IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 September 2022, accepted 9 October 2022, date of publication 17 October 2022, date of current version 25 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3215265

RESEARCH ARTICLE

Research on Improvement of the Click-Through Rate Prediction Model Based on Differential Privacy

LEI TIAN[1,2], LINA GE[2,3,4], ZHE WANG[2,3,4], GUIFEN ZHANG[2,3], CHENYANG XU[2,3], AND XIA QIN[2,3]

[1]School of Electronic Information, Guangxi Minzu University, Nanning 530006, China
[2]Key Laboratory of Network Communication Engineering, Guangxi Minzu University, Nanning 530006, China
[3]School of Artificial Intelligence, Guangxi Minzu University, Nanning 530006, China
[4]Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning 530006, China

Corresponding author: Lina Ge (66436539@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61862007, and in part by the Guangxi Natural Science Foundation under Grant 2020GXNSFBA297103.

ABSTRACT Click-through rate prediction is crucial in network applications such as recommendation systems and online networks. Designing feature extraction schemes to obtain features and modeling users' click behavior are used to estimate the probability of users clicking on recommended items. The AutoInt model is a recent and effective research finding. It constructs combined features by referencing the multi-head attention mechanism but does not fully mine meaningful high-order cross-features and ignores user privacy protection. To address this problem, this study proposes the differential privacy bidirectional long short-term memory network (DP-Bi-LSTM-AutoInt) model, which is an improved AutoInt model. A bidirectional long short-term memory network is added after the embedding layer to deeply mine the nonlinear relationship between user click behaviors and construct high-order features. Further, differential privacy technology is adopted for user privacy protection, and the Gaussian mechanism is used to randomly perturb the gradient descent algorithm of the model. Using the Criteo dataset to conduct experiments, the experimental results show that the accuracy of the Bi-LSTM-AutoInt model proposed herein is improved by 0.65 % compared to the original AutoInt model. When the privacy budget is greater than 3.0, the accuracies of the DP-Bi-LSTM-AutoInt and Bi-LSTM-AutoInt models are nearly equivalent. However, the DP-Bi-LSTM-AutoInt model algorithm is more secure and reliable than the AutoInt model.

INDEX TERMS Click-through rate prediction, differential privacy, gradient descent, long short-term memory.

I. INTRODUCTION

In the context of the big data environment, the daily Internet browsing data generated by users grows exponentially. Internet companies can accurately push personalized recommended advertisements and content for users based on these data, resulting in increased company benefits [1]. However, some browsing data include users' personal data such as hospital medical records, census records, home addresses, and consumption records. If unprotected, it can pose a threat to the user's privacy [2]. Therefore, making better use of user-available browsing data and ensuring user privacy protection becomes a concern that needs to be addressed.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed.

Many companies use a recommendation system as a key technology. The click-through rate (CTR) prediction model is an important component of such a recommendation system. It estimates the probability of a user clicking on a recommended item [3]. In CTR prediction, mining the interaction between features and extracting user interest features are the key factors affecting the prediction rate [4]. Currently, CTR prediction research is mainly divided into two categories—shallow models based on traditional machine learning and models based on deep learning neural networks. The first category includes the logistic regression (LR) [5], factorization machine (FM) [6], gradient boosting tree [7], and other linear models. The LR model is easy to implement and features strong interpretability; however, it cannot extract combined information through feature interaction. With the

achievements of deep learning in other fields, such as images and computer vision; the current focus in the field of CTR prediction models is biased toward finding more diverse feature crossover methods based on different deep learning results [8], and applying deep learning neural networks to CTR prediction. The Wide&Deep model [9] proposes a hierarchical learning structure that combines LR and deep learning technology using the memory ability of linear models to learn shallow features and the generalization ability of deep neural networks to learn high-level features. To better learn the ability of shallow network interaction, the DeepFM model [10] replaces the Wide part in the Wide&Deep model with the FM structure. The Attentional Factorization Machine(AFM) model [11] is the first to use a neural attention network to combine the attention mechanism with FM to learn the importance of each feature interaction. The AutoInt model [12] uses the multi-head attention mechanism to construct combined features and estimates the probability of users clicking on recommended items by designing a feature extraction scheme to obtain features and modeling user click behavior.

However, the efficient development of CTR prediction model technology has data privacy protection issues. The accuracy of recommendation technology often requires a huge amount of user data as support. However, the click record of a user is private information, and attackers can exploit the overfitting defect of the algorithm to reproduce the data trained by the model through gradient descent technology and confidence [13], thus raising a serious privacy concern [14]. In 2018, Cambridge Analytica, a data analysis company in the United States, had a privacy breach. The company secretly leaked the personal information of nearly 50 million Facebook users, triggering strong condemnation from users [15]. Machine learning services such as Google Prediction API and Amazon Machine Learning can leak membership information from purchase records [16]. Therefore, user data privacy research is a critical concern that cannot be avoided in the development of CTR prediction models.

At present, in the field of privacy protection, differential privacy technology [17] is widely used. It provides a mathematical definition of privacy and a provable privacy guarantee for each record in a dataset, which is suitable for the privacy protection requirements of recommendation systems. Since McSherry et al. [18] first introduced differential privacy to the recommendation system and proved its effectiveness, many scholars have proposed their own differential privacy recommendation algorithms. Ren et al. [19] proposed a recommendation model based on autoencoders and differential privacy, and they designed two methods to apply differential privacy to autoencoders: input perturbation and objective function perturbation. This protects the privacy of user data while ensuring the accuracy of recommendations. Zhang et al. [20] designed an output perturbation method to achieve matrix factorization; however, this algorithm is prone to bottlenecks in time performance in case of large amounts of data. Zhu et al. [21] pointed out that adding noise to the dataset is the most straightforward and effective approach; however, this approach affects the utility of the learned model because it heavily relies on attribute values in the training dataset. While adding noise during training, the model can be corrected considering the noise. By contrast, adding noise to the loss function or gradient only slightly affects the utility of the learned model. Adding noise to the loss function or gradient is resistant to membership inference attacks [22], which can be guaranteed by the property of differential privacy.

AutoInt model [12] is one of the more effective research results at present. This model constructs combined features by citing a multi-head attention mechanism; however, it does not fully mine meaningful high-order cross-features and only finds features with similar relationships to combine. The mining of potential features is insufficient, and the protection of users' privacy is ignored. Further, the differential privacy technology based on deep learning is not yet mature. Since gradient disturbance adds noise to the gradient during each iteration of the training process, the noise continues to accumulate, which may affect the final utility of the model.

