IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 26, 2019, accepted June 29, 2019, date of publication July 10, 2019, date of current version August 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927717

R-RNN: Extracting User Recent Behavior Sequence for Click-Through Rate Prediction

MINGXIN GAN AND KEJUN XIAO
Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China

Corresponding author: Mingxin Gan (ganmx@ustb.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71871019, Grant 71471016, Grant 71531013, and Grant 71729001, and in part by the Fundamental Research Funds for the Central Universities under Grant FRF-TP-18-013B1.

ABSTRACT Click-through rate (CTR) prediction is of great importance in such web applications as recommender systems, web search, and online advertising. Exploring the interest of a user through feature interactions behind massive behaviors of users is essential for CTR prediction. Recently, deep learning-based models have been proposed, which follow a similar embedding and multi-layer perception (MLP) paradigm and achieve great success in both business and research fields. However, existing studies ignore the sequential characteristics of click-through behaviors of users. To overcome this limitation, we propose a novel model named recent recurrent neural network (R-RNN) to adaptively learn the representation of the interest of a user from his or her overall historical click-through behaviors. R-RNN puts an emphasis on recent click-through behaviors via a novel neural network architecture. Compared to the state-of-the-art methods including the deep interest network (DIN) model recently proposed by Alibaba, R-RNN not only applies the attention mechanism to help capture the representation of the main interest of a user but also incorporates a long short-term memory (LSTM) unit for exploring the trend of the change of the interest of a user behind his or her recent click-through behaviors. Comprehensive experiments are conducted to demonstrate the effectiveness of R-RNN for CTR prediction on a benchmark data set. The results show that R-RNN significantly outperforms existing deep learning models. The results also demonstrate that the length of the recent click-through behavior sequence has an important effect on the prediction performance of the model.

INDEX TERMS Click-through rate prediction, recommender systems, deep learning, recurrent neural network, user behavior sequence.

I. INTRODUCTION

As for the research problem of CTR prediction, the aim is to estimate, as accurately as possible, the probability that a user would click on an ad displayed after submitting a query [1]. The estimated probability for a click that a user will generate indicates his or her interest on a certain item such as a movie, a cloth, and an advertising post, which influences the subsequent decision making [2]–[4]. Thus in many web applications such as recommender systems, the goal is to maximize the number of clicks, and so the items returned to a user can be ranked by the predicted CTR [5].

As the classical click rate prediction method, linear model [6] is simple and effective but lacks the ability to learn feature interactions. In recent years, such machine learning models as linear logistic regression [7], non-linear gradient boosting decision trees [8] and factorization machines [9], have been proposed for CTR prediction with high dependence on feature engineering to capture high-order latent patterns [4]. Most recently, inspired by the success of deep learning in computer vision [10] and natural language processing [11], deep learning-based methods have been proposed for CTR prediction task, which reduce the burden of feature engineering with outstanding prediction performance [4], [5], [12]. These methods follow a similar Embedding&MLP paradigm to learn the nonlinear relations among features in CTR prediction tasks [13].

Learning latent feature interactions effectively behind click-through behaviors is important and a challenge for CTR prediction. Even for interactions that are easy-to-understand,

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

it is impossible for experts to model them exhaustively, especially when the number of features is large [5]. Most recently, DeepFM [5], DIN [13], Wide&Deep Model [14] and PNN [4] enhance model capability greatly by adopting complex MLP network. According to the insights of the Deep Interest Network (DIN) from Alibaba, adaptively calculating the representation vector of user interests by taking into consideration the relevance of historical behaviors brings improvement on the expressive ability of model [13].

However, it is not reasonable to treat all interest of a certain user as the same when predicting a candidate CTR since most recent interest of the user will influence the click-through behavior significantly. For example, a Ph.D. student will click a recommended suit primarily due to the approaching graduate ceremony rather than a recommended cake for the birthday of his mother in the last month. Consequently, he will continue to concern such recommended dress as shoes or ties in the recent future which may match the selected suit. This phenomenon is common which means that the recent click-through behavior is an important signal for CTR prediction. Therefore, besides mining overall history click-through behavior to locate the general interest of a user, the recent click-through behavior should also play an important role in feature interaction for capturing the most recent interest of a user in CTR prediction.

Therefore, we treat the click-through behavior of a user as the combination of a global click-through behavior sequence and a recent click-through behavior sequence, and we adopt different methods to deal with them. For the global behavior, due to the difference between activity levels and the large variance of the numbers of the behaviors of different users, the general method is to compress the behavior sequence of a user into a vector, *i.e*. average pooling. Besides, the attention mechanism is incorporated in the most recent research [4], [15] which is therefore adopted in our proposed model as well. However, global average pooling does not consider the order of the behavior sequence of a user, which leads to the disadvantage that behaviors of the user cannot be treated as time series as RNN does.

Consequently, we propose a novel model, Recent-Recurrent Neural Network (R-RNN), to consider click-through behaviors of a user into the global click-through sequence and the recent click-through sequence, then adopt a recurrent neural network (RNN) to deal with the recent click-through sequence of a user which approximately reflects the current interest of the user. The introduced RNN model not only takes into account the sequence of behavior occurrence but also resorts to the Long Short-Term Memory (LSTM) unit to solve the problem of long-term dependence in the time series data of the click-through behaviors.

The contributions of this paper are summarized as follows:

- We point out the limitation of taking user behavior over all the time as the same to express diverse interests. We, therefore, design a novel recent recurrent neural network (R-RNN) which distinguish user global and recent click-through behavior to learn the representation of the diversity interest of users. R-RNN can improve the expressive ability of the model and better capture the time-varying characteristic of the interest of a user.
- We introduce a Long Short-Term Memory (LSTM) unit to deal with the recent click-through behavior of a user with the consideration of the order of his or her behavior sequence. The LSTM unit effectively overcomes the problem of a long-term dependence on time-series data. Therefore, the proposed structure in R-RNN can better learn the recent click-through behavior of users and significantly improve prediction accuracy. Besides, we found the length of the sequence of the recent click-through behavior has a significant influence on the CTR prediction performance.
- We conduct extensive experiments on Amazon datasets. Results verify the effectiveness of the proposed R-RNN and the training techniques, which show consistent improvement over existing models for CTR prediction.

II. RELATED WORK

In recent years, deep neural networks (DNNs) [16] have shown strong ability in such classification and regression tasks as computer vision [17], speech recognition [18], natural language processing [19], and other intelligent tasks in the applications of smart cities [20], [21]. Since DNNs is able to explore high-order latent patterns as well as automatically learn more expressive feature representations to provide better prediction performance, it is hopeful in user response prediction [4].

