

# A Non-sequential Approach to Deep User Interest Model for Click-Through Rate Prediction

Keke Zhao\*

Antgroup

Hang Zhou, Zhejiang, China

keke.zhaokk@antgroup.com

Qi Cao

Antgroup

Hang Zhou, Zhejiang, China

qili.cq@antgroup.com

Xing Zhao\*

Antgroup

Hang Zhou, Zhejiang, China

xing1zhao4@yahoo.com

Linjian Mo

Antgroup

Hang Zhou, Zhejiang, China

linyi01@antgroup.com

## ABSTRACT

Click-Through Rate (CTR) prediction plays an important role in many industrial applications, and recently a lot of attention is paid to the deep interest models which use attention mechanism to capture user interests from historical behaviors. However, most current models are based on sequential models which truncate the behavior sequences by a fixed length, thus have difficulties in handling very long behavior sequences. Another big problem is that sequences with the same length can be quite different in terms of time, carrying completely different meanings. In this paper, we propose a non-sequential approach to tackle the above problems. Specifically, we first represent the behavior data in a sparse key-vector format, where the vector contains rich behavior info such as time, count and category. Next, we enhance the Deep Interest Network to take such rich information into account by a novel attention network. The sparse representation makes it practical to handle large scale long behavior sequences. Finally, we introduce a multidimensional partition framework to mine behavior interactions. The framework can partition data into custom designed time buckets to capture the interactions among information aggregated in different time buckets. Similarly, it can also partition the data into different categories and capture the interactions among them. Experiments are conducted on two public datasets: one is an advertising dataset and the other is a production recommender dataset. Our models outperform other state-of-the-art models on both datasets.

## CCS CONCEPTS

• **Information systems** → **Recommender System; Online Advertising.**

\*Corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

Woodstock '18, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

## KEYWORDS

Non-sequential Approach, User Interest Model, Time-aware Attention, Click-Through Rate, Sparse Representation

### ACM Reference Format:

Keke Zhao, Xing Zhao, Qi Cao, and Linjian Mo. 2018. A Non-sequential Approach to Deep User Interest Model for Click-Through Rate Prediction. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION

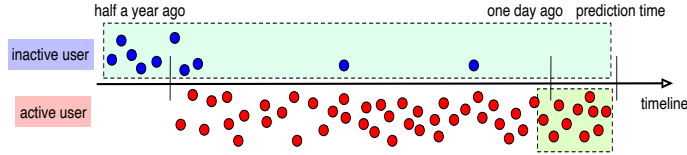
Click-Through Rate (CTR) prediction is a core task in e-commerce such as online advertising and recommendation system, because it is directly related to revenues of the whole platform and also influences user experience and satisfaction. For the CTR prediction model, user behavior modeling is a key part to improve the performance.

Recently, many deep interest models [2, 6, 13, 20, 24, 25] proposed for mining the user's historical behavior have been very effective for practical problems and also have been applied to the real online system successfully. The basic idea behind these models is that user interests are diverse and it will be noisy if captured from user behavior data directly. So attention mechanisms are used to activate historical behaviors with respect to the target item [25]. The behavior closely related to the target item will be given a large attention score while other behaviors will be filtered out by small weights.

Though the interest model is effective to improve performance, most models such as DIEN [24] and DSIN [6] deal with historical behavior data in a sequential manner using GRU or LSTM. A big limitation is that these sequential models can't handle very long sequences in practice. Currently, some practical works are reported, such as the method in [12] is through the help of engineering and the co-design of the machine learning algorithm and online serving system, and the method in [13] is make use of the cascaded search technology to filter the long sequences into short ones. To the best of our knowledge, there are no models specifically designed to deal with long-term behavioral data.

We argue that sequential modeling on raw sequences may not be appropriate for long term behavior data in the case of CTR prediction. On the one hand, sequential modeling techniques such as LSTM have difficulties dealing with long sequence and performance

gets poor for sequence longer than several hundred in practice [12]. It is also well known that computation needed by recurrent network such as RNN, LSTM or GRU either in training or inference is not hardware friendly [23]. In addition, since fixed length input is required for sequential modeling in practice, a max length is usually used. Thus sequences longer than the max length have to be truncated and shorter ones need padding. This is not efficient in terms of data storage and computing. On the other hand, different from sequence data in NLP where interval between words has little information, the time interval between user behaviors may vary greatly and carries valuable information. Sequences with the same length can be quite different in terms of time, carrying completely different meanings, as shown in Figure 1. Intuitively, the behavior sequence patterns of active users are quite different from that of inactive users. For behaviors in a long time horizon, the detailed sequences happened long time ago may not be needed, while capturing the trend change at proper time scale is more important.



**Figure 1: The behavior sequences of active users truncated by a fixed-length are quite different from that of inactive users.**

Motivated by the above observations, we propose a non-sequential approach to deal with the raw behavior data. We first conduct statistical processing on the original behavior data and summarize the count and time distribution of different behaviors. Then the aggregated information is expressed in the sparse key-vector format to eliminate redundancy of storage caused by the repetitions of the same behaviors. Based on the sparse representation, we enhance the Deep Interest Network by introducing the time and count info into the attention calculation. Note that the sparse representation can also reduce the amount of data computation, since the computation amount is related to the average number of keys in the samples, while in the sequence representation, it is determined by the fixed max number of keys in the samples.

