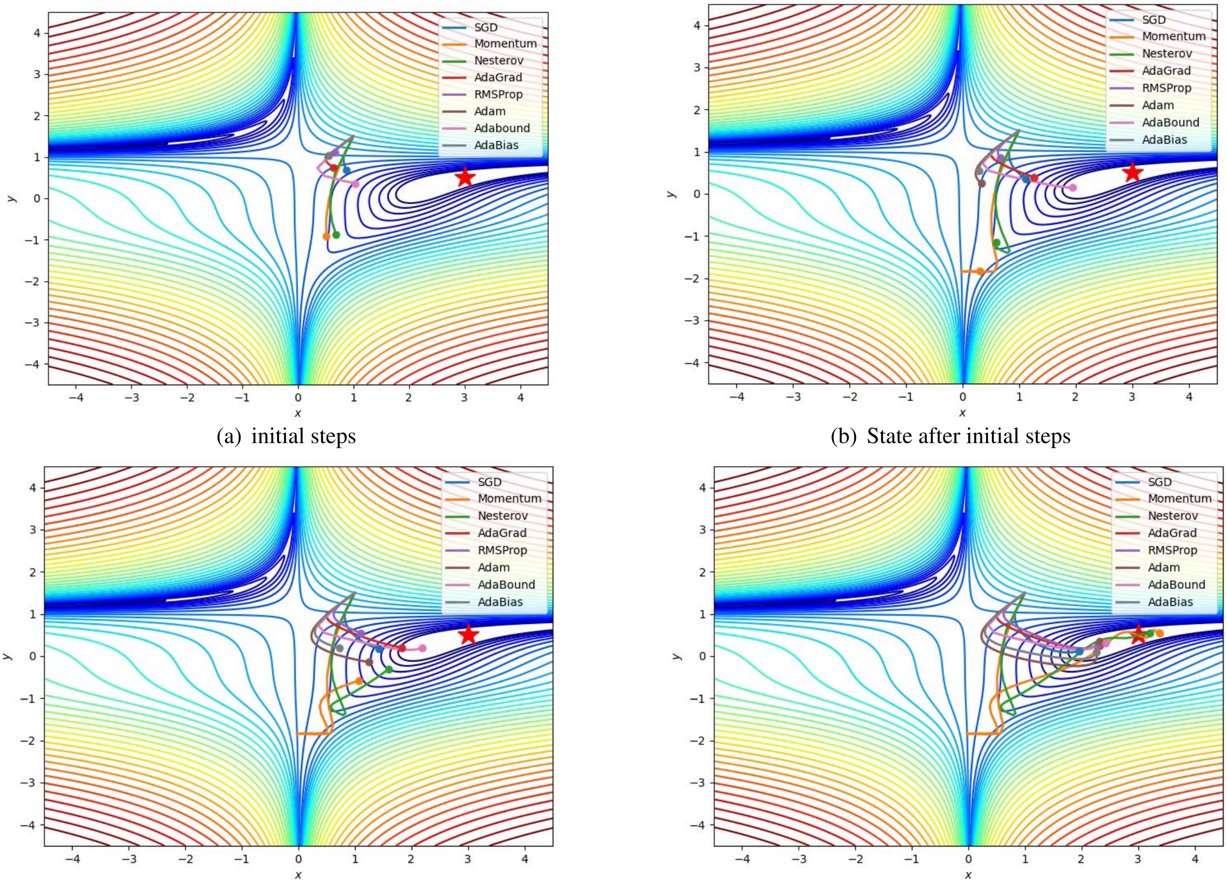
As shown in Fig. 3, we first describe the connections and differences between various existing optimization methods with RDP models. The optimization algorithms are mainly divided into momentum and adaptive algorithms. From the most classic SGD to Adam and then to various variants of Adam, AdaBound's constraint on the learning rate is an excellent new optimization idea.

III. RDP-AdaBias OPTIMIZATION

We improve the AdaBound for Rényi differential privacy protection based on bias correction. To facilitate the presentation, Table 1 summarizes the parameters. The optimization algorithm SGD samples a batch of data from the sample to perform a gradient descent, computes the gradient g_t of the loss function concerning previous parameters, and then refreshes the parameters to obtain a new θ_t, as shown in Equations (4) and (5).

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}). \tag{4}$$

$$\mu_t = \rho \tilde{g}_t. \tag{5}$$



(a) initial steps

(b) State after initial steps

(c) During operation

(d) State of steps when approaching convergence

FIGURE 2. The state of the optimization algorithm on the surface profile of the Beale function loss with a step. Gradient lines are drawn with x and y and star points are set. For example, the definition domain of the Beale function is [−4.5, 4.5] when the star point (x,y) is (3, 0.5) and the global minimum point f is 0. After the initial number of steps, AdaBound is faster and does not take unnecessary routes.

Equation (6) shows that DP-SGD makes two modifications to the standard SGD algorithm: clipping the gradient to a fixed maximum norm C and adding noise to the gradient of a given σ.

$$\tilde{g}_t = \frac{g_t}{max(1, \frac{\|g_t\|_2}{C})} + N(0, \sigma^2 I^d). \tag{6}$$

Our training optimization stage is improved by combining DP-SGD and the AdaBound algorithm. In each training step, we randomly select a prespecified number of examples. Since the size of the gradient has no a priori bound, we clip every single gradient in the ℓ_2 norm. If $\|g_t\|_2 \leq C$, then g_t will be retained, and if $\|g_t\|_2 > C$, it will reduce to norm C. We add sampled Gaussian noise to the aggregated gradient and then use the AdaBias method to trim the step size and update the gradient. Finally, we compute and return the privacy loss using the RDP. After balancing the accuracy and privacy of the model, we choose better parameters for the training and validation of the differentially private learning model.

We notice the β function mentioned in AdaBound [33], which experimented with constants and proved that $\beta_{1t} = \beta_1 \lambda^{t-1}$ (λ is a constant parameter) can be used to guarantee O($\sqrt{T}$) regret. On the other hand, Adam uses the first- and second-moment estimations m_t and v_t and applies bias corrections $\hat{m_t}$ and $\hat{v_t}$ to correct their values. To obtain a smoother momentum decay, we modify the first- and second-order moment estimation functions and turn β into a function of time t. As time increases, the numerical values of the first and second moment estimates decrease, thereby reducing the size of the iteration θ_t, as shown in Equations (7) and (8).

$$m_t = (\beta_1/\sqrt{t})m_{t-1} + (1 - \beta_1/t)\tilde{g}_t. \tag{7}$$

$$v_t = \beta_2 v_{t-1} + ((1 - \beta_2)/t)\tilde{g}_t^2. \tag{8}$$

If we use $\rho_t = \rho/\sqrt{t}$ to reduce the step size, this will cause the learning rate to decay sharply and reduce the performance of the model. Therefore, we omit the step of dividing by $\sqrt{t}$ in our algorithm and update the learning rate in the form of $\theta_{t-1} = \theta_t - \hat{\rho}_t m_t$ to speed up the convergence rate of the model. Different levels of privacy can complicate the learning process. The larger the gradient perturbation is, the more careful the design required when accessing the data, which has a privacy cost. We improve the performance of our models by saving computation time and efficiently utilizing