Several deep learning models for CTR prediction have been developed recently [1], [22], [23]. In order to improve the multi-field categorical data interaction, Zhang et al. presented an embedding methodology based on pre-training of a factorization machine [24]. Based on the concatenated embedding vectors, multi-layer perceptrons (MLPs) were built to explore feature interactions. However, the quality of embedding initialization is largely limited by the factorization machine [4].

Qu *et al*. proposed Product-based Neural Network (PNN) to captures high-order feature interactions by introducing a product layer between the embedding layer and fully-connected layer [4]. Besides, similar to other deep models, PNN capture little low-order feature interactions, which are also important for CTR prediction.

To model both low- and high-order feature interactions, Cheng *et al*. proposed an interesting hybrid network structure (Wide & Deep) which consisted of a "wide" model and a deep model [14]. The Wide&Deep model requires different inputs for the "wide part" and "deep part", respectively, and the input of "wide part" relies on expertise feature engineering.

DeepFM [5] imposed a factorization machine as "wide" module in Wide&Deep [14] with no pre-training and no feature engineering, and emphasized the interaction between low- and high-order features. The feature of this model is the combination of factor decomposition machine and deep neural network.

In order to improve the quality of the learned semantic match and capturing query intent, DeepIntent adopted a word attention mechanism for the query and ad representations [25]. Deep Crossing proposed a deep architecture to automatically learn such features of an impression as query text, ad text or keywords to predict click probability [26]. Deep Character-level model [27] and Match-Tensor [28] proposed very deep dual network architectures for query and ad embeddings with a matching layer to learn ad impression representations for CTR prediction.

By designing a local activation unit, Zhou *et al*. proposed a novel method named DIN, which introduced the attention mechanism to learn relevant parts of historical behaviors and takes a weighted sum pooling to get the representation of the interest of a user [13]. Behaviors with higher relevance to the candidate item get higher activated weights for the representation of the interest of a user. In order to learn representations that focus on important parts of queries and ads, Gligorijevic *et al.* employed attention models from machine translation and propose a deeply supervised architecture that jointly learns the semantic embeddings of a query and an ad as well as their corresponding CTR [1].

Overall, these methods follow a similar model structure to design their network structure with a combination of an embedding layer for the representation of sparse features and a MLP for learning the combination relations of features at different levels automatically. This kind of CTR prediction model reduces the manual feature engineering jobs greatly [13]. Therefore, our model follows this kind of model structure.

However, this does not mean that DNNs can effectively learn all types of features, for example, time series features, such as the behavior sequence of a user, is difficult to deal with [15]. Thus it is difficult for the above models to effectively model sequential data and utilize features of time series, *e.g*. the click-through behavior of a user.

III. METHODS

A. GLOBAL AND RECENT CLICK-THROUGH SEQUENCES

In CTR prediction, the sequence of the click-through behavior of a user is an important characteristic of the interest of the user. However, the interest of a user may change with the influence of various internal and external factors. Ding and Li believed that the recent click-through behavior of a user can better reflect the interest of the user, *i.e*. items rated recently contribute more to the prediction of the recommendation items [29]. Intuitively, recent behavior data should have a higher value. The longer the click-through behavior, the less it reflects the current interest of a user. The more recent the behavior is, the more it reflects the interest of a user. Therefore, we divide the click-through behavior of a user into global click-through behavior and recent click-through behavior.

Besides, click-through behaviors of a user are relevant and ordered according to the time points, which can be represented as sequences. Consequently, we introduce three definitions to illustrate the click-through behaviors of a user, as the *click-through behavior sequence*, the *global click-through sequence*, and the *recent click-through sequence*, respectively, then give them definitions.

We first give the definition of the *click-through behavior sequence* as follows:

Definition 1: Given a user u, the *click-through behavior sequence* of u can be represented by a sequence of items a_i in ascending order in terms of the click time, as

$$sequence(u) = \langle a_1, a_2, \cdots, a_n \rangle . \tag{1}$$

The relative order of clicked items $< a_i, \ldots, a_j >$ in the sequence reflects the detailed trajectories of the change of the interest of a certain user along with time.

In fact, the main scope of the interest of a user, such as listening to music, loving sports or enjoying quiet, will be stable for a long time. Global click-through behavior sequences can effectively reflect the main scope of the interest of a user. Consequently, we remain to give concern to all click-through behaviors of a user in a long time period and give the definition of the *global click-through sequence* as follows:

Definition 2: Given a user u, the *global click-through sequence* of u can be represented by a sequence of all the items a_i clicked by u before a certain time point T, as

$$sequence_{global}(u, T) = \langle a_1, a_2, \cdots, a_T \rangle . \tag{2}$$

Recent click-through behavior sequence can reflect the state of the interest of a user more accurately and also describe the trend of changes of the interest of a user to some extent. Consequently, the definition of the *recent click-through sequence* is given as follows:

Definition 3: The *recent click-through sequence* of a given user u can be represented by a sequence of k items clicked by u just before a certain time point T, as

$$sequence_{recent}(u, T, k) = \langle a_{T-k+1}, a_{T-k+2}, \cdots, a_T \rangle . \tag{3}$$

B. RECENT-RECURRENT NEURAL NETWORK

The framework of our proposed recent recurrent neural network (R-RNN) model is illustrated in Figure 1. R-RNN follows a similar Embedding&MLP paradigm to reduce the burden of feature engineering jobs, as such popular deep learning methods for CTR prediction as DIN [13], Wide & Deep [12], and DeepFM [5]. In the Embedding&MLP paradigm, input features are first mapped into an embedding layer, then concatenated into fully connected layers, *i.e.* multi-layer perceptron (MLP), to learn nonlinear relations. Based on the paradigm, we further introduce a recurrent neural network and a pooling layer with the incorporation of the attention mechanism which is proposed by DIN [13], to deal with the *global click-through sequence* and the *recent click-through sequence*, respectively.