For long term user behaviors, temporal trend contains valuable information, and it is important for the model to be able to mine such info. Though our approach represents data in a non-sequential way, the time info is saved in the vector for each key. In order to mine valuable trend information or interactions among behaviors effectively, we propose a framework called Deep Interest Network with Multidimensional Partitions. The behaviors are partitioned into custom designed time buckets. Behaviors in the same time bucket are aggregated, resulting in a much shorter sequence of aggregated information. The sequence position after partition represents a fixed time bucket, and it is consistent among different samples. The interaction layer such as self-attention is used to capture interactions among these time buckets. For long-term historical behavior data, it can effectively capture the dynamic interest

changes of users over time. Similarly, the framework also can partition the data into different categories and capture interactions among these categories if the categories are available.

The main contributions of this paper are summarized as follows:

- We propose a non-sequential method by representing behavior data in the sparse key-vector format. The repetitions in the original behavior data can be aggregated as count and saved in the vector, thus long sequences can be compressed shorter. Using sparse representation, the amount of data that our method can handle is determined by the average length of data, not the maximum length. This makes our model able to deal with a much longer sequence under the same amount of data storage.
- We proposed a novel Time-aware Attention Mechanism which adds side information such as time and count into the attention network. Intuitively, the behaviors that happened several months ago have a very different impact on the user's interest than those of yesterday, and we should give them different weights. Temporal information is very important for mining historical behaviors spanning a long time horizon.
- We propose a Deep Interest Network with Multidimensional Partitions (DINMP). The long behavior sequences are partitioned into custom designed time buckets and information within the same bucket is aggregated, resulting in the fixed length sequences of aggregated information. The interactions among these time buckets are captured through self-attention, thus long-term temporal trend can be modeled. Similarly, the behaviors are partitioned into different categories, and interactions among categories can be captured.
- We conduct extensive experiments on two public datasets. On the Alimama's advertising dataset, we show that our model can handle users' original behavior sequence as long as tens of thousands. The testing results verify the effectiveness of our proposed DINMP.

Solutions discussed here are also applicable in many related tasks, such as conversion rate prediction and user preference modeling.

## 2 RELATED WORK

Deep learning based methods have achieved great success in CTR prediction task [4, 11, 14, 22]. In early age, models such as DeepFM[7], XDeepFM[10], DCN[18] use a deep neural network to capture interactions between features from different fields so that engineers could get rid of boring feature engineering works. Recently, many works focus on learning the users representation from historical behaviors, using different neural network architecture such as CNN [17, 21], RNN[16], Transformer[6, 15, 19, 23] and Capsule[9], etc. Among them, the user interest models [6, 20, 24, 25] which use attention mechanism to capture user interests from historical behaviors have been widely proven effective.

DIN [25] first introduces the mechanism of attention to activate the historical behaviors w.r.t. given target item, and captures the diverse user interests successfully. ATRANK[23] proposes an attention-based framework modeling the influence between users' heterogeneous behaviors. DIEN[24] points out that the temporal relationship between historical behaviors matters for modeling users'

drifting interest. An interest extraction layer based on GRU with auxiliary loss is designed in DIEN. DSIN [6] assume the behavior sequences are composed of sessions and use self-attention to extract users' interests in each session and apply Bi-LSTM to capture users' cross-session interests. DHAN [20] propose a hierarchical attention network to model users' interests from a higher-level attributes (e.g. category, price or brand) and progressively to lower-level attributes or items.

Due to the limitation of the sequential model, the above user interest models still have difficulties in handling very long behavior sequences. A practical system is proposed in [12] to handle long sequences with the co-design of the learning algorithm and online serving system for CTR prediction task. Cascaded search units are introduced in [13] to further improve the max sequence length handled in the real system.

### 3 DINMP

In this section, our work is introduced in two steps. In subsection 3.1, we propose a sparse key-vector format for more efficient data representation, and show that the Deep Interest Network can be implemented using this sparse non-sequential data representation. Then traditional attention network in DIN is enhanced to take into account extra info such as time and count in the vector. In subsection 3.2, we propose a DIN with Multidimensional Partition (DINMP) framework to capture the interest evolution over time and the interaction of behaviors among different categories. The overview of our model is shown in Figure 3.

#### 3.1 Enhanced Deep Interest Network

**3.1.1 Deep Interest Network.** The overall structure of DIN and its subsequent interest models [6, 24] can be clearly described as follows:

- (1) **Feature Representation.** Overall, three groups of features are used as model inputs: user behavior, ad features, other features e.g. user profile, and context features.
- (2) **Embedding.** Embedding is a common technique. Generally the raw categorical features such as user behavior or ad will be transformed into low-dimensional dense vectors through embedding.
- (3) **Sub-Network to capture Users' interests.** A core network is built from ad embedding and user behaviors' embedding. Its output vector stands for users' interest to the target ad.
- (4) **Multiple Layer Perceptron(MLP).** User interest vector, together with the embedding of ad features and other features are concatenated, flattened, and then fed into MLP. The softmax function is used at last to predict the probability of the user clicking on the target ad.