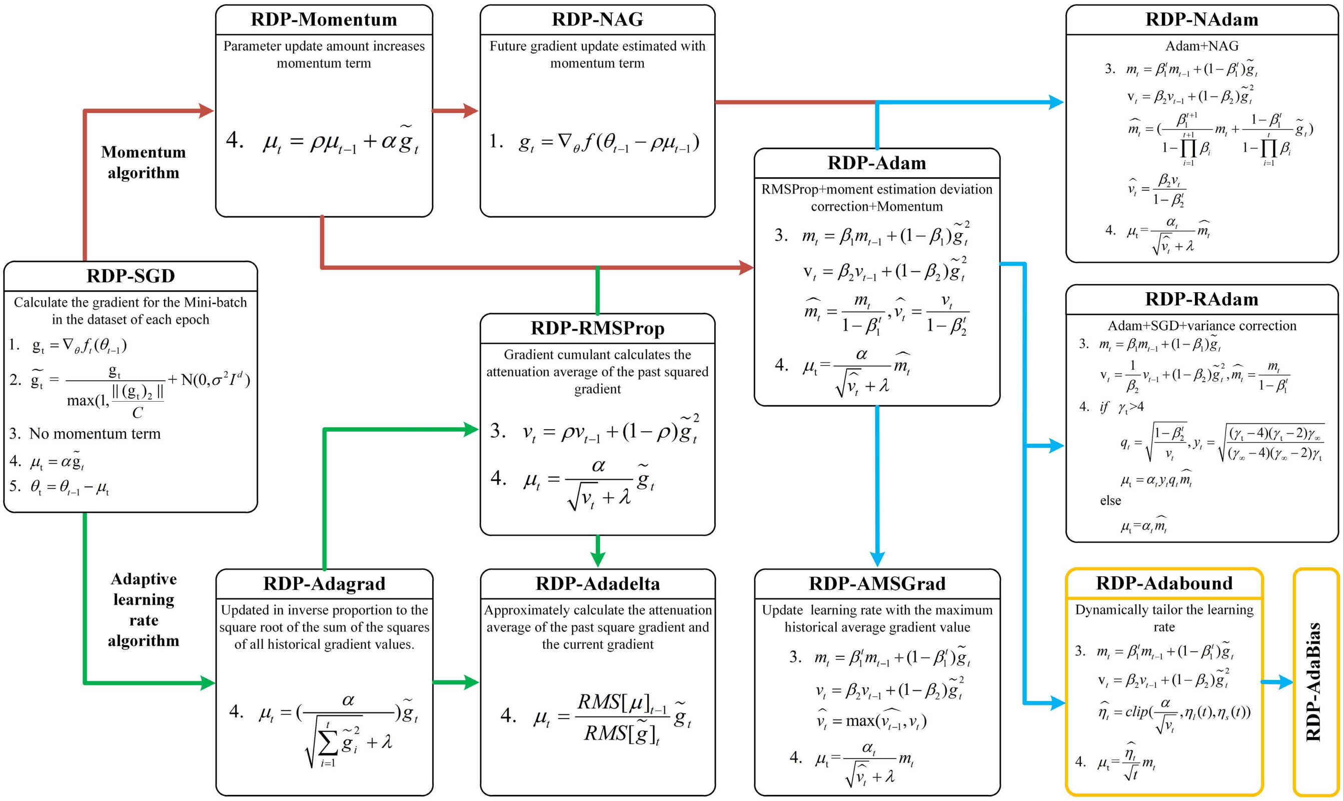

RDP-Momentum
Parameter update amount increases momentum term
4. $\mu_t = \rho\mu_{t-1} + \alpha\tilde{g}_t$
RDP-NAG
Future gradient update estimated with momentum term
1. $g_t = \nabla_\theta f(\theta_{t-1} - \rho\mu_{t-1})$
RDP-NAdam
Adam+NAG
3. $m_t = \beta_1^t m_{t-1} + (1-\beta_1^t)\tilde{g}_t$
$v_t = \beta_2 v_{t-1} + (1-\beta_2)\tilde{g}_t^2$
$\widehat{m}_t = (\frac{\beta_1^{t+1}}{1-\prod_{i=1}^{t+1}\beta_i} m_t + \frac{1-\beta_1^t}{1-\prod_{i=1}^{t}\beta_i}\tilde{g}_t)$
$\widehat{v}_t = \frac{\beta_2 v_t}{1-\beta_2^t}$
4. $\mu_t = \frac{\alpha_t}{\sqrt{\widehat{v}_t + \lambda}}\widehat{m}_t$
Momentum algorithm
RDP-Adam
RMSProp+moment estimation deviation correction+Momentum
3. $m_t = \beta_1 m_{t-1} + (1-\beta_1)\tilde{g}_t^2$
$v_t = \beta_2 v_{t-1} + (1-\beta_2)\tilde{g}_t^2$
$\widehat{m}_t = \frac{m_t}{1-\beta_1^t}, \widehat{v}_t = \frac{v_t}{1-\beta_2^t}$
4. $\mu_t = \frac{\alpha}{\sqrt{\widehat{v}_t + \lambda}}\widehat{m}_t$
RDP-SGD
Calculate the gradient for the Mini-batch in the dataset of each epoch
1. $g_t = \nabla_\theta f_t(\theta_{t-1})$
2. $\tilde{g}_t = \frac{g_t}{\max(1, \frac{\|(g_t)_2\|}{C})} + N(0, \sigma^2 I^d)$
3. No momentum term
4. $\mu_t = \alpha\tilde{g}_t$
5. $\theta_t = \theta_{t-1} - \mu_t$
RDP-RMSProp
Gradient cumulant calculates the attenuation average of the past squared gradient
3. $v_t = \rho v_{t-1} + (1-\rho)\tilde{g}_t^2$
4. $\mu_t = \frac{\alpha}{\sqrt{v_t + \lambda}}\tilde{g}_t$
RDP-RAdam
Adam+SGD+variance correction
3. $m_t = \beta_1 m_{t-1} + (1-\beta_1)\tilde{g}_t$
$v_t = \frac{1}{\beta_2}v_{t-1} + (1-\beta_2)\tilde{g}_t^2, \widehat{m}_t = \frac{m_t}{1-\beta_1^t}$
4. *if* $\gamma_t > 4$
$q_t = \sqrt{\frac{1-\beta_2^t}{v_t}}, y_t = \sqrt{\frac{(\gamma_t-4)(\gamma_t-2)\gamma_\infty}{(\gamma_\infty-4)(\gamma_\infty-2)\gamma_t}}$
$\mu_t = \alpha_t y_t q_t \widehat{m}_t$
else
$\mu_t = \alpha_t \widehat{m}_t$
Adaptive learning rate algorithm
RDP-Adagrad
Updated in inverse proportion to the square root of the sum of the squares of all historical gradient values.
4. $\mu_t = (\frac{\alpha}{\sqrt{\sum_{i=1}^{t}\tilde{g}_i^2 + \lambda}})\tilde{g}_t$
RDP-Adadelta
Approximately calculate the attenuation average of the past square gradient and the current gradient
4. $\mu_t = \frac{RMS[\mu]_{t-1}}{RMS[\tilde{g}]_t}\tilde{g}_t$
RDP-AMSGrad
Update learning rate with the maximum historical average gradient value
3. $m_t = \beta_1^t m_{t-1} + (1-\beta_1^t)\tilde{g}_t$
$v_t = \beta_2 v_{t-1} + (1-\beta_2)\tilde{g}_t^2$
$\widehat{v}_t = \max(\widehat{v_{t-1}}, v_t)$
4. $\mu_t = \frac{\alpha_t}{\sqrt{\widehat{v}_t + \lambda}} m_t$
RDP-Adabound
Dynamically tailor the learning rate
3. $m_t = \beta_1^t m_{t-1} + (1-\beta_1^t)\tilde{g}_t$
$v_t = \beta_2 v_{t-1} + (1-\beta_2)\tilde{g}_t^2$
$\widehat{\eta}_t = clip(\frac{\alpha}{\sqrt{v_t}}, \eta_l(t), \eta_s(t))$
4. $\mu_t = \frac{\widehat{\eta}_t}{\sqrt{t}} m_t$
RDP-AdaBias