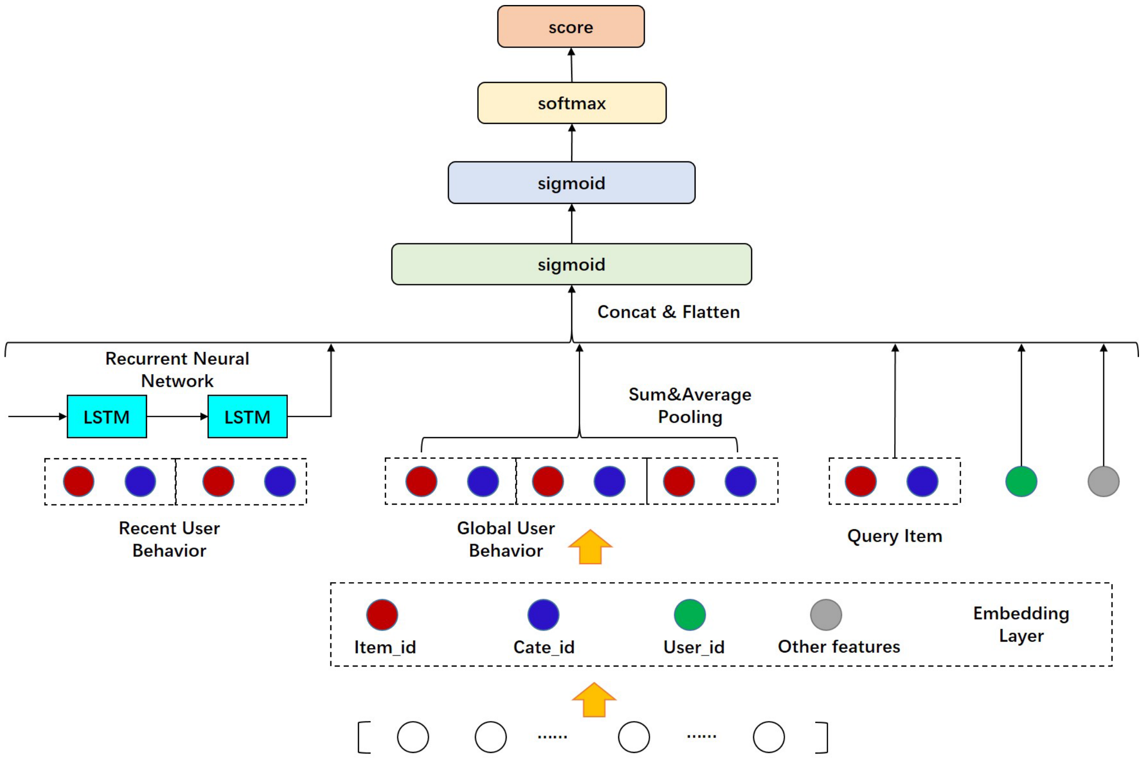

FIGURE 1. **The framework of the proposed recent recurrent neural network (R-RNN) for click-through-rate prediction.**

R-RNN consists of an embedding layer, a pooling layer, a recurrent neural network and multiple fully connected layers. The one-hot feature vectors, such as *user_ids*, *item_ids*, and *category_ids*, are first mapped into low-dimensional embedding vectors via the embedding layer. Then, we construct the click-through behavior of a target user as a global click-through sequence and a recent click-through sequence defined above and feed them into the *sum&average pooing* layer and the recurrent neural network, respectively. Next, the output vectors from the pooling layer and the recurrent neural network are concatenated with other vectors into the fully connected layers (MLP). Finally, the output of MLP is the probability of the predicted click by the target user to a certain item. We listed all symbols used for the proposed R-RNN model in the table of notation (Table 1).

C. EMBEDDING LAYER

To focus on the structure of different neural network models and reduce the influence of feature engineering (feature selection, transformation, and creation) on CTR prediction performance, we only select *user_ids*, *item_ids*, and corresponding *item_category_ids*, and transform them into one-hot vectors as the input features for deep models in performance comparison.

We adopt a one-hot vector $\mathbf{o}_A^i \in R^{1\times|f_A|}$ to denote the feature f_A with $|f_A|$ binary values. For example, if the i^{th} value of f_A equals 1, f_A is represented as

$$[\,\underbrace{00\cdots0}_{i-1}\,10\cdots0\,]. \tag{4}$$

The mapping relation in the embedding layer is represented as

$$\mathbf{e}_A^i = \mathbf{o}_A^i \mathbf{W}_{emb}^A, \tag{5}$$

where $\mathbf{W}_{emb}^A \in R^{|f_A|\times d}$ is the corresponding weight matrix for f_A in the embedding layer, d is the feature dimension of the embedded vectors, $\mathbf{e}_A^i$ denotes the feature vector of the i^{th} value of f_A after the mapping in the embedding layer.

After feature embedding, for a target user u, R-RNN first concatenates the feature vectors for both items and categories of those items together, then sorts the concatenated vectors in ascending order in terms of the time points of the corresponding click-through behavior of the target user. Thus the click-through sequence of the user is represented as

$$sequence_{emb}(u) = \left\langle \left[\mathbf{e}_I^{i1}, \mathbf{e}_C^{j1}\right], \left[\mathbf{e}_I^{i2}, \mathbf{e}_C^{j2}\right], \cdots, \left[\mathbf{e}_I^{in}, \mathbf{e}_C^{jn}\right] \right\rangle, \tag{6}$$

where $\mathbf{e}_I \in R^{1\times d}$ denotes feature vectors for items, $\mathbf{e}_C \in R^{1\times d}$ denotes feature vectors for categories, $\left[\mathbf{e}_I^{i1}, \mathbf{e}_C^{j1}\right]$ denotes the concatenated vectors of both. Consequently, the global click-through sequence for user u before the time point T can be represented as

$$sequence_{global}^{emb}(u, T) = \left\langle \left[\mathbf{e}_I^{i1}, \mathbf{e}_C^{j1}\right], \left[\mathbf{e}_I^{i2}, \mathbf{e}_C^{j2}\right], \cdots, \left[\mathbf{e}_I^{iT}, \mathbf{e}_C^{jT}\right] \right\rangle. \tag{7}$$

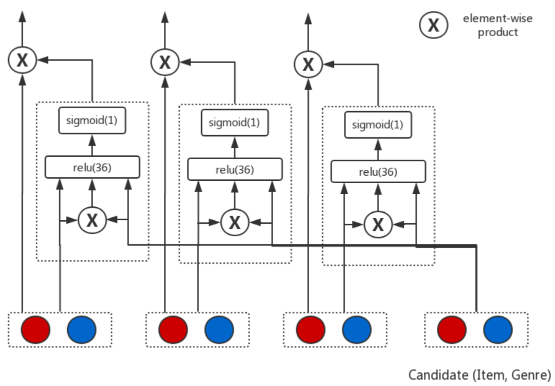

FIGURE 2. The sum&average pooling with the incorporation of attention mechanism.

In addition, for a target user u, his or her global click-through sequences at time T_1 and T_2 are different. If $T_1 > T_2$,

$$\begin{aligned} &length(sequence^{emb}_{global}\,(u, T_1)) \\ &\qquad > length(sequence^{emb}_{global}\,(u, T_2)). \end{aligned} \tag{8}$$

where $length(sequence^{emb}_{global}\,(u, T))$ represents the length of the ordered click-through sequence in terms of time for user u before the time point T.