The main idea of basic DIN is that the user's interest to the target ad is closely related to the part of behaviors that is highly similar to the target ad. Specifically, the model constructs an attention score to measure how closely the target ad is tied to each behavior of the user firstly and then the embedding of user's historical behaviors

are aggregated by weighted sum pooling operation, that is,

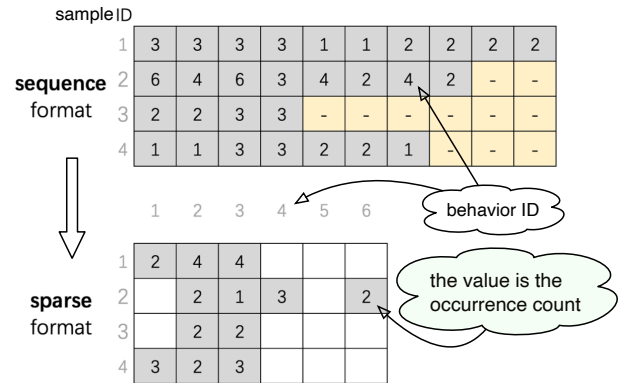
$$\varepsilon_i = \sum_{j=1}^{N_i} a(e_j, q_i) e_j \quad (1)$$

where  $q_i$  is the embedding vector of ad in the current sample  $i$ ,  $\{e_j : j = 1, 2, \dots, N_i\}$  are the embedding vectors of user's historical behaviors, and  $a(e_j, q_i)$  denote the attention function. The formula (1) can also be understood as that we assign the weight  $w_{ij}$  to the embedding vector  $e_j$  of behavior  $j$  in the current sample  $i$ .

Though the attention and sum pooling operation in (1) does not require sequential relations, in practice the DIN as well as other interest methods take the behavior sequence as inputs. In the model implementation, the sequence data is usually represented by a dense tensor with fixed dimension. A max length is picked, and every sequence needs to be either truncated or padded according to this max length. This causes a lot of waste both in storage and computation for short behavior sequences.

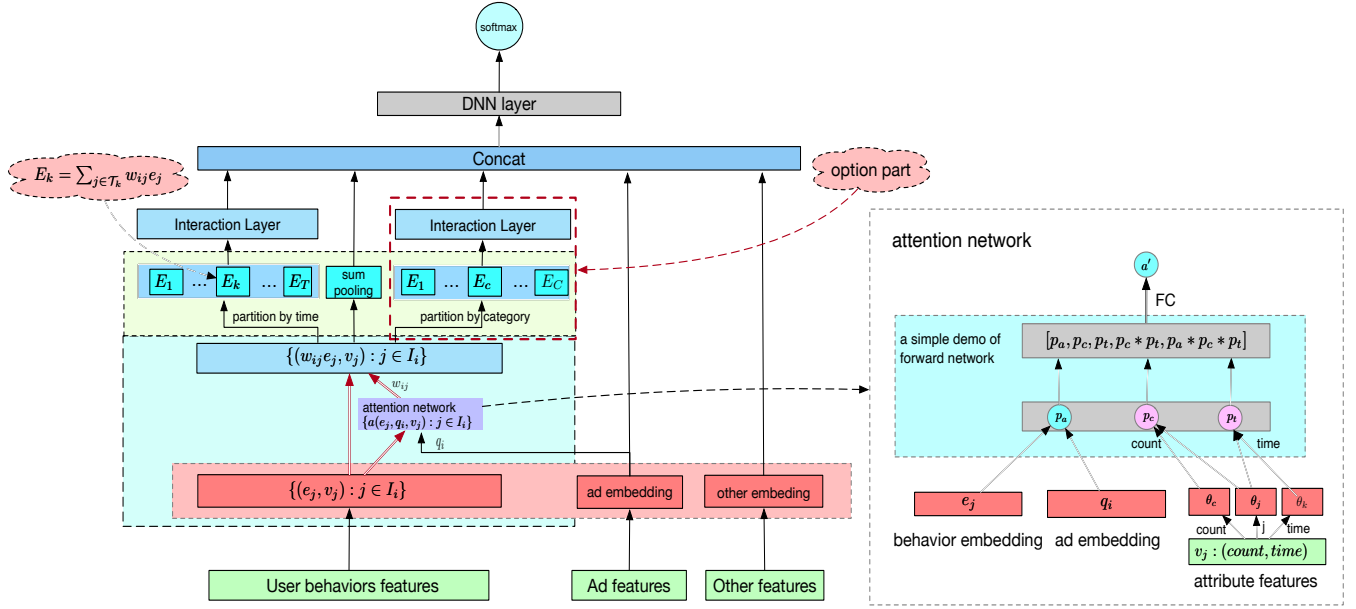
There is another waste related to the repetition of the same behavior in the user's historical behaviors. Taking user transaction data as an example, the user may have repetitive browsing behaviors for the same product at different times. In the current interest models, such repetitive behaviors will be included many times in the raw sequence data, which causes the length of the sequences to be extremely large.

Base on these observations, we introduce a non-sequential sparse key-vector format in order to represent long behaviors more efficiently. The key idea is shown in Figure 2.



**Figure 2: An example is given to illustrate the differences in data representation between the sequence format and the sparse format. The latter is more compact due to no duplicates and padding. Notice that the sparse matrix only stores the non-zero elements.**

**3.1.2 Sparse Key-vector Representation for Behavior Data.** The details of the non-sequential format are introduced next. Note that the user's raw historical behaviors may have repetitive behaviors at different times. We consider a mild aggregation of the original data based on the statistics of each behavior to eliminate redundancy. We summarize the user's behaviors and rewrite the behaviors of



**Figure 3: The overview of our proposed model EDINMP. The behavior embeddings are partitioned by time windows and categories to capture the interest evolution and the interactions among different categories. All the operations are based on the sparse key-vector representation.**

user  $i$  as

$$\{(j, v_j) : j \in J_i\}, \quad (2)$$

where  $v_j$  is a vector of summary info. Some specific forms are as follows:

- $v_j = [c_j]$  that counts the occurrence times of behavior  $j$  directly. We emphasize that this format is sufficient for the basic DIN model.
- $v_j = [t_j, c_j]$  that consists of the time of the last occurrence and the number of occurrences. Note that the time info is a piece of side information which is not included by the current interest models.
- $v_i = [t_i, c_i, \alpha_i]$ . If the behaviors have additional category partitions, the vector can be set as this format, where  $\alpha_i$  represents the category of  $i$ -th behavior belonged.