FIGURE 3. Evolution diagram of the RDP-gradient descent optimization algorithm. Eleven optimization algorithms are summarized here. We use red, green, and yellow to identify the origin and development of various excellent algorithms, divide them into momentum algorithms and adaptive learning rate algorithms, and introduce the characteristics of each algorithm. For example, there are five steps in the RDP-SGD algorithm. The numbers marked in the figure are the differences between the algorithms, and the unmarked parts are the same as RDP-SGD. We use a thick yellow box to identify the algorithm that the article needs to use.

the privacy budget. We admit that dividing β by t is just a trick. The core of our method is to control the learning rate. We use bias corrections $\hat{m_t}$ and $\hat{v_t}$ to constrain the learning rate change; with increasing time, the lower bound of the learning rate increases, while the upper bound becomes smaller. Then, we multiply the learning rate's upper and lower bound functions by an additional sine function, which includes the bias correction $\hat{v_t}$, that is, $\sin(\frac{final\_lr}{\hat{v_t}})$, where $final\_lr$ is the final learning rate we manually set and is generally set to 0.1. In this way, we make the learning rate fluctuate in a cosine-like manner at the beginning to better find the best direction and speed of gradient descent and reduce unnecessary rounds.

$$\hat{m_t} = 1 - \beta_1^t, \quad \hat{v_t} = 1 - \beta_2^t. \tag{9}$$

$$\eta_l(t) = 1 - \frac{1}{\hat{m_t}t + 1}, \quad \eta_u(t) = 1 + \frac{1}{\hat{m_t}t}. \tag{10}$$

$$\hat{\rho_t} = clip(\frac{\rho}{\hat{v_t}\sqrt{v_t}}, \quad \eta_l(t), \quad \eta_u(t). \tag{11}$$

Since AdaBound does not use the method of moment estimation for bias correction, we use a new bias correction method, such as Equation (9). We combine them with the lower limit function $\eta_l(t)$ and upper limit function $\eta_u(t)$ (Equation (10)) of the learning rate. Then, we tailor the learning rate η_t and adjust the step size more reasonably to reduce the loss of the model, as shown in Equation (11).

Compared with RDP-AdaBound, which is shown in Fig. 3, RDP-AdaBias can be described by Equations (4), (6), (7), (8), (9), (10), and (11), it is described as Algorithm 1. Given an initial model *Net* with an RDP optimizer, we use Algorithm 1 to train a final privacy model *Net′* with different datasets.

IV. EXPERIMENTS

We train the Rényi differential privacy model with various neural networks, such as CNN [34], LeNet5 [35], ResNet18 [36], DenseNet121 [37] and LSTM [38]. We compare the proposed method with traditional optimization methods based on different privacy parameters. We also analyze the performance of the proposed method.

A. EXPERIMENT SETTINGS

1) DATASETS

In this paper, we evaluate our method on the MNIST [39], CIFAR10 [40], and IMDB [41] datasets. The MNIST dataset includes a train set of 60,000 samples of handwritten digital images and a test set of 10,000 samples of handwritten digital images. The handwritten digital images are 28 × 28-pixel

Algorithm 1 Improved AdaBound Optimization Method for Rényi Differential Privacy Protection

```
Input: Default settings for a initial model $Net$ are $\rho = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, the optimizer $O$.
Set $m_0 = 0$, $v_0 = 0$
for t=1 to T do
    Calculate the gradient $g_t$ at timestep $t$ using Equation (4)
    Calculate $\tilde{g}_t$ by clipping gradient $g_t$ and add sampled Gaussian noise $N(0, \sigma^2 I^d)$ using Equation (6)
    Update biased first moment estimate $m_t$ by Equation (7)
    Update biased second raw moment estimate $v_t$ by Equation (8)
    Set bias corrections $\hat{m_t}$ and $\hat{v_t}$ by Equation (9)
    Learn a new upper bound function $\eta_l(t)$ and lower bound function $\eta_u(t)$ using Equation (10)
    Tailor the learning rate $\eta_t$ of optimizer using Equation (11)
    Calculate the loss $f$ of initial model $Net$
    Back propagate $f$
    Update gradient $\theta_t$ by $\theta_{t-1} - \hat{\rho}_t m_t$
    Compute the overall privacy cost $\epsilon'(\alpha)$
end for
Output: The final trained model $Net'$
```

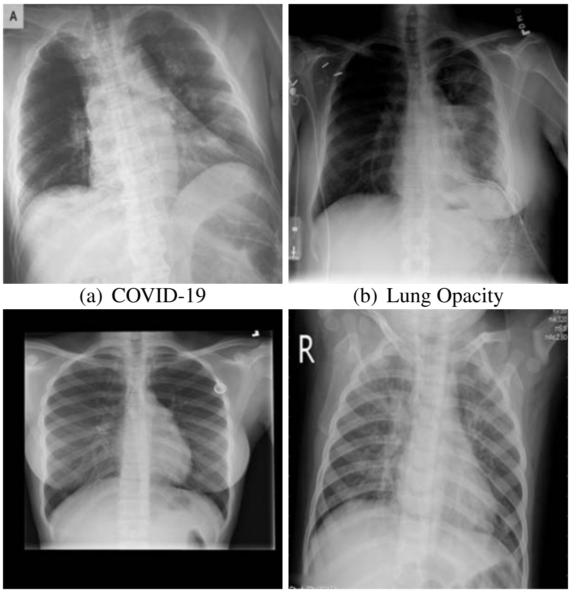

(a) COVID-19 (b) Lung Opacity (c) Normal (d) Viral Pneumonia

FIGURE 4. The examples of the COVID-19 chest x-ray dataset.

grayscale images. The CIFAR10 dataset contains ten color images of different item classifications. The train set has 50,000 samples, and the test set has 10,000 samples. The IMDB dataset contains 50,000 comments with obvious bias, of which 25,000 are used as the train set, and 25,000 are used as the test set. In addition to this, we also train and test with a dataset of Chest X-ray images of COVID-19 [26], including 10,192 normal images, 3,616 COVID-19 images, and 7,357 pneumonia images. The examples of the dataset are shown in the Fig. 4.