Meanwhile, the recent click-through sequence which consists of k click-through behaviors for u can be represented as

$$\begin{aligned} &sequence^{emb}_{recent}(u, T, k) \\ &= \left\langle \begin{array}{l} \left[\mathbf{e}_I^{i,n-k}, \mathbf{e}_C^{j,n-k}\right], \left[\mathbf{e}_I^{i,n-k+1}, \mathbf{e}_C^{j,n-k+1}\right], \cdots \\ , \left[\mathbf{e}_I^{iT}, \mathbf{e}_C^{jT}\right] \end{array} \right\rangle. \end{aligned} \tag{9}$$

D. SUM&AVERAGE POOLING

The attention mechanism originates from the Neural Machine Translation (NMT) field [30], which help capture the primary interest of the user from his or her behaviors, such as click-through, query, and so on. The recently developed DIN model by Alibaba was designed with a local activation unit for relevant click-through behaviors of users and takes a weighted sum pooling to obtain the adaptive representation of the interest of users with respect to a given ad [13]. Therefore, we adopt the attention mechanism in the weighted sum pooling introduced in the DIN model to deal with the global click-through sequence which represents the interest of a user with a consideration of the relevance of click-through sequences before a given T.

We design the structure of the *sum&average pooling* layer for our proposed R-RNN model as illustrated in Figure 2. We adopt a local activation unit for the incorporation of DIN in R-RNN model, which concerns relevant parts of the global click-through sequences and takes a weighted sum pooling to generate the representation of the interest of a user to a candidate item. Thus the click-through sequences with higher relevance to the candidate item get higher activated weights and dominate the representation of the interest of the user.

As for the t^{th} click-through behavior a_t^{local} in a global click-through sequence, the local activation unit is represented as

$$a_t^{local} = \sigma\left(relu\left(\left[\mathbf{v}_g^t, \mathbf{v}_g^t \otimes \mathbf{v}_g^c, \mathbf{v}_g^c\right]\right)\mathbf{W}_{att1}\right)\mathbf{W}_{att2}, \tag{10}$$

where, σ and *relu* are the *sigmoid* and *relu* activate functions, respectively. $\mathbf{W}_{att1} \in R^{3d\times l_1}$ represents the weight matrix of the activate unit *relu* and $\mathbf{W}_{att2} \in R^{l_1\times d}$ represents the weight matrix of the activate unit σ. l_1 is the number of neural units in the first hidden layer in the activate unit *relu*. $\boldsymbol{W}$ will be initialized randomly then trained through the backpropagation network. $\mathbf{v}_g^t \in R^{1\times 2d}$ is the concatenated vector of an item feature vector $\mathbf{e}_I^t$ and its category feature vector $\mathbf{e}_C^t$ of the t^{th} user click-through behavior a_t^{local} in the global click-through sequence, *i.e.* $\mathbf{v}_g^t = \left[\mathbf{e}_I^t, \mathbf{e}_C^t\right]$. $\mathbf{v}_g^c \in R^{1\times 2d}$ is the concatenated vector of the candidate item vector $\mathbf{e}_I^c$ and its category vector $\mathbf{e}_C^c$, *i.e.*$\mathbf{v}_g^c = \left[\mathbf{e}_I^c, \mathbf{e}_C^c\right]$. d is the feature dimension of the embedded vectors. $\otimes$ denotes the element-wise multiply, as

$$\begin{aligned} &[a_1, a_2, \ldots, a_n] \otimes [b_1, b_2, \ldots, b_n] \\ &\qquad = [a_1 \times b_1, a_2 \times b_2, \ldots, a_n \times b_n]\,. \end{aligned}$$

Therefore, for different candidate items, the activation units for each element in the global click-through sequence of the target user are different. After obtaining the local activation unit, the concatenated vector will be weighted using the activation unit to adaptively represent the weight of each feature dimension. Finally, the output of the pooling layer is denoted as

$$\mathbf{g} = a_t^{local} \times \mathbf{v}_g^t. \tag{11}$$

As the output of the *sum&average pooling* $\mathbf{g} \in R^{1\times d}$ represents the global state vector for the target user.

Although the network structure of the attention mechanism for *sum&average pooling* significantly improves the expressive ability of the CTR prediction model, there are still some drawbacks, such as the lack of consideration of the time-related order in the click-through sequence as well as the lack of memory units capable of processing long sequences. These problems will be solved by introducing a recurrent neural network for the recent click-through behavior sequences in our proposed R-RNN model.

E. RECURRENT NEURAL NETWORK IN R-RNN

Recurrent neural network (RNN) can receive the input with variable lengths, so it is good at processing time series data. Because of the problem of long-term dependence [15], the LSTM unit is generally adopted in RNN. Therefore, we adopt RNN to deal with the recent click-through sequence for a target user by selecting the most recent k click-through behaviors in his/her click-through sequence before the time point T.

The processing of the recent click-through sequence of a user by the incorporated LSTM is represented in R-RNN

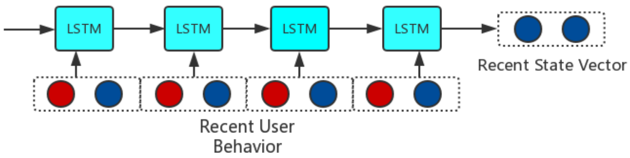

FIGURE 3. Recent click-through sequence is processed by the recurrent neural network (RNN) in the proposed R-RNN model.

model as

$$\mathbf{o}_{lstm}^{t} = lstm_t\left(\mathbf{v}_r^t, \mathbf{o}_{lstm}^{t-1}\right), \tag{12}$$

where $\mathbf{v}_r^t$ is the vector for the t^{th} click-through behavior $a_{T-k+t-1}$ in the recent click-through sequence, which is represented by a concatenated vector from the feature vector of the clicked item and its category vector in $a_{T-k+t-1}$. $\mathbf{o}_{lstm}^{t} \in R^{1\times d}$ represents the output of the processing on the t^{th} user behavior in the recent click-through sequence.

Each unit in LSTM takes both the output of the last unit and the feature vector from a recent click-through sequence as its input, as illustrated in Figure 3. The last output in the RNN will be kept and represented as the recent state vector of the user denoted by $\mathbf{r} \in R^{1\times d}$, and $\mathbf{r} \in \mathbf{o}_{lstm}^{k}$, where k is the length of the recent click-through sequence. Finally, $\mathbf{r}$ will be concatenated with other feature vectors to feed into the full-connection layer in R-RNN model.

F. FULL-CONNECTION LAYER

After the *sum&average pooling* layer and recurrent neural network, the next layer of R-RNN is a three-layer fully connected network which consists of an input layer, a hidden layer, and an output layer.