We call (2) as the key-vector format for behavior data representation which is an extension of the key-value format in machine learning. Similar to the sparse matrix for data representation in key-value format, we use triplet  $(I, J, V)$  for behaviors in key-vector format, where  $I, J$  stands for the row and column index vectors, respectively and  $V$  is a matrix consisting the vectors of all observed elements. Without duplicates or padding, this compact way is more suitable for DIN model.

We emphasize here that the key-vector format reduces the amount of behavior data dramatically in the real world dataset, and it enables to handle long-term behavior data. See Table 1.

**3.1.3 Enhanced Deep Interest Network.** Due to the flexibility of the key-vector format, a lot of additional information of behaviors such as the time, count and category can be added to the attention score. Notice that there is a lot of side information that is not

considered in the interest models. For example, the time information is ignored in DIN[25], DIEN[24], DSIN[6]. We propose a novel Time-aware Attention Mechanism to take the additional information into account. The original attention network  $a(e_j, q_i)$  in (1) is substituted by a new one  $a'(e_j, q_i, v_j)$ .

$$\epsilon_i = \sum_{j \in I_i} a'(e_j, q_i, v_j) e_j = \sum_{j \in I_i} a'(e_j, q_i, t_j, c_j, \alpha_j) e_j \quad (3)$$

here  $v_j = (t_j, c_j, \alpha_j)$  means the last occurrence time, the count and the category of behavior  $j$ . After the modification of (3), we denoted the non-sequential version of DIN as the Enhanced DIN or EDIN for short. EDIN is compatible with the DIN model while introducing additional information as an extended version.

In the rest of this subsection, we will describe how to construct the new attention network in details. We hope that the extra information will be incorporated into the attention network to improve the performance by designing a more reasonable network structure.

Though there is no definitive way on how to design the new attention network, we want to keep a few intuitive principles:

- We keep the old attention as a part, which takes the embedding of target ad and behavior as inputs.
- The impact of additional info such as time and count should be personalized for each behavior. We notice that some behaviors occur frequently while others are always inactive but important.

Motivated by the above principles and through many testing, we design a time-aware attention network as follows.

First, the count and time are designed to interact with the behavior. Take the time  $t_j$  for example, it is parameterized as

$$p_t = e_i^T \theta_k. \quad (4)$$

where  $\theta_j$  is the parameter vector of behavior  $j$  and  $\theta_k$  is the parameter vector for time  $t_j$  assigned into the  $k$ -th time bucket. Similarly, the count is parameterized as  $p_c = \theta_j^T \theta_c$  and the category is parameterized as  $p_\alpha = \theta_j^T \theta_\alpha$ . Note that the new vector  $\theta_j$  for behavior  $j$  is introduced in addition to the embedding  $e_j$  or simply transformed from it  $\theta_j = \theta(e_j)$ . The dimension of  $\theta_j$  is small compared to the embedding vector  $e_j$ .

Beside the old attention value  $p_a = a(e_j, q_i)$ , we have three other factors  $p_t, p_c, p_\alpha$  at hand and the final output is built by a nonlinear combinations:

$$a' = w_0 p_a + w_1 p_t + w_2 p_c + w_3 p_\alpha + w_4 p_t p_c p_\alpha + w_5 p_a p_t p_c p_\alpha$$

where  $w_i$  are the parameters of the network. The time-aware attention network is shown in Figure 3.

### 3.2 DIN with Multidimensional Partition

Similar to DIN, the Enhanced DIN calculates the users' interest for each behavior separately, but lacks the ability to mine interactions among behaviors. To overcome this problem, we propose a DIN with Multidimensional Partition (DINMP), where the behaviors can be partitioned by the time windows or by the behavior categories respectively.

**3.2.1 Multidimensional Partition.** Currently, user interest models do not take time into account when organizing behavior sequences, although some works attempt to use time information, e.g., [23] includes the time information in a simple way i.e. adds the embedding of time to the embedding of behavior directly. Modeling Behavior sequences without considering temporal information can cause problems. For example in Figure 1, if the behavior sequences are truncated by a fixed length, the behavior time range of active users may be quite different from that of inactive users. Most of the behaviors of inactive users may have happened half a year ago, while the behaviors of active users have happened just one day ago. For such an example, the two sequences have very different meanings, but such difference can not be captured by existing sequential models because valuable time info is mostly ignored.

Motivated by the above insights, we align all behaviors of the user before the prediction on the timeline, and we divide it according to the natural date. For example, the behaviors in recent one year can be partitioned by a custom-designed set of time points such as 1 day ago, 2 days ago, 4 days ago, 1 week ago, 2 weeks ago, 1 month ago, 2 months ago, 3 months ago, 6 months ago, etc. After partition, all the behaviors are aligned into fixed groups.

For all behaviors within each time window, we can use the module described in EDIN (3.1.1) to capture users' interest. EDIN represents user behaviors in the key-vector format as inputs, and returns an aggregated vector  $E_k$  as user' interest in that time window. Finally, all of the outputs are collected in sequence as  $[E_1, \dots, E_k, \dots, E_T]$ . Notice that it forms a new time-aligned sequence of aggregated user behavior representations, which allows

the neural network to add some higher-order layers, such as transformer or concatenation, to improve the model's performance further.