Based on the improvement in the AutoInt model, this study proposes a CTR prediction model based on differential privacy technology to deal with nonlinear associations between features, capture the dynamic evolution of user interests, and enhance the privacy protection of the model. First, this study adds a bidirectional long short-term memory network (Bi-LSTM) before the interaction layer in the AutoInt model to deeply mine the nonlinear relationship between user click behavior and construct high-order features. Second, differential privacy is applied to the CTR prediction model, and the user's sensitive information is prevented from leaking during the recommendation process by adding Gaussian noise to the gradient of each step in the model training process.

The contributions of this paper can be summarized as follows:

- This study improves on the original AutoInt model and adds a Bi-LSTM network before the multi-head attention network layer to improve the ability of feature intersection and explore the nonlinear relationship between features.
- This study applies differential privacy technology to the CTR prediction model and protects the privacy of the model by adding Gaussian noise to the gradient of each step.
- A large number of experimental results show that the algorithm guarantees the user's privacy on the premise of ensuring the accuracy of the model.

The following is the structure of this paper: the second part is related work; the third part is the detailed description of the algorithm proposed in this study; the fourth part is the experimental analysis and comparison; and, the last part is the summary and outlook.

II. RELATED WORK

A. EQUATIONS

Differential privacy (DP) has a rigorous mathematical framework for evaluating and protecting data privacy. The model mainly achieves ensures privacy protection by adding

appropriate noise to the query or analysis results. The privacy protection is on data level. Intuitively, DP makes sure that after introducing noise, true data is protected from attacks. The basic definitions and properties of DP involved in this study are as follows:

Definition 1 [17]: Denote the dataset as D, and the model parameters of deep learning are θ. The training database consisting of D subset of all a is denoted as D'. The parameter space is T. A stochastic deep learning training mechanism takes the training dataset as input and uses a gradient descent algorithm for training. The parameters after output training are recorded as $M : D' \to T$. We call this training mechanism satisfying $(\varepsilon, \delta) - DP$. If any two adjacent training sets $d, d' \in D$, and any parameter ranges $S \subset T$, the parameter distribution of its output satisfies (1):

$$\Pr[M(d) \in S] \leq e^{\varepsilon} \Pr\left[M(d') \in S\right] + \delta \tag{1}$$

Definition 2 (Gaussian Mechanism) [23]: For function $f{:}D \to T$ on the dataset D, Sensitivity is represented by (2):

$$\Delta_2(f) = \max_{d,d' \in D} \left\| f(d) - f\left(d'\right) \right\|_2 \tag{2}$$

The Gaussian mechanism adds noise that is sampled from a zero-mean isotropic Gaussian distribution. Then, for any $\delta \in (0, 1)$, the given random noise follows a normal distribution $N\left(0, \sigma^2\right)$, then random algorithm $M(d) = f(d) + N\left(0, \sigma^2\right)$ obey $(\varepsilon, \delta) - RDP$, in $\varepsilon \geq \left(\sqrt{2\ln(1.25/\delta)}\right) / (\sigma / \Delta_2 f)$.

Where sensitivity determines the noise required for a particular query in the mechanism. It is only relevant for the query type. Sensitivity reflects the maximum range of a query function when queried on two datasets D and D' that differ by only one individual. It is independent of the data set and is determined only by the query function itself. DP has different implementation mechanisms for different algorithms. The Laplace and Gaussian mechanisms are typically used to protect numerical results, while the exponential mechanism is suitable for non-numerical results.

Property 1 (Sequence Compositionality [24]): set multiple random algorithms $A_1, A_2, \cdots, A_n$. The privacy budget corresponding to each random algorithm is $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$, and each satisfies $\varepsilon-$DP. Then, the combined algorithm A composed of these algorithms satisfies $\varepsilon-$DP for the same datasets.

Property 2 (Parallel Compositionality [24]): set multiple random algorithms $A_1, A_2, \cdots, A_n$. The privacy budget corresponding to each random algorithm is $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$, and each satisfies $\varepsilon-$DP. Then, the combined algorithm A composed of these algorithms satisfies $\varepsilon-$DP for disjoint datasets.

These two properties play an important role in proving whether the relevant algorithm satisfies the DP process.

B. LONG SHORT-TERM MEMORY

LSTM is a recurrent neural network (RNN) variant [25]. Although RNNs can theoretically solve long-distance dependencies, problems such as gradient disappearance and explosion make this difficult to achieve. In this regard, LSTM provides a solution by introducing a gate mechanism and memory unit, that is, designing input gates, output gates, and forget gates inside each neuron. Benefiting from memory and forgetting mechanisms, LSTMs can automatically update themselves to determine the amount of information that must be forgotten and remembered at each time step. Therefore, LSTM can obtain discriminative information and learn the dependencies in the features as well as mine latent features. Khan et al. [26] used a multi-layer Bi-LSTM (Mbi-LSTM) convolutional neural network (CNN) to optimize the extraction of learned features through various convolution and pooling layers and passed the features to MBD-LSTM for the classification study. Hou et al. [27] adopted the parallel structure of a CNN and Bi-LSTM with a self-attention mechanism for dataset entity mining, which has good cross-domain learning and recognition capabilities. Du et al. [28] used an attention-based Bi-LSTM to model the sequential dependencies of entities and relationships in each connection path, ultimately generating recommendation results and explanations.

Recently, scholars have also begun using the LSTM model [29] in the CTR prediction problem and have obtained better accuracy. Li et al. [30] proposed a CTR prediction model based on attention mechanism and LSTM and applied it to the Weibo service. The R-RNN model [31] not only applies an attention mechanism to help capture the representation of the user's main interests but also incorporates an LSTM unit for exploring the interest-changing trends behind the user's recent click behavior. Wang et al. [32] used a Bi-LSTM network to model dependencies between actions for capturing the importance of underlying user interests behind user behavior data, which can effectively learn functional interactions. The DSIN model [33] simulates the user behavior that is closely related to the session. First, the user's historical click behavior is divided into different sessions. Then, to better complete the CTR prediction, use Transformer to learn each session to obtain the interest vector and Bi-LSTM to learn the user's interest changes across multiple sessions.

C. AutoInt MODEL

As shown in Fig. 1, the AutoInt model [12] learns the interaction information between shallow and deep features by constructing high-order features using a multi-head self-attention mechanism [34]. It is mainly divided into four parts: input layer, embedding layer, interaction layer, and output layer.