TABLE 2. Factors and categories affecting our experiment.

Experimental Factors	Category	Values
Datasets	3	MNIST, CIFAR10, IMDB
Networks	6	CNN1(4layer), LeNet5 ResNet18, DenseNet121 CNN2(2layer), LSTM
optimizers	5	RDP-AdaBound, RDP-AdaBias RDP-SGD(M), RDP-Adam RDP-Adagrad
noise scales σ	6	0.5, 1, 1.5, 2, 2.5, 3
probability δ	1	1e-5
momentum parameters β_1, β_2	2	0.9, 0.999
initial learnrate ρ	3	0.001, 0.01, 0.1

2) EVALUATION METRICS

The accuracy rate is the ratio of correctly predicted samples to the total number of input samples. The higher the classification accuracy is, the higher the model's utility, so we use it as the primary measurement standard. The loss function reflects the degree to which the model fits the data and illustrates the difference between the predicted and actual values. The model's generalization performance refers to the learned model's predictive ability on anonymous data, which is often related to the test loss. Robustness means that the model's efficiency changes little when the learning rate changes and is often used to measure the algorithm's stability. Finally, we use them to measure the effects of different RDP optimization models.

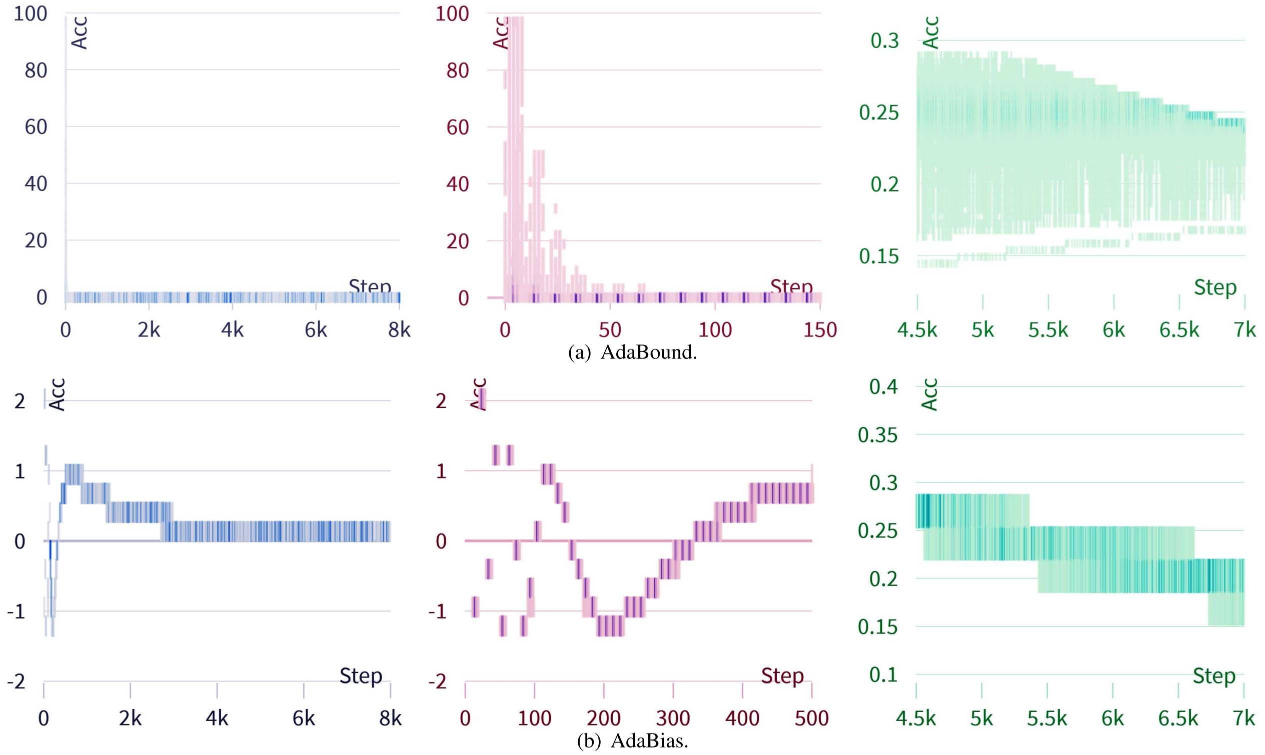
3) BASELINES

We compare our algorithm not only with RDP-AdaBound [33] but also with the traditional privacy-preserving optimization methods RDP-SGD [9], RDP-Adagrad [28] and RDP-Adam [32]. Our method can show considerable advantages by using different noise scales and gradient clipping thresholds to train the model. Since our method performs mediocrely in low noise, and its privacy model performs better in a high noise environment, we infer that our approach can better preserve model privacy.

4) SETTINGS

The experimental environment comprises the Linux 18.04 platform, 12 GB GPU memory, and Python 3.6. The code is improved based on Opacus [42]. Table 2 lists the essential parameters that affect the experimental results of differential privacy learning models and the different values.

We consider the effect of the additive Gaussian noise scale σ and the gradient clipping threshold C. Accuracy decreases as the privacy budget decreases. Table 3 also shows that noise affects the efficiency of the model. When we increase the noise scale, while the model accuracy shows an overall



(a) AdaBound.

(b) AdaBias.

FIGURE 5. When σ is 3 and ϵ is 0.26, the accuracy of the AdaBound method on the MNIST dataset is 67.88%, and the accuracy of our approach is 84.44%. The horizontal axis is the number of steps, and the vertical axis is the size of the learning rate. We performed similar experiments on the remaining models, with both learning rate changes as shown above.

TABLE 3. The impact of different noise scales σ of RDP on the MNIST dataset (Gradient clipping threshold C = 1).

Noise(σ)	Epsilon(ϵ)	CNN1	LeNet5	ResNet	DenseNet	CNN2	LSTM
0.25	400	97.44	96.87	64.06	63.06	71.18	70.78
0.5	20	96.50	95.43	60.93	56.05	69.01	63.21
0.75	5	96.48	96.47	57.03	51.86	69.62	62.00
1	2.5	96.42	96.01	53.90	46.09	68.71	61.10
1.25	2	95.98	95.86	47.65	45.23	70.12	55.04
1.5	1	94.83	93.44	41.32	38.87	65.10	51.36

TABLE 4. The effect of different gradient clipping rates *C* of RDP.

Datasets	C=0.5		C=1		C=1.5	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
MNIST	**0.9672**	0.0002	0.9671	0.0002	0.9624	0.0003
	0.9563	0.0003	**0.9594**	0.0003	0.9521	0.0005
CIFAR	**0.5131**	1.8614	0.4862	1.8948	0.4625	1.8827
	0.4732	1.8521	0.4738	1.8131	**0.4784**	1.8024
IMDB	0.6637	1.0069	0.7036	0.7866	**0.7054**	0.7493
	0.6005	1.5016	0.5761	1.2623	0.5319	1.2087

downward trend, it also fluctuates due to various factors, such as network structure, batch size, and the number of experimental rounds. Then, we train different privacy models when C is 0.5, 1, and 1.5, as shown in Table 4. The experiments defaulted to 0.001 as the learning rate. We obtained the privacy-preserving experimental parameters of different models on the MNIST, CIFAR10, and IMDB datasets and tested the model performance with varying optimization algorithms.