The above partition and aggregation way to improve performance can be extended to multiple dimensions. For example, a natural way is to partition and aggregate by category of the behaviors.

**3.2.2 Model Overview.** The core network structures of our DINMP are shown in Figure 3. Most key calculations are listed as the following:

**Input:** User behavior feature in key-vector format, or mathematically as follows  $\{(j, v_j) : j \in I_i\}$ , where  $I_i$  is the index set of keys in sample  $i$ .

1. Look up embeddings for behaviors, and we get  $\{(e_j, v_j) : j \in I_i\}$ .
2. Use attention network to compute weight between ad and behavior i.e.  $w_{ij} = a'(e_j, q_i, v_j)$ , here  $q_i$  means ad embedding.
3. Compute weighted embedding sets, and get a new set  $\{(w_{ij}e_j, v_j) : j \in I_i\}$ .
4. Partition the behaviors and aggregate embeddings.
  - (a) Denote  $\{\mathcal{T}_k\}_{k=1}^T$  be the partition by time window, i.e.,  $I_i = \cup_{k=1}^T \mathcal{T}_k$ . Aggregate the embeddings as  $E_k = \sum_{j \in \mathcal{T}_k} w_{ij}e_j$ ,  $k = 1, \dots, T$ .
  - (b) Let  $\{C_k\}_{k=1}^C$  be the partition of  $I_i$  by category, i.e.,  $I_i = \cup_{k=1}^C C_k$ . Aggregate the embeddings as  $E_c = \sum_{j \in C_c} w_{ij}e_j$ ,  $c = 1, \dots, C$ .
  - (c) Sum pooling i.e., aggregate all the weighted embeddings.  $E = \sum_{j \in I_i} w_{ij}e_j$

Notice that the network described in the above box can be implemented with TensorFlow [1] using sparse representation methods.

## 4 EXPERIMENTS

In this section, we mainly present our experiments in detail, including three datasets introduction, evaluation metric, experimental setup, comparison with baseline methods, ablation study, and visualization analysis. Experiments on public datasets with user behaviors demonstrate the effectiveness of our proposed approach that outperforms state-of-the-art methods on the CTR prediction task. Both the public datasets and experiment codes are made available.

### 4.1 Datasets and Experimental Setup

**Advertising dataset:** Advertising dataset is a public dataset released by Alimama<sup>1</sup>, an online advertising platform in China. It contains 26 million records from ad display/click logs of 1 million users and 800 thousand ads in 8 days. We used the first 7 days' logs as training samples, and the last day's logs as test samples. The user behavior data set covers the shopping behavior in 22 days. We process the user behaviors data by only extract the **ipv**(click) behaviors and ignore other three types (**buy**, **cart**, **fav**) of behaviors as in [6].

<sup>1</sup><https://tianchi.aliyun.com/dataset/dataDetail?dataId=56>

Sample information is given in the following table 1. Note that after sparse processing, the average number of keys in each sample is only 175. In the detailed sequence, the maximum length is over 40 thousands. It is nearly 250 times as long as processed.

**Table 1: The description of the advertising dataset. By simple aggregation, the raw behavior sequence data can be reduced into a very compact key-vector format.**

train sample	$2.37 \times 10^7$	test sample	$2.8 \times 10^6$
max #behavior	43251	avg #behavior	606
max #key	3809	avg #key	175
#key < 500	96%	#key < 400	92%

**Taobao<sup>2</sup>:** Taobao Dataset is a collection of user behaviors from Taobao’s recommender system[26], which consists of different types of user behaviors such as click, purchase, etc. It contains user behavior sequences of about one million users. We take the click behaviors for each user and sort them according to time in an attempt to construct the behavior sequence. Assuming there are  $T$  behaviors of a user, we use the former  $T - 1$  clicked products as features to predict whether users will click the  $T$ -th product. Note that our model takes all the  $T - 1$  clicked products as input features, while for other comparison methods, the behavior sequence is truncated at length 200.

Sample information is given in the following table 2. the average number of behaviors is 90, which is small compare to truncated length 200.

**Table 2: The description of the Taobao recommender dataset. The average number of behaviors is 90.**

train sample	691456	test sample	296192
max #behavior	846	avg #behavior	90
#behavior < 400	99%	#behavior < 200	90%

**Amazon(Electro):** Amazon Electronics data set is used to evaluate our proposed multi-dimensional partition module. The entire dataset contains product review information and metadata from Amazon. Our task is to predict  $(k + 1)$ -th reviewed item by using the first  $k(k \geq 5)$  reviewed items. We construct one positive sample by taking user’s historical reviews, and generate three negative samples from items not reviewed by this user. The numbers of training and testing sets are 697092 and 190468. Notice that this data set includes coarse category information that is different from the above datasets and all the items can be categorized into 51 groups.

**Experimental Setup:** For all our tests, we use Adam[8] solver and set the learning rate to be 0.001. Layers of DNN are set by (128, 256, 80, 256). For taobao dataset, the behaviors are represented in two features, i.e.,  $(items : [category, time])$  and  $(category, [time])$ , and the dimension of embedding is 16 which is same as all the other methods [12]. For alimama dataset, two behavior features are  $(brand : [count, time])$  and  $(category : [count, time])$ , and the dimension of embedding for behaviors features is 8 while that for

other features is 4. We take AUC as the metric for measurement of model performance.