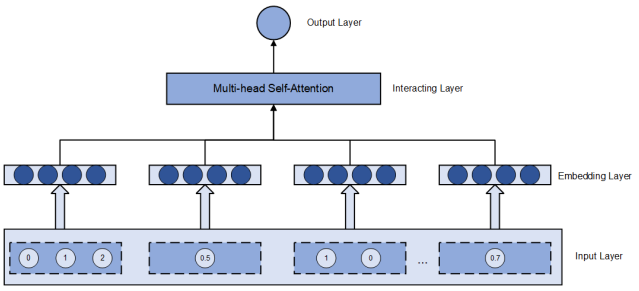


FIGURE 1. AutoInt model.

1) INPUT LAYER

X represents the feature set, $X = [x_1, \ldots, x_n], x_i \in R^n$, n is the total number of feature fields. x_i is the feature representation of the i field. For numeric features, x_j is the scalar of the eigenvalues of the j numeric domain. For categorical features, one hot encoding is required.

2) EMBEDDING LAYER

For categorical features, the low-dimensional dense vector is represented as $e_i = V_i x_i$, where V_i is the embedding matrix corresponding to the feature group i. x_i is a one hot vector, $e_i = V_i x_i / q$, where q represents the number of the i feature group values of a sample. For numerical features, the continuous features are expressed as $e_m = v_m x_m$, where v_m is the embedding matrix corresponding to the feature group m. x_m is a scalar value.

3) INTERACTION LAYER

The model employs a multi-head attention network to process the output of the embedding layer and a key-value attention mechanism to determine which feature combinations are meaningful. For example, for features e_m, first define the correlation k between it and the feature under a specific attention head h. The specific equations are (3) and (4).

$$\psi^{(h)}(e_m, e_k) = \left\langle W^{(h)}_{Query} e_m, W^{(h)}_{Key} e_k \right\rangle \tag{3}$$

$$\alpha^{(h)}_{m,k} = \frac{\exp\left(\psi^{(h)}(e_m, e_k)\right)}{\sum_{l=1}^{M} \exp\left(\psi\ (h)\ (e_m, e_1)\right)} \tag{4}$$

$\psi^{(h)}(\cdot, \cdot)$ is the attention function that measures the distance between two vectors. Formula (3) is to calculate the similarity between e_m and other features e_k. The model uses the inner product of vectors to represent distances. $W^{(h)}_{Query}$, $W^{(h)}_{Key}$ is the transformation matrix, for transforming the original embeddings into the new feature space. Equation 4 is to calculate the softmax normalized attention distribution.Then, update the feature m, and weighted summation using α coefficients over the M correlated features. That is, Eq. (5):

$$\widetilde{e}^{(h)}_m = \sum_{k=1}^{M} \alpha^{(h)}_{m,k} \left(W^{(h)}_{Value} e_k \right) \tag{5}$$

Furthermore, a feature may also involve different combined features; the model collects the combined features learned in all subspaces: $\widetilde{e}_m = \widetilde{e}^{(1)}_m \oplus \widetilde{e}^{(2)}_m \oplus \cdots \oplus \widetilde{e}^{(H)}_m$, where $\oplus$ is the connection operator, H is the total number of heads. $\widetilde{e}_m$ is the learned combined feature. To preserve the previously learned combined features, including the original individual (that is, first-order) features, the model adds standard residual connections to the network, which is (6).

$$e^{\text{Res}}_m = \text{ReLu}\left(\widetilde{e}_m + W_{\text{Res}} e_m\right) \tag{6}$$

W_{Res} is used to map e_m to the same size as $\widetilde{e}_m$. e^{Res}_m is the final output of the multi-head self-attention network, which represents the higher-order feature corresponding to feature m.

4) OUTPUT LAYER

a set of feature vectors $\left\{e^{\text{Res}}_m\right\}_{m=1}^{M}$ of the output of the interaction layer. It has the original individual features retained by the residual block and the combined features learned through the multi-head attention mechanism. For the final CTR prediction, concatenate the results for each feature and compute the final output value:

$$\widehat{y} = \sigma\left(w^T \left(e^{\text{Res}}_1 \oplus e^{\text{Res}}_2 \oplus \cdots \oplus e^{\text{Res}}_M\right) + b\right)$$

The shallow and deep features can be effectively fused using this model, overcoming the problem of over-reliance on high-order combined features.

III. MODEL DESIGN

The Bi-LSTM-AutoInt model, based on the AutoInt model, is proposed in this section to improve the accuracy of the model algorithm. Then, noise is added during the perturbation stage of model optimization, and the DP-Bi-LSTM-AutoInt model is proposed to protect the privacy of the recommendation algorithm. The details are as follows:

A. BI-LSTM-AutoInt MODEL

The AutoInt model adopts a multi-head attention mechanism. It estimates the probability of a user clicking on a recommended item by designing a feature extraction scheme to obtain features and model user click behavior. However, since the model cannot fully reduce the high-order significant intersection features, it only finds features with similar relationships to combine. The problem is that mining latent features is insufficient. The following scheme designs the Bi-LSTM-AutoInt model by adding the Bi-LSTM network module. The model learns user click changes in a short period and can fully mine meaningful high-order cross-features.

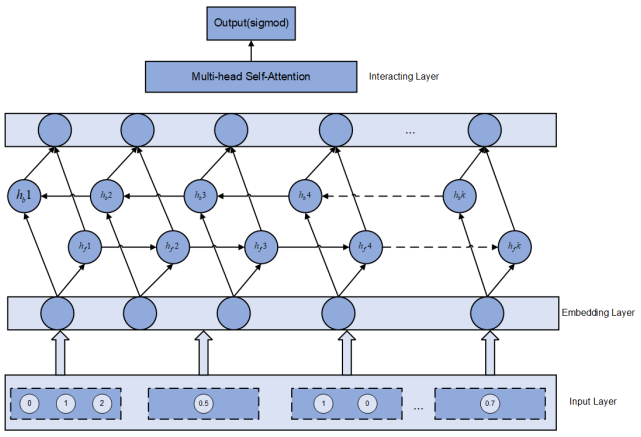


FIGURE 2. Bi-LSTM-AutoInt model.

1) MODEL FRAME

As shown in Fig. 2, the Bi-LSTM-AutoInt model proposed herein can be divided into four parts—the input, embedding, interaction, and output layers. In this study, a Bi-LSTM network is added to the interaction layer. Higher-order

interaction patterns can be captured through the stacking of interaction layers using the Bi-LSTM network to learn the latent expression form between features.

The improved AutoInt model is mainly divided into four parts: original feature input, feature embedding, feature intersection, and target estimation. The specific implementation steps are as follows.