TABLE 5. The performance comparison on the COVID-19 dataset between RDP-Adabound and RDP-AdaBias.

Method	Loss	Accuracy	Precision	Recall
RDP-AdaBound	1.2960	0.6099	0.5169	0.2839
RDP-AdaBias	0.8530 ↓	0.6659 ↑	0.6064 ↑	0.4545 ↑

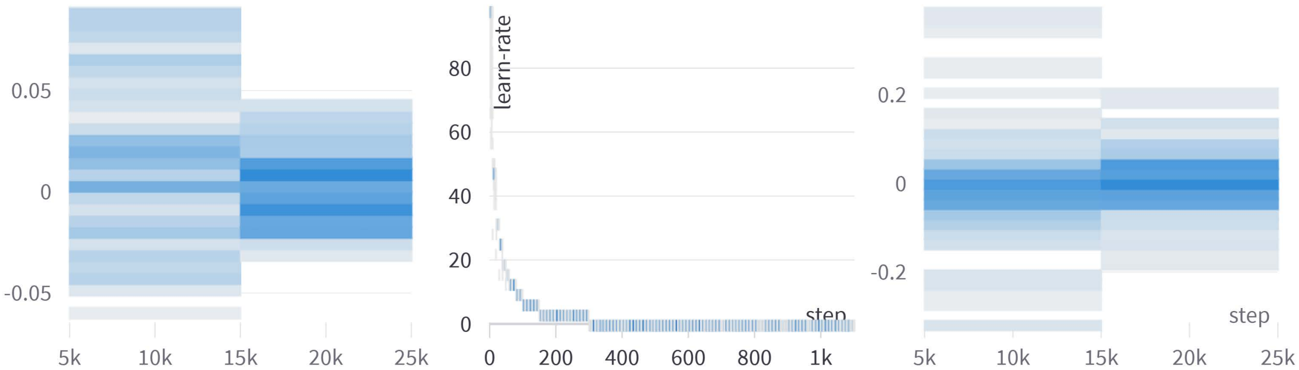
We also compare our AdaBias with Adabound on the COVID-19 dataset to verify the performance of RDP. The performance comparison result show as Table 5. From the Table 5, our AdaBias have higher accuracy, precision and recall with lower loss, which indicate that the AdaBias performs much better than Adabound. In other words, our method can indeed significantly improve the privacy-preserving efficiency on the COVID-19 dataset.

B. ANALYSIS AND DISCUSSION

The ideal learning rate algorithm searches with a large learning rate early on and then adjusts with a small learning rate. The gradient update equation of the privacy model can be simplified as "future gradient = original gradient - learning rate * current gradient." The initial learning rate of RDP-AdaBound in Fig. 5 (a) reaches above almost 100 and then decreases. When we adjusted the learning rate at equal intervals, the model accuracy was significantly improved, reaching 91.2%, as shown in Fig. 6 (b). When the noise increases, the information carried by the existing gradient

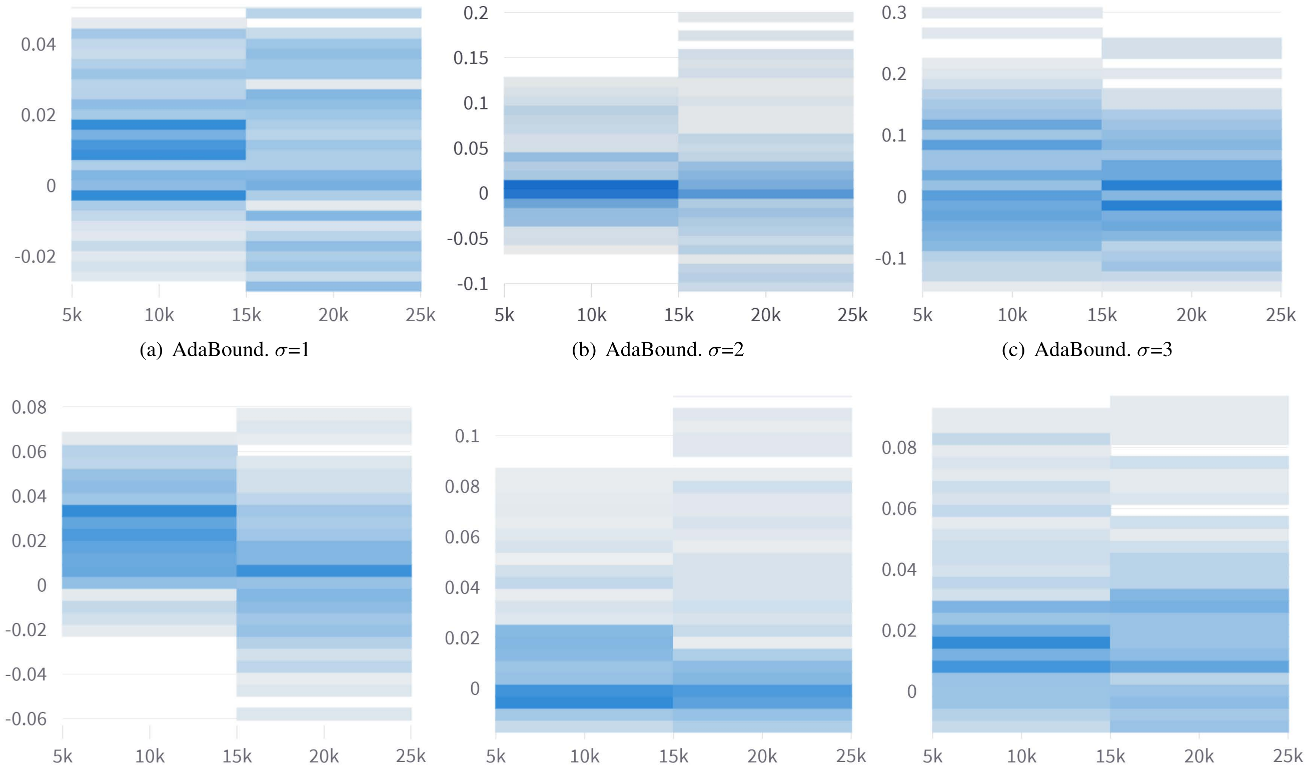
TABLE 6. The test accuracy of different optimization methods of RDP (%; mean ± std).

Datasets	Model	Traditional optimization methods			AdaBound and AdaBias	
		RDP-SGD(M)	RDP-Adagrad	RDP-Adam	RDP-AdaBound	RDP-AdaBias
MNIST	CNN1	60.39±4.01	67.82±1.75	95.02±0.17	92.82±0.03	**94.52±0.18**
	LeNet5	76.71±4.29	67.46±3.86	93.43±0.09	92.72±0.18	**95.17±0.39**
CIFAR10	ResNet18	36.91±0.36	31.15±0.25	41.36±0.85	43.93±0.21	**44.65±0.25**
	DenseNet121	31.87±0.27	26.77±0.52	35.15±0.22	34.73±0.47	**39.98±0.25**
IMDB	CNN2	56.04±0.25	50.74±0.69	59.73±0.83	64.27±0.97	**67.07±0.76**
	LSTM	51.05±0.00	50.33±0.41	52.38%±0.14	52.41±0.29	**53.51±0.27**



(a) Model gradients without privacy. (b) Learning rate changes under privacy. (c) Privacy-preserving model gradients.