The processing of the input layer of the full connection layer can be expressed as

$$\mathbf{o}_1 = \sigma\left([\mathbf{g}, \mathbf{r}, \mathbf{v}_c]\,\mathbf{W}_1 + b_1\right), \tag{13}$$

where $\mathbf{g}$ is the global state vector from the output of the *sum&average pooling* layer, $\mathbf{r}$ is the recent state vector from the output of the recurrent neural network, and $\mathbf{v}_c$ is the feature vector of the candidate item. $[\mathbf{g}, \mathbf{r}, \mathbf{v}_c]$ represents the concatenated vector of the above three vectors. $\mathbf{W}_1 \in R^{3d\times l_{f1}}$ is the weight matrix of the input layer of the full-connection network. $b_1 \in R^{1\times l_{f1}}$ is the vector of the bias and l_{f1} is the number of neural units in the input layer. $o_1 \in R^{1\times l_{f1}}$ is the output of the input layer and σ is the sigmoid function defined as

$$sigmoid = \frac{1}{1+e^{x}}. \tag{14}$$

The hidden layer in the fully connected network is defined as

$$\mathbf{o}_2 = \sigma\left(\mathbf{o}_1\mathbf{W}_2 + b_2\right), \tag{15}$$

where $\mathbf{W}_2 \in R^{l_{f1}\times l_{f2}}$ and $b_2 \in R^{1\times l_{f2}}$ are the weight matrix and the bias matrix of the hidden layer, respectively. l_{f2} is the number of neural units, $\mathbf{o}_2 \in R^{l_{f2}}$ is the output of the hidden layer. The output layer of the fully connected network is represented as

$$\mathbf{o}_3 = softmax\left(\mathbf{o}_2\mathbf{W}_3 + b_3\right), \tag{16}$$

$$\hat{y} = argmax\left(\mathbf{o}_3\right), \tag{17}$$

where $\mathbf{W}_3 \in R^{l_{f2}\times l_{f3}}$ and $b_3 \in R^{1\times l_{f3}}$ are the weight matrix and the bias matrix of the output layer, respectively. l_{f3} is the number of the neural units of the output layer, which is equal to 2 in CTR prediction. $\mathbf{o}_3 \in R^{1\times 2}$ is the output vector of the fully connected network, and *softmax* is the activate function defined as

$$softmax = \frac{e^{x_i}}{\sum_{i=1}^{l_{f3}} e^{x_i}}. \tag{18}$$

Softmax function normalizes the output vector. The first dimension indicates the probability that the user will not click on the candidate item, while the second dimension indicates the probability that the user will click on the candidate item. The *argmax* function chooses the maximum value in $\boldsymbol{o}_3$ and outputs its index, which is defined as

$$argmax_{f(x)} = \{x|\forall x : f(y) \le f(x)\}, \hat{y} \in \{0, 1\}. \tag{19}$$

Ultimately, R-RNN predicts whether a user will click on a candidate item, that is, 0 indicates that the user will not click and 1 indicates that the user will click.

G. LOSS FUNCTION

The objective function used in the base model is the negative log-likelihood function defined as:

$$L = -\frac{1}{N}\sum_{(x,y)\epsilon S}\left[y\log\left(p\left(x\right)\right) + (1-y)\,log\left(1-p\left(x\right)\right)\right], \tag{20}$$

where S is the training set of size N, with x as the input of the network and $y \in \{0, 1\}$ as the label, $p(x)$ is the output of the network after the *softmax* function, representing the predicted probability of sample x being clicked.

IV. EXPERIMENTS

A. DATASETS

Amazon Dataset. Amazon Dataset contains product reviews and metadata from Amazon, which is a benchmark dataset for CTR prediction. We conduct experiments on a subset named Electronics, which contains 192,403 users, 63,001 items, 801 categories and 1,689,188 click-through behavior records. Each user or item has more than 5 clicks in the data set. Features adopted in our study include *user_id*, *item_id,* and *item_category_id*.

Since the data set only contains positive samples, following the method in [19], we rank items according to their popularity, *i.e.* times of being clicked, reviewed or checked, and collect the number of negative samples as same as positive samples according to the rule that the more popular the item is, the higher the probability it is selected as a negative sample.

TABLE 1. Notations.

Notation	Meaning
u	a user
T	a certain time point
k	the number of a sequence of items clicked by a user just before a certain time point
a_t^{local}	the t^{th} click-through behavior in a global click-through sequence
x	the input of the network
$\mathbf{o}_A^i$	the feature f_A with $\|f_A\|$ binary values
$\mathbf{W}_{emb}^A$	the weight matrix for f_A in the embedding layer
$\mathbf{W}_{att1}$	the weight matrix of the activate unit *relu*
$\mathbf{W}_{att2}$	the weight matrix of the activate unit σ
$\mathbf{W}_1$	the weight matrix of the input layer of the full-connection network
$\mathbf{W}_2$	the weight matrix of the hidden layer
$\mathbf{W}_3$	the weight matrix of the output layer
d	the feature dimension of vectors
$\mathbf{e}_A^i$	the feature vector of the i^{th} value of f_A after the mapping in the embedding layer
$\mathbf{e}_I$	the feature vectors for items
$\mathbf{e}_I^t$	the feature vectors of the t^{th} item
$\mathbf{e}_C^t$	the feature vectors of the category of the t^{th} item
$\mathbf{e}_I^c$	the feature vectors of the candidate item
$\mathbf{e}_C^c$	the feature vectors of the category of the candidate item
σ	the *sigmoid* activate function
relu	the *relu* activate functions
l_1	the number of neural units in the first hidden layer in the activate unit *relu*
l_{f1}	the number of neural units in the input layer
l_{f2}	the number of neural units in the hidden layer
l_{f3}	the number of the neural units of the output layer
$\mathbf{v}_g^t$	the concatenated vector of an item feature vector
$\mathbf{v}_g^c$	the concatenated vector of the candidate item vector
$\mathbf{v}_r^t$	the vector for the t^{th} click-through behavior $a_{T-k+t-1}$ in the recent click-through sequence
$\mathbf{v}_c$	the feature vector of the candidate item
$\mathbf{g}$	the global state vector from the output of *sum&average pooling* layer
$\mathbf{r}$	the recent state vector from the output of the recurrent neural network
b_1	the bias matrix of the input layer
b_2	the bias matrix of the hidden layer
b_3	the bias matrix of the output layer
$\mathbf{o}_1$	the output of the input layer
$\mathbf{o}_2$	the output of the hidden layer
$\mathbf{o}_3$	the output vector of the full connected network
$\mathbf{o}_{lstm}^t$	the output of the processing on the t^{th} user behavior in the recent click-through sequence

We select a positive and a negative sample from the last click-through behavior of each user as the test set, while other samples are used as the training set.

B. EXPERIMENTAL SETUP

In the training process, we adopt the method Adam [31] as the optimizer, which is an adaptive optimizer and can jump out of the local optimal trap to some extent. We also set up an automatic attenuation mechanism of the learning rate, which gradually decreases from 0.001 with the training iteration. The batch size in training is set to 128. The embedding size is set to 32 for the proposed model, which is consistent for other methods in comparison. The number of fully-connected layers is three and the number of units in the fully-connected neural network in each layer is set to 64, 32 and 1, respectively.