## 4.2 Compared Methods

The compared methods are listed here.

- **Embedding&MLP.** These methods represent a general paradigm in the early CTR prediction field. It first maps large scale sparse features into low dimensional embedding space. Afterward, embedding vectors are transformed into fixed-length vectors in a group-wise manner fed into multilayer perceptron (MLP) to predict the result.
- **YoutubetNet[5].** YoutubeNet is a technically designed model which treats users’ historical behaviors equally and utilizes average pooling operation to obtain user representation.
- **Wide&Deep[4].** Wide&Deep is trained by wide linear models and deep neural networks jointly, which combines the benefits of memorization and generalization.
- **DIN[25].** A local activation unit is designed in DIN to adaptive learn the interest’s representation of the user from historical behaviors. This representation vector varies over different ads, improving the expressive ability of the model greatly.
- **GRU4Rec[16].** It is the first work to exploit the RNN models in the recommender system and introduces a new ranking loss function suited to the task of training these models.
- **RUM [3].** RUM leverages the external memory matrix to store and update the user’s historical records, which also enhances the per item-level or feature-level correlations between the user’s historical records and future interests.
- **DIEN [24].** DIEN extracts latent temporal interests from user behaviors and models interests evolving process. Auxiliary loss makes hidden states more expressive to represent latent interests and AUGRU models the specific interest evolving processes for different target items.
- **DSIN [6].** It divides users’ behavior sequences into sessions by their occurring time. It extracts users’ interests in each session first and then model how users’ interests evolve and interact among sessions.
- **MIMN.** MIMN adds memory utilization regularization and memory induction unit in a traditional memory network that make it more efficient for dealing with long user behavior sequences under limited storage and brings remarkable gain over model performance.

## 4.3 Metrics

Area Under ROC Curve (AUC) is a widely used metric in CTR prediction field, which means the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example, reflects the ranking ability of the model. Following the previous works, AUC is regarded as the evaluation metric to validate the performance of the proposed DINMP for CTR prediction on the Alimama and Taobao dataset. The definition of AUC is described as follows:

$$AUC = \frac{1}{m^+m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} \mathbf{1}(f(x^+) > f(x^-))$$

<sup>2</sup><https://tianchi.aliyun.com/dataset/dataDetail?dataId=649>



where  $D^+$  is the collection of all positive examples,  $D^-$  is the collection of all negative examples,  $f(\cdot)$  is the result of the model's prediction of the sample  $x$  and  $1$  is the indicator function.

RelaImpr [20] metric is used to measure the relative improvement over models, which is defined as:

$$RelaImpr = \left( \frac{AUC(\text{measured model}) - 0.5}{AUC(\text{base model}) - 0.5} - 1 \right) \times 100\%.$$

#### 4.4 Results on Alimama Datasets

To evaluate the effectiveness of the proposed DINMP, we compare DINMP to some state-of-the-art methods for CTR prediction. Existing methods used in our comparative experiments mainly include YoutubeNet, Wide&Deep, DIN, DIEN, as well as DSIN.

**Table 3: Model performance (AUC) on Alimama**

Model	AUC	RelaImpr (%)
YoutubeNet	0.6313	-1.28
Wide&Deep	0.6326	-0.30
DIN	0.6330	0.00
DIEN	0.6343	0.98
DSIN	0.6375	3.38
DINMP	0.6442	8.42

**Experimental results on the Alimama dataset:** In this part, the performance of compared baseline approaches are directly adopted from Feng et al. experiments [6]. As described in Table 3, making comparisons among previous models, we report several conclusions: 1) As expected, DIN has achieved superior performance compared with simple model including YoutubeNet, Wide&Deep with improvements of 0.0027 and 0.0004 on AUC. It validates again the importance of user interest. 2) Moreover, DSIN is treated as the state-of-the-art baseline beating basic user interest model (DIN and DIEN) with its capability of finding behaviors' session information. 3) The proposed DINMP method yields a new state-of-the-art performance for CTR prediction on the Alimama advertising dataset, which exceeds DSIN with improvements of 0.0067. It could be that our models make use of all behaviors covering 22 days whereas previous approaches truncated behavior sequence in a fixed length resulting in information lose. Another reason is that DINMP adopts time-aware attention mechanism that makes use of the behavior time as an important factor to achieve effective interest expression.

#### 4.5 Result on Taobao Datasets

In this section, we make an experiment on Taobao dataset to further verify the efficiency of the proposed DINMP model. Some state-of-the-art baselines for CTR prediction on this dataset are applied to the comparative experiments, e.g. RUM, DIEN, and MIMN. Among them, the data are quoted from the experiment of Pi et al

**Experimental results on the Taobao dataset:** As shown in Table 5, we have the following observations. 1) The baseline deep learning approach Embedding&MLP is the lowest performance, which only obtains 0.8709 AUC. 2) By focusing on user interest information, DIN improves 0.0124 compared with Embedding&MLP.

**Table 4: Model performance (AUC) on Taobao Datasets**

Model	AUC	RelaImpr (%)
Embedding&MLP	0.8709	3.24
DIN	0.8833	0.00
GRU4REC	0.9006	4.51
RUM	0.9018	4.83
DIEN	0.9081	6.47
MIMN	0.9179	9.03
DINMP	0.9342	13.28

3) Owing to capture interest evolving process, DIEN has made a further promotion in performance. 4) Thanks to a novel memory-based architecture, MIMN achieves the best baseline performance by capturing user interests from long sequential behavior data. Although existing methods obtains great breakthrough, it still faces the limited sequence length problem. 5) Compared with all approaches, the proposed DINMP contributes a new state-of-the-art (0.9342 AUC) by employing non-sequential behavior representation.