Step1: Embed both numerical and categorical variables into low-dimensional embeddings to obtain a vector representation in a low-dimensional space.

Step2: The encoded vector of the embedding layer is input into Bi-LSTM for training, and the nonlinear data features are output.

Step3: Capture high-order feature intersections through multi-head self-attention. Using multi-head attention, different types of feature combinations can be obtained by mapping features to different spaces, and different levels of modeling can be achieved by stacking multiple interaction layers.

Step4: Input the vector generated by the interaction layer to the output layer and use the sigmoid function to estimate the CTR.

2) BIDIRECTIONAL LSTM NETWORK DESIGN

A Bi-LSTM network is added after the embedding layer to explore user behaviors and interactions between features and discover hidden relationships between features. It has the following advantages: 1) It can learn long-term time-dependent information; 2) It can solve the long-term dependence problem in the click behavior time series data; 3) It can capture user hidden features within a specific amount of time; and 4) It has powerful sequence modeling ability. Fig. 3 shows Bi-LSTM.

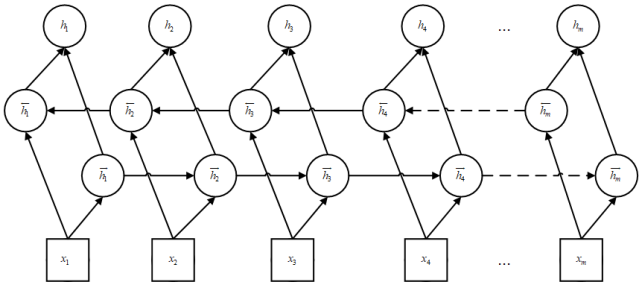


FIGURE 3. Bi-LSTM network diagram.

$\overrightarrow{h}_m$ is the model forward propagation hidden vector, $\overleftarrow{h}_m$ is the model back-propagation hidden vector. Combine $\overrightarrow{h}_m$ with $\overleftarrow{h}_m$ to generate the final hidden vector. The specific calculation equations of LSTM are (7)-(12):

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{7}$$

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{8}$$

$$\widetilde{C}_t = \tanh\left(W_{\widetilde{C}}\left[h_{t-1}, x_t\right] + b_{\widetilde{C}}\right) \tag{9}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \widetilde{C}_t \tag{10}$$

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{11}$$

$$h_t = o_t \otimes \tanh\left(C_t\right) \tag{12}$$

W_f, W_i, $W_{\widetilde{C}}$, W_o is the weight parameter, b_f, b_i, $b_{\widetilde{C}}$, b_o are input gate bias, forget gate bias, cell state bias and output gate bias, x_t as the input sequence, combined with the state of the previous hidden layer h_{t-1}. Form the forget gate f_t through the activation function. Input gate i_t and output gate o_t are also computed by x_t and h_{t-1}. The forget gate f_t is combined with the previous cell state C_{t-1} to determine whether to discard the information. σ represents the sigmoid function, and tanh is the hyperbolic tangent activation function. The model parameters are updated using gradient descent iteration.

This study adopts a bidirectional LSTM module to learn the dependencies between features. In the bidirectional architecture, there are two layers of hidden nodes from two independent LSTM encoders. The forward layer considers historical data in a left-to-right sequence. The backward layer considers future data. This allows the network to preserve information from previous and subsequent states to explore changes in user behavior before and after, processing forward and reverse hidden layer input sequence data. Two LSTM encoders are used to capture dependencies in different directions. The output of the behavior is integrated as shown in (13):

$$F_t = [F\_h_t, F\_c_t, B\_h_t, B\_c_t] \tag{13}$$

F represents forward, B represents backward. In this study, the encoded vectors of the embedding layer are input into Bi-LSTM for training.

B. DP-BI-LSTM-AutoInt MODEL

In the process of deep learning, there is a problem of privacy data leakage in the training data and prediction stages. To solve this problem, this study adopts the differential privacy technology, proposes the DP-Bi-LSTM-AutoInt model and realizes the privacy protection of the CTR model.

DP techniques based on deep learning can be divided into three types: input perturbation, gradient perturbation, and output perturbation. Compared with input perturbation and output perturbation, the gradient perturbation method is more suitable for deep learning algorithms. This is because gradient perturbation does not require strong target assumptions, it only needs to limit the sensitivity of each gradient update, not the entire learning process [25]. Second, since post-processing does not affect DP, gradient perturbation can release noisy gradients at each iteration without breaking privacy guarantees. This study will use gradient perturbation to achieve privacy protection.

For the CTR estimation field, the loss function usually chooses the logarithmic loss function, and the final output expression is (14):

$$Logloss = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log \widehat{y_i} + (1 - y_i)\log\left(1 - \widehat{y_i}\right)\right) \tag{14}$$

y_i and $\hat{y}_i$ are the true and predicted values of the sample i, respectively. n is the total sample size.

In the training of deep learning, the model parameters are generally updated by a gradient descent algorithm, which is (15):

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta_t}{N} \sum_{i=1}^{N} \nabla_{\theta_t} loss\left(x_i, \theta_{t-1}\right) \tag{15}$$

θ_0 is a random initialization parameter. Gradient is the result of direct calculation using input data, and it is the main operation for updating model parameters to achieve privacy protection. Based on the ideas of previous studies [17], this study chooses to add Gaussian noise to the gradient. In each training step T, compute the gradient of L/n random sample L_i of a with sampling probability and constrain the gradient by the l_2-norm, then add noise to the constrained gradient, and finally update the parameter θ_{t+1}, its pseudocode is **Algorithm 1** as follows:

Algorithm 1 Differential Privacy Gradient Descent Algorithm

Input: training samples $\{x_1, \ldots x_N\}$, loss function *Logloss*. Parameter: learning rate α_i, sample size L, preset gradient threshold S

Output: θ_t

1. For $t \in [T]$
2. For $i \in L_t$,calculate $g_t\left(x_i\right) \leftarrow \nabla_{\theta_t} Logloss$
3. $\quad \overline{g}_t\left(x_i\right) \leftarrow g_t\left(x_i\right) / \max\left(1, \frac{\left\|g_t\left(x_i\right)\right\|_2}{S}\right)$
4. $\quad \widetilde{g}_t \leftarrow \frac{1}{L}\left(\sum_i \overline{g}_t\left(x_i\right) + \mathrm{N}\left(0, \sigma^2 S^2 \mathrm{I}\right)\right)$
5. $\quad \theta_{t+1} \leftarrow \theta_t - \alpha_t \bullet \widetilde{g}_t$
6. End For
7. End For