C. METRICS

We adopt AUC which is a widely used metric in our experiments to measure CTR prediction performance. According to the predicted probability of being clicked for all samples in descending order, AUC randomly selects a positive sample and a negative sample. For each pair of a positive and negative sample, the probability that the positive sample ranks ahead of the negative sample is the AUC score. By summarizing AUC scores for all pairs of selected positive and negative samples, we obtain the AUC score for the prediction method. The calculation is as follows:

$$\begin{aligned} AUC &= P\{x_1, x_2 | rank(x_1) < rank(x_2) \cap x_1 \subseteq X_+ \cap x_2 \subseteq X_-\} \\ &= \frac{\sum_i^{N_+} rank\ (i) - \frac{(N_+ +1)N_+}{2}}{N_+ N_-}, \end{aligned} \tag{21}$$

where x_1 and x_2 are the randomly selected samples. X_+ and X_- denote the sets of positive samples and negative samples, respectively. N_+ and N_- represent numbers of positive and negative samples, respectively. $rank(i)$ denotes the rank position of sample i in the ranking list of CTR prediction.

Besides, we introduce RelaImpr metric [13], [32] to measure relative improvement over models. For a random guesser, the value of AUC is 0.5. Therefore, RelaImpr is defined as:

$$RelaImpr = \left(\frac{AUC_{model1} - 0.5}{AUC_{model2} - 0.5} - 1 \right) * 100\%. \tag{22}$$

D. METHODS FOR COMPARISON

We compare the proposed R-RNN model with the existing five deep models for CTR prediction.

BaseModel. BaseModel is a deep learning model paradigm which follows the Embedding&MLP architecture for click-through rate prediction. It acts as a strong baseline for model comparison.

Wide&Deep [14]. Wide&Deep model has been widely accepted, which consists of a wide model, to handle the manually designed cross-product features and a deep model which equals to the BaseModel, to extract nonlinear feature relations automatically.

PNN [4]. PNN can be viewed as an improved version of BaseModel by introducing a product layer after the embedding layer to capture high-order feature interactions.

DeepFM [5]. DeepFM imposes a factorization machine as "wide" module in Wide&Deep to save feature engineering jobs. DeepFM requires neither pre-training nor feature engineering and captures both low- and high-order feature interactions.

DIN [13]. Deep interest network (DIN) is developed in the online advertisement system in Alibaba, which represents the diverse interest of users. DIN also constructs a network based on preferences of users to assign values indicating ads similarities.

TABLE 2. Comparison on prediction performance.

Model	AUC	RelaImpr
Basemodel	0.8442	3.86%
Wide&Deep	0.8458	3.38%
PNN	0.8467	3.11%
DeepFM	0.8471	3.00%
DIN	0.8523	1.48%
R-RNN	**0.8575**	-

V. RESULTS

A. PREDICTION PERFORMANCE

Experimental results in prediction performance for models are illustrated in Table 2. We observe that Wide&Deep, PNN and DeepFM achieve approximately AUC. Because the network structures of the three are adjusted slightly on the base of the BaseModel, there is no significant difference between them. Results demonstrate that well-designed network structure does increase CTR prediction performance of traditional DNN models.

Compared with other models, the improvement of DIN and R-RNN on AUC is significant. The common structure of the design of DIN and R-RNN is the *sum&average pooling* of the attention mechanism. Therefore, the excellent performance of DIN and R-RNN may attribute to the attention mechanism.

We observe that the proposed R-RNN model outperforms BaseModel, Wide& Deep, PNN, DeepFM, by 1.58%, 1.38%, 1.28% and 1.23%, respectively, on AUC, and by 3.86%, 3.38%, 3.11% and 3.00% on RelaImpr. Besides, R-RNN model significantly outperforms DIN by 0.61% on AUC and 1.48% on RelaImpr, primarily due to both the introduced recent click-through behavior sequences and the adopted recurrent neural network which extracts user state vectors from the recent click-through sequences in the design of R-RNN for CTR prediction.

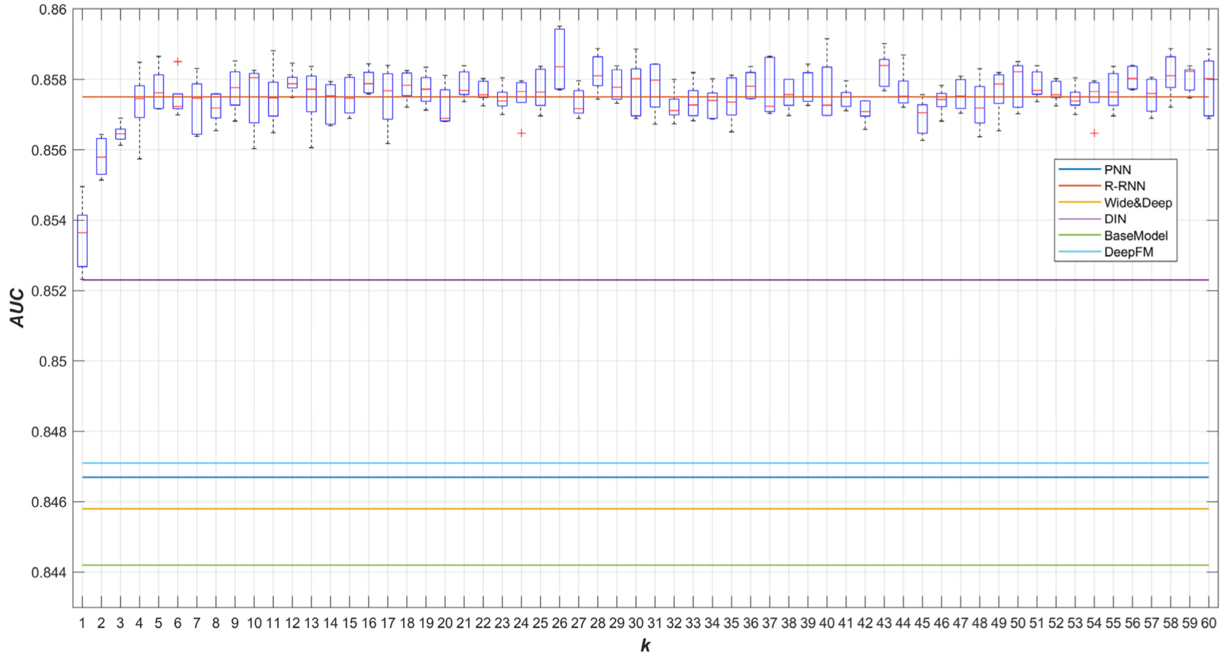

FIGURE 4. Prediction performance of R-RNN with comparison to other state-of-the-art on AUC with various values of k (1-60).