#### 4.6 Ablation Studies

In this section, we perform ablation studies to verify the effectiveness of each module (i.e., Sparse Key Vector representation, the Time-aware Attention mechanism, Multi-dimensional Partition module) for the proposed method DINMP on three published datasets. In detail, we design five different models and each of them is described as follows:

- **DIN.** In ablation studies, we employ basic DIN as a baseline system, which takes truncated behavior sequences as input to obtain user interest representation.
- **DINSKV.** Based on basic DIN, we implement a sparse version of DIN by representing behavior data in sparse key-vector format. Due to the sparse representation, the original behavior data is compressed shorter and the amount of data that the model can handle is determined by the average length of data, not the maximum length. Thus, it is able to tackle all behavior data as input.
- **EDIN.** Considering the impact of side information such as time on user interest, on the basis of DINSKV, we use a time-aware attention mechanism to enhance the local activation unit in DIN.
- **DINTP.** Adding time partition branch to EDIN model, in this way, DINTP can further capture the interactions among different time buckets.
- **DINMP.** DINMP makes further optimization on DINTP method by adding category partition, which captures the interactions among different category.

**Ablation Studies on Alimama Dataset:** From the result in Table 5, we have the following observation on Alimama dataset. First, the model effect has been greatly improved on Alimama dataset when DIN use Sparse Key Vector (SKV) representation. Compared with baseline model DIN, DINSKV rise from 0.6330 to 0.6424 in AUC, which proves the effectiveness of SKV format. Here, we point out that since the SKV representation was effective, the average number of keys involved in each sample is only 175, and the data

**Table 5: Ablation Studies Result on Alimama Advertising Dataset and Taobao Recommender Dataset.**

Model	Alimama Advertising Dataset		Taobao Recommender Dataset	
	AUC	RelaImpr (%)	AUC	RelaImpr (%)
DIN	0.6330	0.00	0.8833	0.00
DINSKV	0.6424	7.07	0.8876	1.12
EDIN	0.6436	7.97	0.9285	11.79
DINMP	0.6442	8.42	0.9342	13.28

storage do not increase. Second, by adding a Time-aware Attention (TAA) mechanism in DINSKV, which takes time, count and category info into account, the effect on Alimama datasets has been improved 0.0012 AUC. It proved that TAA has digged out user interest by considering time evolution in behavior. Finally, by introducing Multi-dimensional Partition (MP) module, DINMP makes a remarkable improvement (0.6442 vs. 0.6436) on performance compared to EDIN. It indicates that MP module brings an effective gain.

**Ablation Studies on Taobao Dataset:** In Taobao dataset, we obtain the same observation with Alimama dataset, which further proves the efficiency of our proposed methods. It is worth noting that AUC score becomes 0.9285, when adding TAA module in DINSKV rising 0.0409. Moreover, the AUC increase from 0.9285 to 0.934 comparing DINMP with EDIN, which is a very significant lift. It shows tha MP module further improves the performance.

**Table 6: Ablation Studies Result on Amazon Dataset**

Method	AUC	RelaImpr (%)
EDIN	0.8210	0.00
DINTP	0.8237	0.80
DINMP	0.8285	2.30

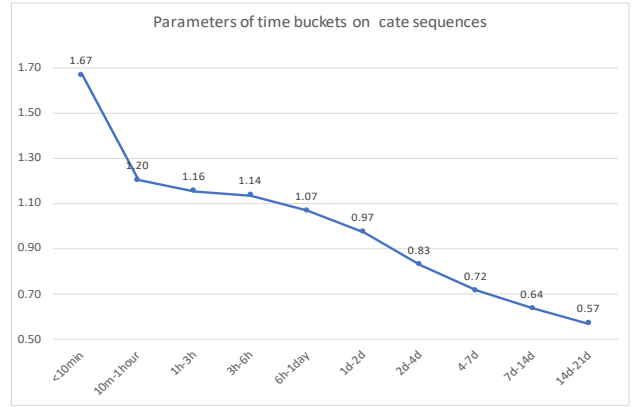
**Ablation Studies on Amazon Dataset:** This part mainly verifies the effectiveness of the proposed multi-dimensional partition module because only Amazon dataset satisfied multi-dimension. As is shown above, DINTP outperforms EDIN (0.819 vs. 0.815) on AUC, which shows that the time partition of behavior sequence can lead to better performance. Moreover, compared with DINTP model which only exploits time partition, DINMP benefits from the combination of time and category partition improving 2.3% on RelaImpr, which indicates the superior of multi-dimensional partition.

#### 4.7 Visualization of Time-aware Attention

To further verify the superiority of the proposed time-aware attention mechanism, we perform a visualization experiment to visualize the time factor in the TAA mechanism, shown in Fig. 4. For the sake of visualization, we simplify the time factor in Eq. 4 to be  $p_t = \theta_k$ , where  $\theta_k \in \mathbb{R}^1$  is parameter for time  $t$  assigned into the  $k$ -th time buckets. We retrained our DINMP model on the Alimama dataset and saved the final parameters.

We plot the time factor  $p_t$  over different time buckets in Fig. 4 corresponding category behavior feature on the Alimama dataset. From this figure, we can intuitively see that the time factor is almost

getting smaller as time goes on. In particular, the time factor of the recent 10-minute bucket is nearly three times as large as that of the 2-3-week bucket. The above phenomenons indicate that time factor is discriminative, and our time-aware attention method with time and category information can effectively make the network focus on user recent interest.