During gradient descent, the sensitivity of the solved gradient in neighboring datasets must be less than some upper bound. Otherwise, too much sensitivity will introduce too much noise, resulting in a low signal-to-noise ratio during training and rendering training meaningless. In this study, a method for controlling the gradient is adopted to actively control the gradient sensitivity and keep the gradient change within a controllable range. In the process of clipping the gradient, if the second normal form of the gradient is less than the given gradient threshold S, the gradient is retained; otherwise, it is adjusted by the ratio of $\left\|g_t\left(x_i\right)\right\|_2 / S$. Restrict the two-normal form of the gradient back to S. Through this operation, the two-normal form of the gradient will not exceed the given two-normal form threshold. Thus, the sensitivity of the gradient is limited. According to Definition 2 given in the original manuscript, the sensitivity formula is: $\Delta_2(f) = \max_{d, d' \in D} \left\|f(d) - f\left(d'\right)\right\|_2$, We denote the gradient as $g(t)$, After trimming with parameter S, this will become $\left\|L2\_clip\left(g(t), S\right) - L2\_clip\left(g\left(t'\right)\right)\right\|_2$. In the worst case, the L_2 norm of $L2\_clip\left(g(t), S\right)$ is S, $L2\_clip\left(g\left(t'\right)\right)$ is 0. Therefore the L_2 sensitivity of the clipping gradient is limited by the clipping parameter S.The sensitivity of the gradient at this time is $\Delta_2(f) = \max_{x_i \in D} \left\|L2\_clip\left(g(t), S\right) - L2\_clip\left(g\left(t'\right)\right)\right\|_2 \leq S$.

Theorem 1: The DP-Bi-LSTM-AutoInt model satisfies differential privacy.

Prove: Since the noise added by the DP-BiLSTM-AutoInt model is $\sigma = S\sqrt{2\ln\frac{1.25}{\delta}} \Big/ \varepsilon$, In this case, the increase or decrease of a sample will only bring no more than S to the gradient, So sensitivity $\Delta_2(f) = \max_{x_i \in D} \left\|L2\_clip\left(g(t), S\right) - L2\_clip\left(g\left(t'\right)\right)\right\|_2 \leq S$, This meets the requirements of Definition 2, then the noise addition process at each iteration satisfies (ε_i, δ) −RDP. According to the characteristics of DP, the algorithm as a whole satisfies (ε, δ) − DP.

□.

IV. EXPERIMENT ANALYSIS

This section will mainly introduce the experimental evaluation of the DP-LSTM-AutoInt model. The results show that the algorithm in this study has the best performance and can obtain better accuracy under privacy protection compared with three other state-of-the-art models.

A. EXPERIMENTAL SETTINGS

1) DATASET

This study adopts the Criteo dataset, which is a classic dataset in the CTR field, and contains click records of 45 million users that are sorted by time. There are 13 continuous features and 26 categorical features. Owing to the large sparsity of the samples, first remove low-frequency features and set low-frequency features to unknown. For the Criteo dataset, filter the features below 10, 5, and 10 times, respectively. And regularize the numerical features. If the feature value is greater than 2, then the log is squared. This processing method is more common in Criteo. We randomly split the dataset into two parts: 80% for training and 20% for testing.

2) EVALUATION METRICS

This study uses two metrics to evaluate the model: AUC (area under the ROC curve) and Logloss (cross-entropy). AUC represents the probability that positive samples will be ranked in front of negative samples and can be used to comprehensively represent the performance of the model. The larger the value of AUC, the more accurate the classification result. Logloss measures the distance between the predicted score and the true label for each sample, where the smaller the loss value, the better the model.

3) EXPERIMENTAL ENVIRONMENT

This study uses python language to implement all models and adjusts the parameters to record the training effect of each model. The experimental environment of this study is shown in Table 1.

4) PARAMETERS

The main privacy parameters of our model is ε, it controls the privacy-utility trade-off. Again, we set δ to 1e-6, which is

TABLE 1. Experimental environment.

EXPERIMENTAL ENVIRONMENT	SPECIFIC PROJECT
GPU	Quadro P4000
Memory	125GB
Deep Learning Framework	Tensorflow-gpu+Keras
Programming language	Python3.6

usually a derivative smaller than the size of the training data. Other parameters related to the model are shown in Table 2.

TABLE 2. Experimental environment.

PARAMETERS	DESCRIBE	VALUE
Learning_rate	Learning rate for gradient descent algorithm	0.001
Num_microbatcher	The batch size into which the input data for each step of the original training algorithm is divided	256
L2_norm_clip	The maximum value S of the L_2 norm when the cumulative gradients of all network parameters for each micro-batch are clipped	1.5
Noise_multiplier	The amount of noise σ added during model training	0.3~1.5
batchsize	batchsize	1024

5) COMPARISON

The algorithm proposed in this study is compared with the work similar to the algorithm in this study to better verify the performance and effect of the algorithm in this study. This study will perform gradient perturbation for the three models and compare the algorithm AUC and Logloss after gradient perturbation.

- DCN model [36]: The full name is Deep & Cross-Network, which can automatically learn the network structure of the cross-network with cross-features, and explicitly perform feature cross.
- FM model [6]: Second-order interaction is achieved through the inner product of hidden vectors.
- DeepFM model [10]: A FM is imposed as a "Wide" module in Wide&Deep to save feature engineering work. DeepFM requires neither pre-training nor feature engineering and can capture both low- and high-order feature interactions.
- EDIF model [8]: Explicit higher-order interactions are performed after the embedding operation, giving different weights to different feature interactions with the SENet attention module and dynamically learning the importance of feature interactions.

B. ANALYSIS OF RESULTS

This section mainly conducts experiments from four aspects: the prediction results of several control models and optimization models, model prediction results under the same privacy protection level, performance comparison of improved models under different privacy budgets, and impact of privacy budgets on CTR prediction. The experimental results are analyzed to verify the superiority and effectiveness of the optimization model proposed in this study from different aspects.

Based on experience, we first set the AutoInt model and Bi-LSTM-AutoInt model embedding dimension d to 16. The number of hidden units d' is 32. There are three Interacting Layers at the same time, and the head of Attention is set to 2.

The error caused by the chance of a single model training in the experimental results was avoided while ensuring the accuracy and stability of the experimental conclusions. 3 and 4 are the optimal values after multiple parameter adjustments, and the other two sections are for 100 experiments to obtain the average value.