B. ROBUSTNESS TO PARAMETER K

We carry out a series of experiments to explore the effect of the super-parameter k to AUC, which controls the length of recent click-through behavior sequence. As shown in Figure 4, the value of k is from 1 to 60 and the R-RNN model equals DIN when k is 0. We carry out five experiments on each value of k for R-RNN and obtain the average value of AUCs.

We observe that the AUC for R-RNN increases sharply when k increases from 0 to 5 on the Amazon dataset, which demonstrated that using RNN to deal with users' recent click-through behavior sequences significantly improve CTR prediction performance. When k is larger than 5, AUC fluctuates in the range of 0.857-0.858 consistently, due to the random errors and random initialization of parameters as well as the increased occupation of the recent click-through behavior sequence among all sequence, which indicates the improvement of the model on precision being stable as the value of $k \geq 5$.

We further plot the ratio of the number of users to the total number of users, whose behavior sequence is greater than or equal to the value of k, as shown in Figure 5.

According to section 3.1, the sequence length of all users in the data set is no less than 5. Therefore, when the value of k varies from 1 to 5, the ratio remains 1, while when the value of k is greater than 5, the curve shows a distinctly long tail distribution: more than half of users have a sequence with the length less than or equal to 8, while nearly 80% of users have a sequence with the length less than 10. The rest have a long tail distribution with 20% of behavior sequences of users have a length between 11 and 401.

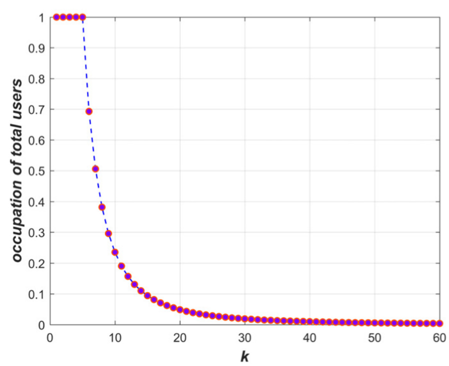

FIGURE 5. Prediction performance of R-RNN on AUC with various values of k (1-60).

C. STATISTICAL ANALYSIS

We conducted statistical analysis on the global and recent sequence for each user and sorted out the distribution of top 20 items in the global sequence and those in the recent sequence, respectively. Results are shown in Figure 6 and 7.

We observe that the distribution of items in both sequences are significantly different, except the first three categories as "*case*", "*headphones*" and "*HDMI Cables*". For example, "*Point & Shoot Digital*" which ranks fourth in the global sequence slips to ninth in the recent sequence while "*Streaming Media Players*" which ranks seventh in the recent sequence ranks fourteenth in the global sequence. The relatively high proportion of "*MP3 Players*" does not even rank top 20 of the recent sequence. This not only reflects the changes of the interest of users, but also conforms to the process of renewal and evolution for electronic product.

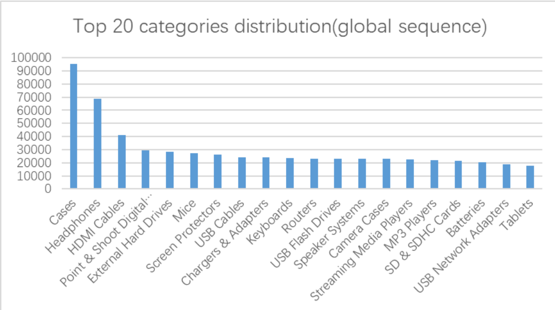

FIGURE 6. Distribution of the top 20 categories in global sequences.

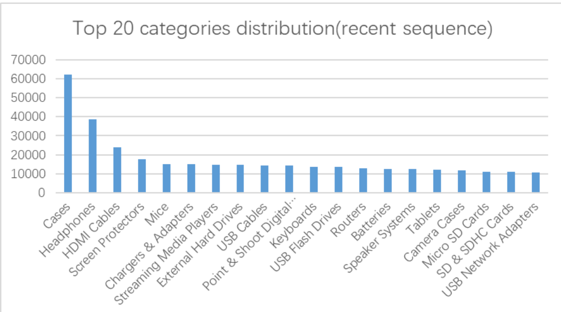

FIGURE 7. Distribution of the top 20 categories in recent sequences.

TABLE 3. Comparison of top 10 categories of global sequences and recent sequences.

Global Sequence		Recent Sequence	
Category	**Rate (%)**	**Category**	**Rate (%)**
Cases	15.79	Cases	17.68
Headphones	13.08	Headphones	12.93
HDMI Cables	8.62	HDMI Cables	8.78
Point & Shoot Digital Cameras	6.51	Screen Protectors	6.91
External Hard Drives	6.46	Mice	6.10
Mice	6.34	Chargers & Adapters	6.23
Screen Protectors	6.32	Streaming Media Players	6.30
USB Cables	6.95	External Hard Drives	6.45
Chargers & Adapters	6.08	USB Cables	6.50
Keyboards	6.12	Point & Shoot Digital Cameras	6.68

The proportion of top 10 categories in both global and recent sequences are illustrated in Table 3. We observe that even if a category ranks at almost the same positions in both sequences, the proportion differs significantly. For example, "*Cases*" ranks top 1 in both sequence, but in the recent sequence, the proportion is 17.68%, while in the global sequence, the proportion is 15.79%, the difference is 1.89%. Moreover, the total proportion of top 10 categories in the recent sequence is 84.56%, which is 3.26% higher than those of the global sequence (81.30%). It demonstrates that the difference of the distributions for categories between the recent and global sequences is significant, which indicates that separating the click-through sequences into global and recent ones can better reflect the statistical difference of them and consequently reveal either the preference of users or the probabilities for the next click-through behaviors.

VI. DISCUSSION AND CONCLUSIONS

We have proposed a novel recent recurrent neural network (R-RNN) to overcome the limitation of current deep models for CTR prediction which take click-through behavior over all the time as the same to express the interest of users. R-RNN distinguishes the global and recent click-through behaviors of users to learn the representation of diversity interests and introduces a recurrent neural network to extract state vectors of users from the recent click-through sequences.

Compared with other state-of-the-art, the improvement of the proposed R-RNN model is significant. For example, as for AUC, R-RNN outperforms Wide&Deep by 1.38%, PNN by 1.28%, DeepFM by 1.23% and DIN by 0.61%, respectively. As for RelaImpr, R-RNN outperforms Wide& Deep by 3.86%, PNN by 3.38%, DeepFM by 3.11% and DIN by 3.00%, respectively.