**Figure 4: The time factor  $p_t$  over different time buckets on Alimama dataset.**

## 5 CONCLUSIONS

In this paper, we propose a Deep Interest Network with Multi-dimensional Partition (DINMP) to improve the performance of CTR prediction. The proposed DINMP consists of three part. First, a non-sequential approach is employed to represent the behaviors sequence in a sparse key-vector format. The vector contains rich behavior info such as time, frequency, and category. Moreover, a novel time-aware attention mechanism takes such rich information into account mining deep interest of the user. In addition, a multi-dimensional partition module is introduced, where behaviors are partitioned into different time buckets to capture interaction in different time. We also argued that behaviors can also be partitioned into different categories and interactions among these categories can be captured. At last, experiment results demonstrate the effectiveness of our models both on advertising and recommender datasets.

It is worth mentioning in this work that the sparse key-vector format can represent more valuable data in a compact way. We believe this non-sequential approach opens a new direction for modeling extremely long-term user behavior data. In the future, we



will push forward the research on learning algorithms as well as the online serving system.

## REFERENCES

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} '16)*. 265–283.
- [2] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-Commerce Recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (Anchorage, Alaska) (DLP-KDD '19)*. New York, NY, USA, Article 12, 4 pages. <https://doi.org/10.1145/3326937.3341261>
- [3] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (Marina Del Rey, CA, USA) (WSDM '18)*. New York, NY, USA, 108–116. <https://doi.org/10.1145/3159652.3159668>
- [4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh Aradhya, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Isipir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihari Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (Boston, MA, USA) (DLRS 2016)*. New York, NY, USA, 7–10. <https://doi.org/10.1145/2988450.2988454>
- [5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (Boston, Massachusetts, USA) (RecSys '16)*. New York, NY, USA, 191–198. <https://doi.org/10.1145/2959100.2959190>
- [6] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 2301–2307. <https://doi.org/10.24963/ijcai.2019/319>
- [7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (Melbourne, Australia) (IJCAI'17)*. 1725–1731.
- [8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- [9] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (Beijing, China) (CIKM '19)*. New York, NY, USA, 2615–2623. <https://doi.org/10.1145/3357384.3357814>
- [10] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. XDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (London, United Kingdom) (KDD '18)*. New York, NY, USA, 1754–1763. <https://doi.org/10.1145/3219819.3220023>
- [11] Feng Liu, Wei Guo, Huifeng Guo, Ruiming Tang, Yunming Ye, and Xiuqiang He. 2020. Dual-Attentional Factorization-Machines Based Neural Network for User Response Prediction. In *Companion Proceedings of the Web Conference 2020 (Taipei, Taiwan) (WWW '20)*. New York, NY, USA, 26–27. <https://doi.org/10.1145/3366424.3382680>
- [12] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.
- [13] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-Based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (Virtual Event, Ireland) (CIKM '20)*. New York, NY, USA, 2685–2692. <https://doi.org/10.1145/3340531.3412744>
- [14] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-Based Neural Networks for User Response Prediction over Multi-Field Categorical Data. *ACM Trans. Inf. Syst.* 37, 1, Article 5 (Oct. 2018), 35 pages. <https://doi.org/10.1145/3233770>
- [15] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (Beijing, China) (CIKM '19)*. New York, NY, USA, 1441–1450. <https://doi.org/10.1145/3357384.3357895>
- [16] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-Based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (Boston, MA, USA) (DLRS 2016)*. New York, NY, USA, 17–22. <https://doi.org/10.1145/2988450.2988452>
- [17] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (Marina Del Rey, CA, USA) (WSDM '18)*. New York, NY, USA, 565–573. <https://doi.org/10.1145/3159652.3159656>
- [18] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17 (Halifax, NS, Canada) (ADKDD'17)*. New York, NY, USA, Article 12, 7 pages. <https://doi.org/10.1145/3124749.3124754>
- [19] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential Recommendation Via Personalized Transformer. In *Fourteenth ACM Conference on Recommender Systems (Virtual Event, Brazil) (RecSys '20)*. New York, NY, USA, 328–337. <https://doi.org/10.1145/3383313.3412258>
- [20] Weinan Xu, Hengxu He, Minshi Tan, Yunming Li, Jun Lang, and Dongbai Guo. 2020. Deep Interest with Hierarchical Attention Network for Click-Through Rate Prediction. New York, NY, USA, 1905–1908. <https://doi.org/10.1145/3397271.3401310>
- [21] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian McAuley. 2019. CosRec: 2D Convolutional Neural Networks for Sequential Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (Beijing, China) (CIKM '19)*. New York, NY, USA, 2173–2176. <https://doi.org/10.1145/3357384.3358113>
- [22] Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. DeepIntent: Learning Attentions for Online Advertising with Recurrent Neural Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Francisco, California, USA) (KDD '16)*. New York, NY, USA, 1295–1304. <https://doi.org/10.1145/2939672.2939759>
- [23] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiuis Chen, and Jun Gao. 2018. ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). <https://ojs.aaai.org/index.php/AAAI/article/view/11618>
- [24] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.
- [25] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction (KDD '18). New York, NY, USA, 1059–1068. <https://doi.org/10.1145/3219819.3219823>
- [26] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning Tree-Based Deep Model for Recommender Systems. New York, NY, USA, 1079–1088. <https://doi.org/10.1145/3219819.3219826>