1) BI-LSTM-AUTOINT MODEL PERFORMANCE COMPARISON

This section will mainly examine the performance comparison between the Bi-LSTM-AutoInt model and the classic CTR prediction model, to verify that the optimization method proposed in this study is superior to other classic CTR prediction models. Table 3 shows the performance of the Bi-AutoInt model and original AutoInt model [12], EDIF model [8], DeepFM model [10], and FM model [6] under the Criteo dataset, where the epoch is 20.

TABLE 3. Performance comparison between Bi-LSTM-AutoInt model and THE classic CTR model.

METHOD	AUC	LOGLOSS
AutoInt	0.8061	0.4455
FM	0.7892	0.4607
DeepFM	0.8007	0.4508
EDIF	0.8002	0.4513
Bi-LSTM-AutoInt	**0.8126**	**0.4393**

Compared with the original AutoInt model, the Bi-LSTM-AutoInt model proposed in this study has an AUC improvement of 0.65% and a Logloss reduction of 0.62% in the Criteo dataset. This is because the Bi-LSTM-AutoInt model adds a bidirectional LSTM network, which improves the high-order interactive learning ability of features.

Secondly, the accuracy of the DeepFM model is increased by 1.15% compared to the FM model. This is because the FM model is mainly used to process sparse features but cannot interact with high-order features: layer feature interaction representation and deep feature interaction representation. Overall, the Bi-LSTM-AutoInt model proposed in this study has the best performance, which confirms the effectiveness of the algorithm in this study.

2) DP-BI-LSTM-AUTOINT MODEL PERFORMANCE COMPARISON

This section will mainly examine the recommendation performance of different methods under the same gradient disturbance to verify that the proposed method can still guarantee recommendation accuracy after privacy protection. Table 4 shows the performance of different models under the Criteo dataset after gradient perturbation. The privacy budget is set to 1, and the batch size is 1024.

Table 4 shows that when the same model parameters and degree of privacy protection are used, the algorithm proposed in this study outperforms the other three algorithms when performing CTR estimation. Compared with the DeepFM

TABLE 4. Performance comparison of different models after gradient perturbation.

METHOD	AUC	LOGLOSS
DP-DCN	0.7761	0.4751
DP-FM	0.7458	0.4775
DP-DeepFM	0.7721	0.4628
DP-Bi-LSTM-AutoInt	**0.7898**	**0.4602**

model, its AUC index is improved by 1.77%. The AUC index and Logloss index of the DeepFM model are significantly better than the FM model without learning high-order features because high-order and low-order features are extracted simultaneously. Compared with the FM model, the DeepFM model was improved by 2.63% in the optimal AUC index. The results show that the model proposed in this study can effectively learn the interests of users and improve the accuracy of CTR prediction.

3) PERFORMANCE COMPARISON BETWEEN DP-BI-LSTM-AUTOINT MODEL AND BI-LSTM-AUTOINT MODEL

Table 4 compares the DP-Bi-LSTM-AutoInt model under different privacy budgets with the Bi-LSTM-AutoInt model without differential privacy. In this experiment, the privacy budgets in the DP-Bi-LSTM-AutoInt model are respectively set as {0, 1, 2, 3, 4, 5}

As can be seen from Table 4, the accuracy of the Bi-LSTM-AutoInt model is higher, which is due to the data loss caused by the noise added to our method. When the privacy budget is 1, the model adds the most noise, and the model accuracy drops by 2.28 %. However, when the privacy budget is 3, which indicates that moderate noise is added to the model, the accuracy is only 1.45 % worse compared to the Bi-LSTM-AutoInt model. When the privacy budget is 5, the DP-Bi-LSTM-AutoInt model at this time is comparable to the Bi-LSTM-AutoInt model without privacy protection. Although the perturbation of data by our method causes information loss, when the privacy budget is greater than 3.0, that is, when less noise is added, the DP-Bi-LSTM-AutoInt model is nearly equal to the Bi-LSTM-AutoInt model. This shows that the proposed method can effectively guarantee the prediction performance of the model while protecting privacy.

4) IMPACT OF PARAMETERS

This section will mainly examine the performance analysis of the DP-BiLSTM-AutoInt model under different batches. In this experiment, the batchsize is set to {64, 128, 256, 512, 1024} and the privacy budget is set to 1. Tabel 5 reflect the effects on prediction accuracy and loss values under different batches, respectively.

As shown in table 6, the accuracy of the DP-BiLSTM-AutoInt model proposed in this study also differs under different batches. When batchsize = 1024, the model has the best accuracy, which is 0.7% higher than batchsize = 512. When batchsize = 64, the accuracy of the model is poor. The training loss of the DP-BiLSTM-AutoInt model gradually decreases as the number of iterations increases, and the recommendation performance improves. Among the four batches, batchsize = 1024 has the lowest training loss, and the recommendation performance also shows the same trend.

TABLE 5. Performance comparison between DP-Bi-LSTM-AutoInt model and Bi-LSTM-AutoInt model.

METHOD	AUC	LOGLOSS
Bi-LSTM-AutoInt	0.8126	0.4393
DP-Bi-LSTM-AutoInt($\varepsilon = 1$)	0.7898	0.4602
DP-Bi-LSTM-AutoInt($\varepsilon = 2$)	0.7923	0.4564
DP-Bi-LSTM-AutoInt($\varepsilon = 3$)	0.7981	0.4472
DP-Bi-LSTM-AutoInt($\varepsilon = 4$)	0.8003	0.4457
DP-Bi-LSTM-AutoInt($\varepsilon = 5$)	0.8015	0.4439

TABLE 6. Impact of parameters.

DP-BiLSTM-AutoInt	AUC	LOGLOSS
Batchsize=64	0.7535	0.4865
Batchsize=128	0.7628	0.4862
Batchsize=256	0.7646	0.4764
Batchsize=512	0.7726	0.4728
Batchsize=1024	0.7791	0.4677

5) IMPACT OF DIFFERENT PRIVACY BUDGETS

This section mainly considers the impact of different privacy budgets on the accuracy of the algorithm. In this experiment, we set the privacy budget as {1, 2, 3, 4, 5, 6}.

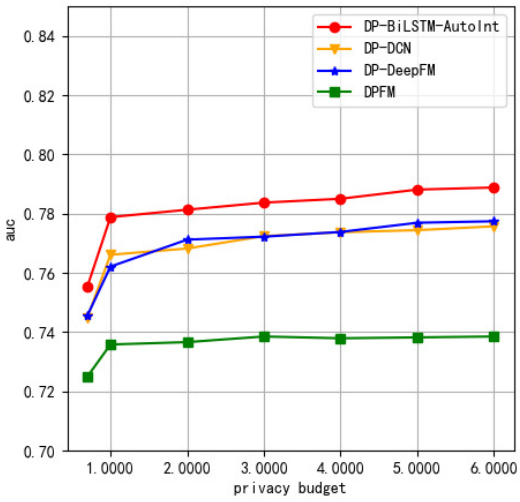


FIGURE 4. Comparison of the model accuracy under different privacy budgets.