FIGURE 6. (a) is the gradient change under the noiseless model. (b) and (c) are the changes in the learning rate and first layer gradient weights of RDP-AdaBound after using the equally spaced learning rate descent strategy. At this time, the model's accuracy can reach 91.2% at the beginning, but it will not increase in the later period and stabilize at approximately 91.4%.

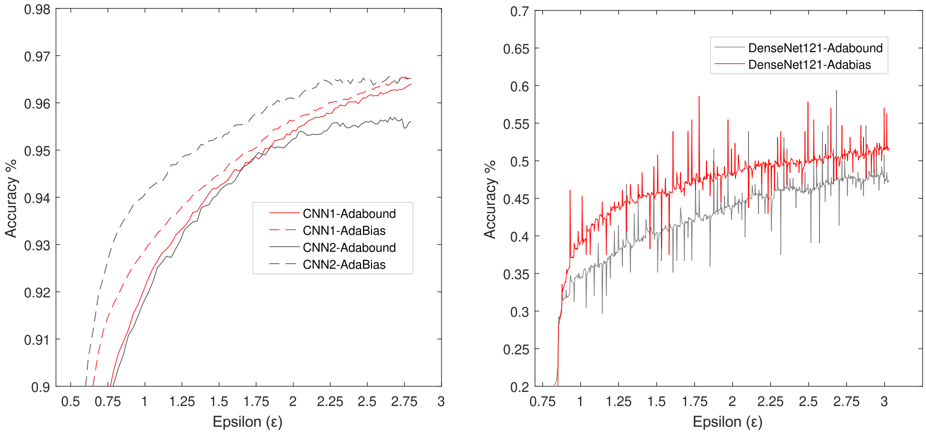


(a) AdaBound. σ=1 (b) AdaBound. σ=2 (c) AdaBound. σ=3

(d) AdaBias. σ=1 (e) AdaBias. σ=2 (f) AdaBias. σ=3

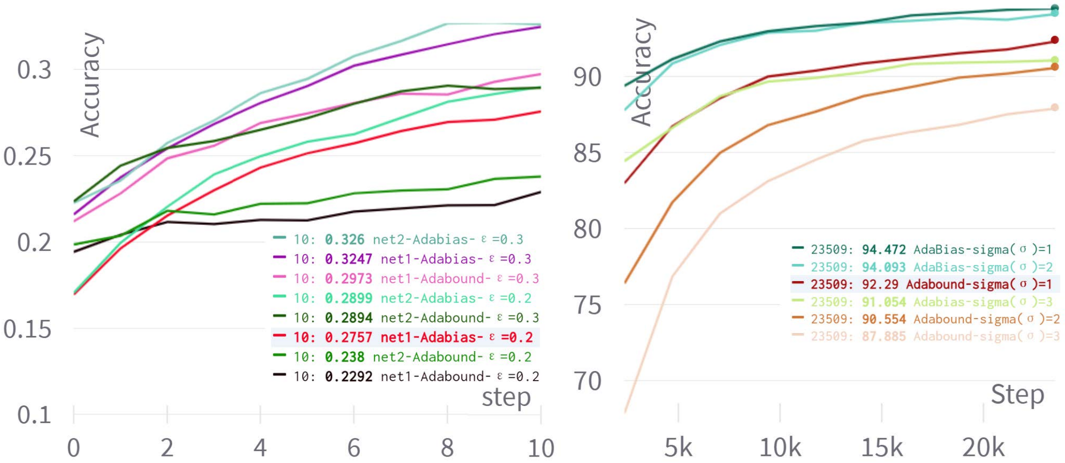
FIGURE 7. The changes in model gradient weights using RDP-AdaBias and RDP-AdaBound on the MNIST dataset as the noise scale σ increases.

increases, and an excessively high learning rate will affect the gradient update and reduce the model's performance. Therefore, we bias-correct the upper and lower bound functions of the learning rate so that the learning rate starts



(a) MNIST. (b) CIFAR10.

FIGURE 8. The relationship between epsilon ϵ and train accuracy of the two models on MNIST and DenseNet121 on CIFAR10.



(a) CIFAR10. when ϵ is 0.2 and 0.3. (b) MNIST. when σ is 1, 2, and 3.

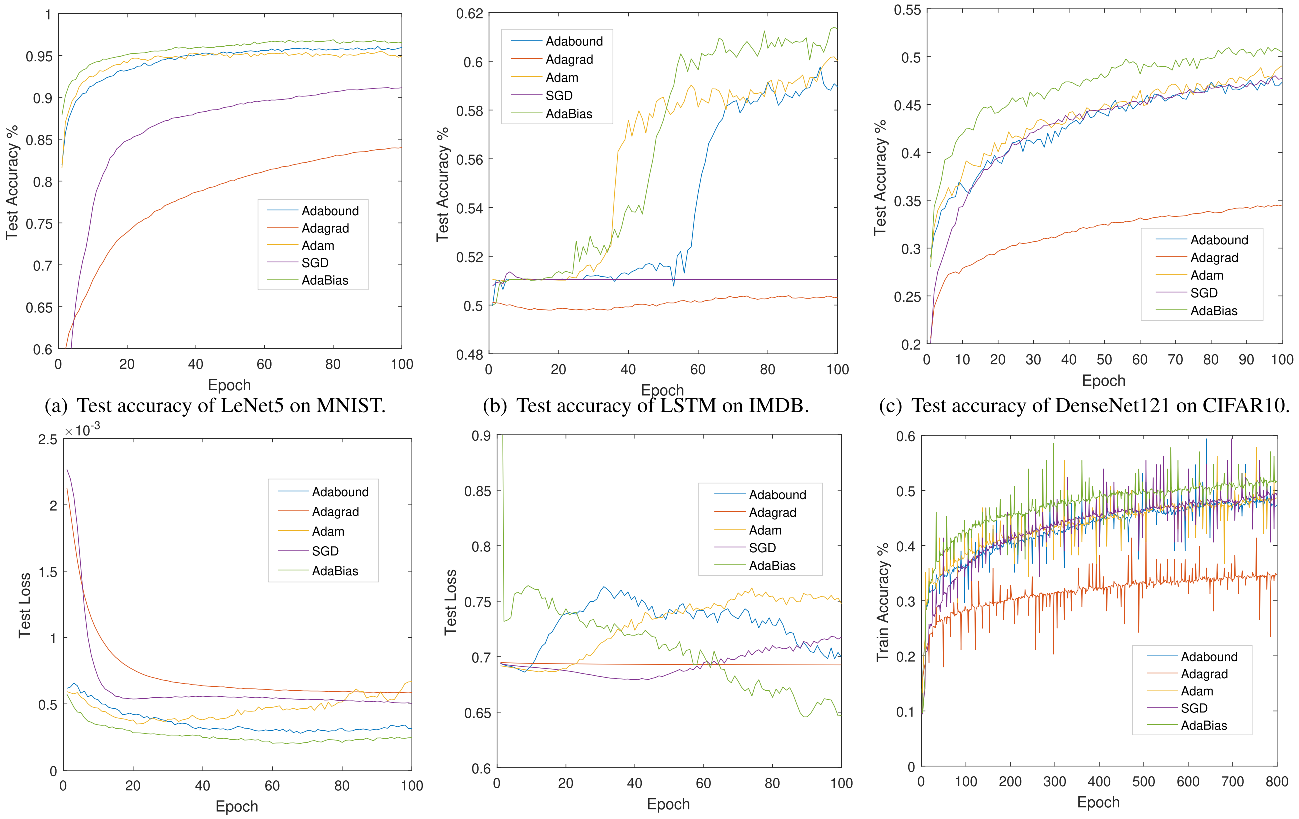
FIGURE 9. Analysis and Discussion with privacy budget ϵ and noise scale σ.

with a cosine-like fluctuation and then decreases slowly in a step-like fashion, as shown in the three-stage changes of the learning rate of Fig. 5 (b).