On one hand, the excellent performance of the proposed R-RNN model may attribute to the*sum&average pooling* of the attention mechanism. On the other hand, the success of the R-RNN model is mainly due to the designed structure on three aspects as below:

(1) Dividing the click-through behaviors into global click-through sequence and recent click-through sequence, R-RNN puts an emphasis on the recent click-through behavior which can better reflect the current interest of users. Therefore, R-RNN improves the expressive ability of the model and better capture the time-varying characteristic of the interest of users.

(2) The recurrent neural network is used to deal with the click-through behavior sequence of users with a consideration of the sequence of behavior in the proposed R-RNN model, which make R-RNN accurately describe the preference of users and capture trajectories of the change of the interest of users.

(3) A long-term and short-term memory unit (LSTM) is introduced in R-RNN to deal with the recent click-through behavior of users with the consideration of the order of behavior sequence, which can self-adaptively learn features from the recent behavior of users. The LSTM unit effectively overcomes the problem of a long-term dependence on time-series data. Besides, the length of the recent click-through behavior sequence has a significant influence on CTR prediction performance.

We, therefore, conclude that considering feature interactions from both global and recent click-through behavior simultaneously brings additional improvement in prediction accuracy. The proposed R-RNN model can be further employed in commercial advertising systems.

After dividing click-through behavior sequence into recent sequence and global sequence, we only realized a combination of average pooling and recurrent neural network, which introduced the attention mechanism. More combinations of different structures can be tried in future work.

REFERENCES

[1] J. Gligorijevic, D. Gligorijevic, I. Stojkovic, X. Bai, A. Goyal, and Z. Obradovic, ''Deeply supervised model for click-through rate prediction in sponsored search,'' in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Anchorage, Ak, USA, Apr. 2019, pp. 1–22.

[2] B. De los Santos and S. Koulayev, ''Optimizing click-through in online rankings with endogenous search refinement,'' *Marketing Sci.*, vol. 36, no. 4, pp. 542–564, 2017.

[3] W. Zhang, S. Yuan, and J. Wang, ''Optimal real-time bidding for display advertising,'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 1077–1086.

[4] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, ''Product-based neural networks for user response prediction,'' in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.

[5] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, ''DeepFM: A factorization-machine based neural network for CTR prediction,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1725–1731.

[6] O. Chapelle, E. Manavoglu, and R. Rosales, ''Simple and scalable response prediction for display advertising,'' *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 1–34, Jan. 2015.

[7] K.-C. Lee, B. Orten, A. Dasdan, and W. Li, ''Estimating conversion rate in display advertising from past performance data,'' in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 768–776.

[8] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers and J. Candela, ''Practical lessons from predicting clicks on ads at Facebook,'' in *Proc. 8th Int. Workshop Data Mining Online Advertising*, New York, NY, USA, Aug. 2014, pp. 1–9.

[9] A.-P. Ta, ''Factorization machines with follow-the-regularized-leader for CTR prediction in display advertising,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct./Nov. 2015, pp. 2889–2891.

[10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[11] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' in *Proc. 3rd Int. Conf. Learn. Represent.*, pp. 1–15, 2014.

[12] 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.*, Sep. 2016, pp. 1–4.

[13] G. Zhou, ''Deep interest evolution network for click-through rate prediction,'' Tech. Rep., 2018.

[14] 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.*, Sep. 2016, pp. 7–10.

[15] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[16] N. Rusk, ''Deep learning,'' *Nature Methods*, vol. 13, p. 35, Dec. 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[18] A. Graves, A.-R. Mohamed, and G. Hinton, ''Speech recognition with deep recurrent neural networks,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed representations of words and phrases and their compositionality,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 3111–3119.

[20] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, ''An intrusion detection system for connected vehicles in smart cities,'' *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101842.

[21] S. Otoum, B. Kantarci, and H. T. Mouftah, ''On the feasibility of deep learning in sensor network intrusion detection,'' *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 68–71, Jun. 2019.

[22] Y. Qu, B. Fang, W. Zhang, R. Tang, M. Niu, H. Guo, Y. Yu, and X. He, ''Product-based neural networks for user response prediction over multi-field categorical data,'' *ACM Trans. Inf. Syst.*, vol. 37, no. 1, p. 5, Oct. 2018.

[23] Y. Xie, D. Jiang, X. Wang, and R. Xu, ''Robust transfer integrated locally kernel embedding for click-through rate prediction,'' *Inf. Sci.*, vol. 491, pp. 190–203, Jul. 2019.

[24] W. Zhang, T. Du, and J. Wang, ''Deep learning over multi-field categorical data—A case study on user response prediction,'' in *Proc. ECIR*, 2016, pp. 45–57.

[25] S. Zhai, K. Chang, R. Zhang, and Z. M. Zhang, ''DeepIntent: Learning attentions for online advertising with recurrent neural networks,'' in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1295–1304.

[26] Y. Shan, T. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, ''Deep crossing: Web-scale modeling without manually crafted combinatorial features,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 255–262.

[27] B. Edizel, A. Mantrach, and X. Bai, ''Deep character-level click-through rate prediction for sponsored search,'' in *Proc. 40th Int. ACM SIGGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 305–314.

[28] A. Jaech, H. Kamisetty, E. Ringger, and C. Clarke, ''Match-Tensor: A deep relevance model for search,'' Jan. 2017, *arXiv:1701.07795*. [Online]. Available: https://arxiv.org/abs/1701.07795

[29] Y. Ding and X. Li, ''Time weight collaborative filtering,'' in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2005, pp. 485–492.

[30] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural Machine Translation by Jointly Learning to Align and Translate,'' in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–8.

[31] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[32] L. Yan, W.-J. Li, G.-R. Xue, and D. Han, ''Coupled group lasso for Web-scale CTR prediction in display advertising,'' in *Proc. 31st Int. Conf. Mach. Learn.*, Jan. 2014, pp. 802–810.

MINGXIN GAN received the B.A. degree in automation from Tsinghua University, China, in 2001, and the Ph.D. degree in management science and engineering from the Beijing Institute of Technology, China, in 2006. She is currently a Professor with the Department of Management Science and Engineering, Donlinks School of Economics and Management, University of Science and Technology Beijing, China. She has published more than 40 papers. Her papers have appeared in *Decision Support Systems*, *Information Systems Frontiers*, *Expert Systems with Applications*, *Bioinformatics*, *BMC Systems Biology*, *Future Generation Computer Systems*, *PLOS ONE*, *World Wide Web Journal*, and so on. Her current research interests include intelligent recommender systems, social network analysis, deep learning, urban computing, and wise medical.

KEJUN XIAO received the bachelor's degree from Hubei University, China, in 2017. He is currently pursuing the master's degree with the Department of Management Science and Engineering, Donlinks School of Economics and Management, University of Science and Technology Beijing, China. His research interests include recommender systems, social network analysis, and deep learning.