Figures 4 and 5 show that, as the privacy budget is increased, the noise injected during the gradient perturbation process decreases, level of privacy protection gradually decreases, accuracy rate gradually increases, and loss rate gradually decreases. When the privacy budget is greater than 2.0, the prediction accuracy of the model stays the same, and the accuracy of this model is slightly higher than that of the other three model data. This shows that the method herein can effectively guarantee the performance of the model while protecting the privacy of users.

The purpose of introducing DP technology into the prediction model, based on the experimental results in the previous

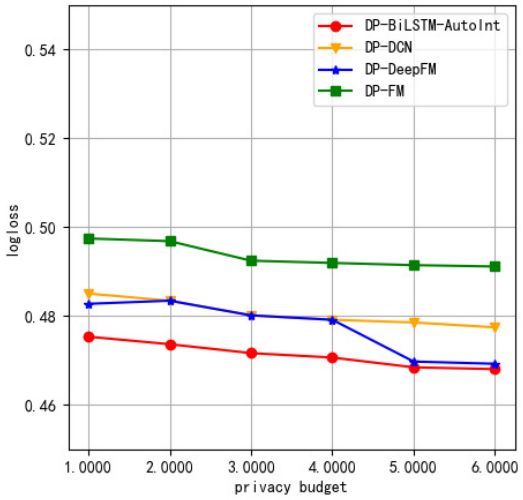


FIGURE 5. Comparison of the model loss values under different privacy budgets.

section, is to sacrifice part of the accuracy in exchange for data security. Although the method used in this study causes information loss due to the disturbance of the data, the results show that it is ineffective. The DP-Bi-LSTM-AutoInt model still outperforms the privacy-preserving FM model.

In summary, the prediction results of the CTR prediction model proposed herein are more secure and reliable than those of the DeepFM, FM, and DCN models. This is because adding the latent representation of the bidirectional LSTM network learning feature improves recommendation accuracy. This study adopts the gradient disturbance method and sets the gradient threshold to prevent the gradient from being too large or too small to affect the model convergence.

V. CONCLUSION

User historical behavior data often contain a large amount of sensitive information. Once such information is obtained by attackers, it will cause unpredictable consequences to users. The AutoInt model, as one of the more effective research outcomes, constructs combined features by citing a multi-head attention mechanism, but this model has the problems of insufficient mining of potential features and privacy concerns. Therefore, this study first improves the AutoInt model and adds a layer of the Bi-LSTM network to explore the changes in user behavior before and after;

Aiming at the problem of privacy concern, it is proposed to apply DP to the CTR prediction model, use the Gaussian mechanism to perturb the gradient, and set the gradient threshold to prevent the model from being unusable due to excessive noise. Through a large number of experiments, it is proved that the algorithm proposed herein can ensure the utility of the recommendation results while protecting data privacy. The next step is to further optimize the CTR prediction model and improve the accuracy of the algorithm on the premise of improving privacy protection, thereby achieving a balance between recommendation accuracy, algorithm performance, and privacy protection.

REFERENCES

[1] E. C. Malthouse, Y. K. Hessary, K. A. Vakeel, R. Burke, and M. Fudurić, "An algorithm for allocating sponsored recommendations and content: Unifying programmatic advertising and recommender systems," *J. Advertising*, vol. 48, no. 4, pp. 366–379, Aug. 2019, doi: 10.1080/00913367.2019.1652123.

[2] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "Online advertising: Analysis of privacy threats and protection approaches," *Comput. Commun.*, vol. 100, pp. 32–51, Mar. 2017, doi: 10.1016/j.comcom.2016.12.016.

[3] Q. Wang, F. Liu, S. Xing, X. Zhao, and T. Li, "Research on CTR prediction based on deep learning," *IEEE Access*, vol. 7, pp. 12779–12789, 2019, doi: 10.1109/ACCESS.2018.2885399.

[4] Q. Wang, F. Liu, S. Xing, and X. Zhao, "Research on CTR prediction based on stacked autoencoder," *Int. J. Speech Technol.*, vol. 49, no. 8, pp. 2970–2981, Aug. 2019, doi: 10.1007/s10489-019-01416-5.

[5] R. Kumar, S. M. Naik, V. D. Naik, S. Shiralli, and M. Husain, "Predicting clicks: CTR estimation of advertisements using logistic regression classifier," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Banglore, India, Jun. 2015, pp. 1134–1138, doi: 10.1109/IADCC.2015.7154880.

[6] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 995–1000, doi: 10.1109/ICDM.2010.127.

[7] H. Jerome Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001. [Online]. Available: http://www.jstor.org/stable/2699986

[8] L. Yang, W. Zheng, and Y. Xiao, "Exploring different interaction among features for CTR prediction," *Soft Comput.*, vol. 26, no. 13, pp. 6233–6243, Jul. 2022, doi: 10.1007/s00500-022-07149-x.

[9] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, New York, NY, USA, Sep. 2016, pp. 7–10, doi: 10.1145/2988450.2988454.

[10] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247*.

[11] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," 2017, *arXiv:1708.04617*.

[12] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Nov. 2019, pp. 1161–1170, doi: 10.1145/3357384.3357925.

[13] 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.*, Oct. 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.

[14] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1770–1782, Sep. 2018, doi: 10.1109/TKDE.2018.2805356.

[15] J. Hinds, E. J. Williams, and A. N. Joinson, "'It wouldn't happen to me': Privacy concerns and perspectives following the Cambridge analytica scandal," *Int. J. Hum.-Comput. Stud.*, vol. 143, Nov. 2020, Art. no. 102498, doi: 10.1016/j.ijhcs.2020.102498.

[16] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 3–18, doi: 10.1109/SP.2017.41.

[17] 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.*, New York, NY, USA, Oct. 2016, pp. 308–318, doi: 10.1145/2976749.2978318.

[18] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the net," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2009, pp. 627–636, doi: 10.1145/1557019.1557090.

[19] J. Ren, X. Xu, Z. Yao, and H. Yu, "Recommender systems based on autoencoder and differential privacy," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Milwaukee, WI, USA, Jul. 2019, pp. 358–363, doi: 10.1109/COMPSAC.2019.00059.