We experimented with the weights of the privacy-free model and compared them with the weights in the noisy case, as seen in Fig. 6 (a)(c). As the noise increases, the gradient weight range updated by the AdaBias method increases, which can effectively adapt to the influence of noise. In the same situation, as shown in Fig. 7, the weight of AdaBound does not change significantly with the increase in noise. The learning rate and clipping norm have no a priori bounds, and the hyperparameter grid generated during optimization increases privacy costs. In this way, we assign the value of the learning rate more reasonably, improving the privacy model's performance.

As the privacy budget ϵ increases, the cost of privacy recovery increases, which leads to an increase in accuracy but a decrease in privacy. Only some of the results are shown here for illustration. We denote our proposed method RDP-AdaBias with a dashed line in Fig. 8 (a), which can achieve higher accuracy under the same privacy budget. That is, our differentially private learning model has better privacy, and (b) also confirms this. From Fig. 9, we observe that on the MNIST or CIFAR10 dataset, when the noise increases, σ is set to 3 and ϵ is 0.2, which is higher than the model accuracy when sigma is 2 and ϵ is 0.3. The degree of privacy protection increases, but the difference in accuracy between our method and the original method increases.

Table 6 shows the difference in mean accuracy of our model with that of the original method RDP-AdaBound under different privacy budgets. AdaBias can maintain higher accuracy and better protect privacy data when the privacy budget is small, with a maximum increase of 5.17%. Table 7 is the final result of model testing with different optimization algorithms on the RDP model. With such a high privacy budget, the improvement effect of our method is also acceptable compared to the original form and the traditional method.



(a) Test accuracy of LeNet5 on MNIST.

(b) Test accuracy of LSTM on IMDB.

(c) Test accuracy of DenseNet121 on CIFAR10.

(d) Test loss of LeNet5 on MNIST.

(e) Test loss of CNN on IMDB.

(f) Train accuracy of DenseNet121 on CIFAR10.

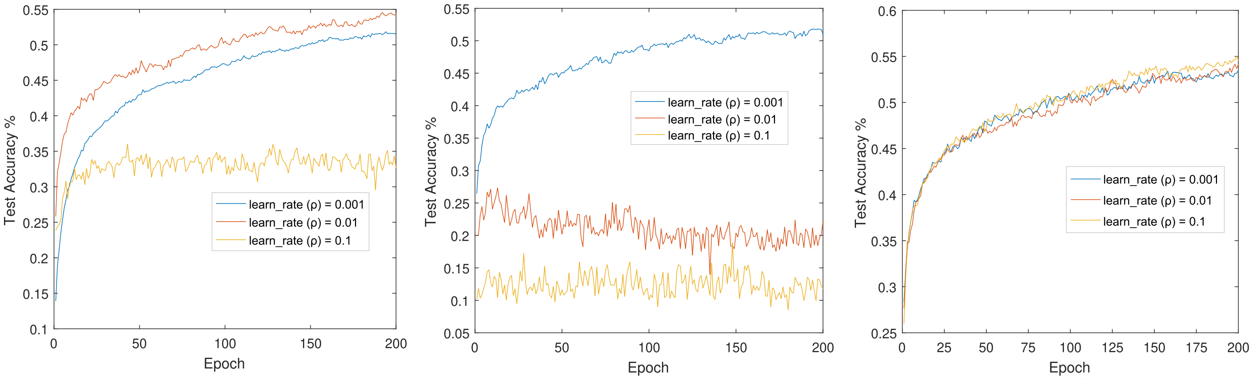
FIGURE 10. (a)(b)(d)(e) are accuracy and loss of test on the MNIST and IMDB datasets; (c)(f) are train and test accuracy on the CIFAR10 dataset of DenseNet121.

TABLE 7. Mean test accuracy of different models corresponding to different ϵ values of AdaBound and AdaBias (3 runs).

Epsilon(ϵ)	Models	RDP-AdaBound	RDP-AdaBias	Gap
0.9	MNIST(LeNet5)	92.06	94.34	↑ 2.28
	CIFAR(DenseNet)	31.90	35.80	↑ 3.90
	IMDB(CNN)	62.90	66.01	↑ 3.10
1.2	MNIST(LeNet5)	93.72	95.28	↑ 1.56
	CIFAR(DenseNet)	37.66	42.83	↑ 5.17
	IMDB(CNN)	66.59	68.80	↑ 2.21
1.5	MNIST(LeNet5)	94.73	95.93	↑ 1.20
	CIFAR(DenseNet)	41.01	45.22	↑ 4.21
	IMDB(CNN)	69.41	70.83	↑ 1.42

Fig. 10 (a)(b) and (d)(e) depicts the variation in accuracy and loss for model testing on different MNIST and IMDB datasets. It can be seen from the figure that compared with RDP-AdaBound, our method has a faster improvement in accuracy and a smoother convergence trend. The lower the test loss is, the better the model's generalization performance. Our approach has shown promising results in all three situations. Although the final effect of LSTM loss on IMDB is not ideal, it reaches the end earlier and is more stable than the original method. As shown by the orange lines in Fig. 10 (b)(e), compared to our green lines, due to the complexity of the experimental network model environment and the applicability of different optimization algorithms, even if the parameters are the same, the RDP-Adagrad algorithm does not converge in the first 100 epochs but does not converge until 200 periods. Even so, our method achieves better results than other methods in the presence of privacy protection. It can reach the end point earlier and is more stable than the original method. Furthermore, the privacy budget of the privacy-preserving gradient optimization method is inversely proportional to the square of the number of steps. The faster the convergence, the higher the privacy guarantee, which also proves the superiority of RDP-AdaBais.

Fig. 10 (c)(f) shows the train and test accuracy on the CIFAR10 dataset. Due to the limitation of image size, we compress the ups and downs of the data in Fig. 10 (f) to a minimum. As a result, there is a histogram-like vertical line effect on each epoch, but an upward trend in accuracy can be seen. Compared with the original method, the accuracy of our approach is greatly improved. AdaBound's article points out that in complex models such as ResNet, the learning rate is frequently too large or too small, affecting the final convergence result. So we verify the robustness of the proposed algorithm concerning the learning rate. We selected three common initial learning rates of 0.1, 0.01, and 0.001 for training. As shown in Fig. 11, the accuracy of the RDP-AdaBias model changes less when the rest of the settings are the same except for the learning rate. Therefore, the results are more robust than RDP-SGD and RDP-Adam.