[20] F. Zhang, V. E. Lee, and K.-K. Raymond Choo, "Jo-DPMF: Differentially private matrix factorization learning through joint optimization," *Inf. Sci.*, vol. 467, pp. 271–281, Oct. 2018, doi: 10.1016/j.ins.2018.07.070.

[21] T. Zhu, D. Ye, W. Wang, W. Zhou, and P. S. Yu, "More than privacy: Applying differential privacy in key areas of artificial intelligence," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2824–2843, Jun. 2022, doi: 10.1109/TKDE.2020.3014246.

[22] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 739–753, doi: 10.1109/SP.2019.00065.

[23] Z. Bu, J. Dong, Q. Long, and S. Weijie, "Deep learning with Gaussian differential privacy," *Harvard Data Sci. Rev.*, vol. 2020, pp. 1–36, Jul. 2020, doi: 10.1162/99608f92.cfc5dd25.

[24] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*. Berlin, Germany: Springer, vol. 4978, 2008, pp. 1–19, doi: 10.1007/978-3-540-79228-4_1.

[25] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks* (Book Series Studies in Computational Intelligence). Berlin, Germany: Springer, vol. 385, 2012, pp. 37–45, doi: 10.1007/978-3-642-24797-2_4.

[26] S. U. Khan and R. Baik, "MPPIF-Net: Identification of plasmodium falciparum parasite mitochondrial proteins using deep features with multilayer bi-directional LSTM," *Processes*, vol. 8, no. 6, p. 725, Jun. 2020, doi: 10.3390/pr8060725.

[27] L. Hou, J. Zhang, O. Wu, T. Yu, Z. Wang, Z. Li, J. Gao, Y. Ye, and R. Yao, "Method and dataset entity mining in scientific literature: A CNN + BiLSTM model with self-attention," *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107621, doi: 10.1016/j.knosys.2021.107621.

[28] W. Du, Q. Yan, W. Zhang, and J. Ma, "Leveraging online behaviors for interpretable knowledge-aware patent recommendation," *Internet Res.*, vol. 32, no. 2, pp. 568–587, Mar. 2022, doi: 10.1108/INTR-08-2020-0473.

[29] B. Souissi and A. Ghorbel, "Upper confidence bound integrated genetic algorithm-optimized long short-term memory network for click-through rate prediction," *Appl. Stochastic Models Bus. Ind.*, vol. 38, no. 3, pp. 475–496, May 2022, doi: 10.1002/asmb.2671.

[30] Y. Li, T. Liu, J. Jiang, and L. Zhang, "Hashtag recommendation with topical attention-based LSTM," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, 2016, pp. 3019–3029.

[31] M. Gan and K. Xiao, "R-RNN: Extracting user recent behavior sequence for click-through rate prediction," *IEEE Access*, vol. 7, pp. 111767–111777, 2019, doi: 10.1109/ACCESS.2019.2927717.

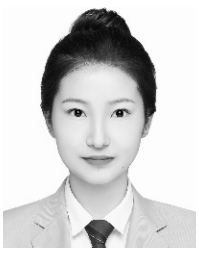
[32] Q. Wang, F. Liu, P. Huang, S. Xing, and X. Zhao, "A hierarchical attention model for CTR prediction based on user interest," *IEEE Syst. J.*, vol. 14, no. 3, pp. 4015–4024, Sep. 2020, doi: 10.1109/JSYST.2019.2943914.

[33] Z. Xiao, L. Yang, W. Jiang, Y. Wei, Y. Hu, and H. Wang, "Deep multi-interest network for click-through rate prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2020, pp. 2265–2268, doi: 10.1145/3340531.3412092.

[34] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 21–25, doi: 10.1109/ICASSP39728.2021.9413901.

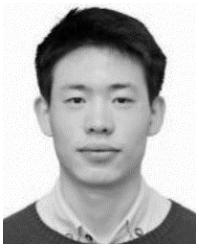
[35] P. Viana and M. Soares, "A hybrid approach for personalized news recommendation in a mobility scenario using long-short user interest," *Int. J. Artif. Intell. Tools*, vol. 26, no. 2, Apr. 2017, Art. no. 1760012, doi: 10.1142/S0218213017600120.

[36] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proc. ADKDD*. New York, NY, USA, 2017, pp. 1–7, doi: 10.1145/3124749.3124754.



LEI TIAN was born in Shandong, China, in 1998. She received the B.S. degree in computer science and technology from Liaocheng University, in 2020. She is currently pursuing the M.S. degree with the School of Electronic Information, Guangxi Minzu University, Nanning, China. Her research interest includes differential privacy.

LINA GE was born in Huanjiang, Guangxi, China, in 1969. She received the M.S. degree in computer science and technology from Guangxi University, Nanning, China, in 2004, and the Ph.D. degree in computer science and technology from the South China University of Technology, Guangdong, China, in 2009. Since 2010, she has been a Professor at the School of Artificial Intelligence, Guangxi Minzu University. She is the author of more than 50 articles. She holds three patents and more than ten software copyrights. Her research interests include information security, the IoTs, and intelligent computing.



ZHE WANG was born in Nanyang, Henan, China, in 1991. He received the B.S. and M.S. degrees in control theory and control engineering from the Zhongyuan University of Technology, Zhengzhou, in 2012, and the Ph.D. degree in intelligent information technology and engineering from Guangxi University, Nanning, in 2019. Since 2019, he has been a Lecturer at the School of Artificial Intelligence, Guangxi Minzu University. His research interests include energy harvesting networks, edge computing, sensor clouds, and the IoTs.

GUIFEN ZHANG was born in Nanning, Guangxi, China, in 1978. She received the B.S. and M.S. degrees in computer science and technology from Guangxi University, in 2008. Since 2009, she has been an Assistant Professor at the School of Artificial Intelligence, Guangxi Minzu University. She is the author of one book and more than ten inventions. She holds two patents and more than ten software copyrights. Her research interests include information security, the IoTs, and intelligent computing.

CHENYANG XU was born in Xuzhou, Jiangsu, China, in 1996. He received the B.S. degree in engineering from the Nanjing University of Posts and Telecommunications, Nanjing, in 2019. He is currently pursuing the M.S. degree in engineering with the School of Artificial Intelligence, Guangxi Minzu University. His research interests include differential privacy and federated learning.

XIA QIN was born in Guilin, Guangxi, China, in 1997. She received the B.S. degree in computer science and technology from Guangxi Minzu University, Nanning, China, in 2020, where she is currently pursuing the M.S. degree with the School of Artificial Intelligence. Her current research interest includes information security.