(a) SGD(M). (b) Adam. (c) AdaBias.

FIGURE 11. The accuracy of RDP-SGD(M), RDP-Adam and RDP-AdaBias at different learning rate learning rateρ in ResNet18 on the CIFAR10 dataset.

V. CONCLUSION

Rényi differential privacy plays an essential role in protecting data privacy and models in social networks, and its application to deep learning will become a popular trend. In this paper, we have improved the RDP-AdaBound method. To further illustrate the performance of this method, we also extended other RDP optimization methods to compare and analyze them. Experimental evaluation and analysis verified the effectiveness of our proposed modification method. Our method is suitable for model environments with high noise and can achieve higher accuracy and lower loss than the original method, effectively improving the privacy protection efficiency of deep learning models.

Machine learning and privacy research are not necessarily a zero-sum game between utility and privacy as they have similar goals. Since our method performs modestly on privacy models at low noise scales, we will focus on improving model performance in this area in the next step.

REFERENCES

[1] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.* Berlin, Germany: Springer, 2008, pp. 1–19.

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

[3] I. Mironov, K. Talwar, and L. Zhang, "Rényi differential privacy of the sampled Gaussian mechanism," 2019, *arXiv:1908.10530*.

[4] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*.

[5] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 1310–1321.

[6] M. Abadi, A. Chu, and I. Goodfellow, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[7] B. Zhao, K. R. Mopuri, and H. Bilen, "IDLG: Improved deep leakage from gradients," 2020, *arXiv:2001.02610*.

[8] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, p. 32.

[9] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[10] Z. Bu, J. Dong, Q. Long, and S. Weijie, "Deep learning with Gaussian differential privacy," *Harvard Data Sci. Rev.*, vol. 23, Jul. 2020.

[11] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 245–248.

[12] R. Bassily, V. Feldman, C. Guzmán, and K. Talwar, "Stability of stochastic gradient descent on nonsmooth convex losses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4381–4391.

[13] 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.

[14] Y. Liu, M. Zhang, Z. Zhong, and X. Zeng, "A novel adaptive cubic quasi-newton optimizer for deep learning based medical image analysis tasks, validated on detection of COVID-19 and segmentation for COVID-19 lung infection, liver tumor, and optic disc/cup," *Med. Phys.*, Sep. 2022.

[15] Y. Zhou, K. Huang, C. Cheng, X. Wang, and X. Liu, "LightAdam: Towards a fast and accurate adaptive momentum online algorithm," *Cognit. Comput.*, vol. 14, no. 2, pp. 764–779, Mar. 2022.

[16] Z. Yu, G. Sun, and J. Lv, "A fractional-order momentum optimization approach of deep neural networks," *Neural Comput. Appl.*, vol. 34, no. 9, pp. 7091–7111, May 2022.

[17] R. Jie, J. Gao, A. Vasnev, and M.-N. Tran, "Adaptive hierarchical hyper-gradient descent," *Int. J. Mach. Learn. Cybern.*, pp. 1–21, Aug. 2022.

[18] O. Frisk, F. Dormann, C. M. Lillelund, and C. F. Pedersen, "Super-convergence and differential privacy: Training faster with better privacy guarantees," in *Proc. 55th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2021, pp. 1–6.

[19] Y. Zhou, X. Chen, M. Hong, Z. Steven Wu, and A. Banerjee, "Private stochastic non-convex optimization: Adaptive algorithms and tighter generalization bounds," 2020, *arXiv:2006.13501*.

[20] Q. Wu, M. Li, J. Zhu, R. Zheng, L. Xing, and M. Zhang, "DP-RBAdaBound: A differentially private randomized block-coordinate adaptive gradient algorithm for training deep neural networks," *Exp. Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118574.

[21] A. Koskela and A. Honkela, "Learning rate adaptation for differentially private learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2465–2475.

[22] B. Wang, Q. Gu, M. Boedihardjo, F. Barekat, and S. J. Osher, "DP-LSSGD: A stochastic optimization method to lift the utility in privacy-preserving ERM," 2019, *arXiv:1906.12056*.

[23] J. Zhang, K. Zheng, W. Mou, and L. Wang, "Renyi differentially private erm for smooth objectives," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2037–2046.

[24] R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi, "Large-scale differentially private BERT," 2021, *arXiv:2108.01624*.

[25] Y. X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled rényi differential privacy and analytical moments accountant," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1226–1235.

[26] A. M. Tahir, M. E. H. Chowdhury, A. Khandakar, T. Rahman, Y. Qiblawey, U. Khurshid, S. Kiranyaz, N. Ibtehaz, M. S. Rahman, S. Al-Maadeed, S. Mahmud, M. Ezeddin, K. Hameed, and T. Hamid, "COVID-19 infection localization and severity grading from chest X-ray images," *Comput. Biol. Med.*, vol. 139, Dec. 2021, Art. no. 105002.

[27] 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.* Berlin, Germany: Springer, 2006, pp. 486–503.

[28] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 1–39, 2011.

[29] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6A overview of mini-batch gradient descent," vol. 14, no. 8, p. 2, 2012.

[30] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999.

[31] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence O $(1/^2)$," *Doklady USSR*, vol. 269, pp. 543–547, 1983.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[33] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," 2019, *arXiv:1902.09843*.

[34] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[35] Y. LeCun "LeNet-5, convolutional neural networks," vol. 20, no. 5, p. 14, 2015. [Online]. Available: http://yann.lecun.com/exdb/lenet

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[37] G. Huang, Z. Liu, and L. Van Der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 4700–4708.

[38] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks*, 2012, pp. 37–45.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[40] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[41] K. Dodds, "Popular geopolitics and audience dispositions: James bond and the internet movie database (IMDb)," *Trans. Inst. Brit. Geographers*, vol. 31, no. 2, pp. 116–130, Jun. 2006.

[42] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov, "Opacus: User-friendly differential privacy library in PyTorch," 2021, *arXiv:2109.12298*.

XUANYU ZHAO is currently a Graduate Student with the College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi, China. Her current research interests include image processing and privacy protection.

TAO HU (Member, IEEE) received the B.S. degree in software engineering from the School of Computer, Wuhan University of Technology, Wuhan, in 2006, the M.S. degree in software engineering from the School of Software and Microelectronics, Peking University, Beijing, in 2009, and the Ph.D. degree in computer science from the School of Computer science, Wuhan University, Wuhan, in 2020. He currently works as an Associate Professor with the College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi, China. His current research interests include deep learning, computer animation, and image processing.

JUN LI received the bachelor's degree in mathematics from Hubei University for Nationalities, in 1995, and the master's degree in computer application from the Huazhong University of Science and Technology, in 1999. He currently works as a Professor with the College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi, China. His research interests include computer graphics and image processing, virtual reality technology, and digital protection of intangible cultural heritage.

CHUNXIA MAO is currently a Graduate Student with the College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi, China. Her current research interests include information hiding and privacy protection.