Adversarial Filtering Modeling on Long-term User Behavior Sequences for Click-Through Rate Prediction

Xiaochen Li
Alibaba Group
Beijing, China
xingke.lxc@alibaba-inc.com

Jian Liang
Alibaba Group
Beijing, China
xuelang.lj@alibaba-inc.com

Xialong Liu
Alibaba Group
Beijing, China
xialong.lxl@alibaba-inc.com

Yu Zhang
Lazada Group
Beijing, China
daoji@lazada.com

ABSTRACT

Rich user behavior information is of great importance for capturing and understanding user interest in click-through rate (CTR) prediction. To improve the richness, collecting long-term behaviors becomes a typical approach in academy and industry but at the cost of increasing online storage and latency. Recently, researchers have proposed several approaches to shorten long-term behavior sequence and then model user interests. These approaches reduce online cost efficiently but do not well handle the noisy information in long-term user behavior, which may deteriorate the performance of CTR prediction significantly. To obtain better cost/performance trade-off, we propose a novel Adversarial Filtering Model (ADFM) to model long-term user behavior. ADFM uses a hierarchical aggregation representation to compress raw behavior sequence and then learns to remove useless behavior information with an adversarial filtering mechanism. The selected user behaviors are fed into interest extraction module for CTR prediction. Experimental results on public datasets and industrial dataset demonstrate that our method achieves significant improvements over state-of-the-art models.

CCS CONCEPTS

• **Information systems** → **Online advertising**.

KEYWORDS

Click-Through Rate Prediction; User Behavior Modeling; Long-Term User Behavior

ACM Reference Format:
Xiaochen Li, Jian Liang, Xialong Liu, and Yu Zhang. 2022. Adversarial Filtering Modeling on Long-term User Behavior Sequences for Click-Through Rate Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3477495.3531788

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531788

1 INTRODUCTION

Click through rate prediction plays a vital role in recommender system and online advertising. Due to the rapid growth of user historical behavior data, user behavior modeling has been widely adopted in CTR prediction, which focuses on capturing the dynamics of user interest from user historical behaviors [1, 12, 14–16]. These models are mainly designed for short-term behavior sequence with limited length (e.g., less than 100). A natural extension is expanding the time window of historical behavior and incorporating more information. The length can be tens of thousands whereas the cost of storage and latency becomes rather high, especially for large-scale recommender system. How to design an efficient model for long-term user behavior has become a great challenge.

A straightforward solution is transforming long sequence to short sequence and applying classical user behavior modeling approaches directly. There has been some recent works trying to shorten long sequence [8, 9, 11]. MIMN [8] proposes a memory network-based model for long sequential user behavior modeling. It maintains a fixed-length memory and incrementally updates the memory when new behavior arrives. A similar model is HPMN [11] which uses memory network in lifelong sequential modeling. Memory network-based models have great advantages in saving storage and latency cost but fail to precisely capture user interest given a specific target item since encoding all historical behaviors into a fixed-size memory introduces massive noise. To overcome the limitations, SIM [9] designs a search-based model to selects relevant behaviors by hard search or soft search. Hard search selects sequence behaviors belonging to the category of target item. Soft search uses maximum inner product search to retrieval sequence behaviors similar to target item based on embedding vectors of items. SIM performs better than MIMN but at the risk of increasing online storage. Moreover, the selection strategies of SIM may not accurately identify relevant behaviors. For hard search, it is a rule-based strategy which causes information loss and may not remove noisy information efficiently. For soft search, it relies on embedding vectors of items to calculate the similarity between target item and sequence behaviors. However, embedding vectors are mainly learned for CTR prediction task and the similarity does not necessarily imply the relevance of sequence behaviors. The experimental results also show that soft search only performs slightly better than hard search [9]. UBR [10] organizes user history behaviors into a

inverted index and generates search queries to retrieve relevant behaviors. It ranks the candidate behaviors using BM25 and feeds selected behaviors to an attention-based CTR model. Although interesting, the ranking function in UBR is independent with CTR model, which can not be end-to-end optimized and may yield sub-optimal performance in retrieving relevant behaviors. In summary, these models reduce online cost but does not well handle the noisy information in long-term user behavior, which may deteriorate the performance significantly.

To better understand the problem, we carefully examine cases of long-term user behavior sequence and find that there are two types of noise, including duplicate behaviors and useless behaviors. Duplicate behaviors can be popular items, brands, shops that user visits multiple times. Keeping duplicate behaviors does not bring much new information but makes the limited-length sequence dominated by several hot items. Useless behaviors can be items clicked accidentally or long-tail items. These behaviors may not reflect user recent interests and have little value in online prediction.

We propose a novel Adversarial Filtering Model to remove duplicate and useless behavior from long-term user behavior sequences with low cost of storage and latency. In specific, we first use a hierarchical aggregation representation to group duplicate behaviors, score and select the top-k useful behaviors, and then use multi-head attention to extract user interest from selected sequences. We also propose an adversarial filtering mechanism to encourage the selection of useful behaviors.

The rest of the paper is organized as follows: Section 2 introduces our ADFM model. Section 3 presents experimental results on public datasets and industrial dataset. Section 4 concludes the paper.

2 THE PROPOSED APPROACH

In this Section, we first introduce a base CTR model as benchmark. Then we detail model structure and optimization strategy of ADFM.

2.1 Base CTR model

Input features. The features used in base CTR model include: (1) Item profile: Item id and its side information (e.g., brand id, shop id, category id). (2) User profile: User id, age, gender and income level. (3) Short-term user behavior: For each user $u \in U$, there are several historical behavior sequences $s^u_{t,v} = [b^1_{t,v}, ..., b^i_{t,v}, ..., b^n_{t,v}]$, where $b^i_{t,v}$ is the i-th behavior with behavior type $t \in \{impression, click, add\ to\ cart, pay\}$ and behavior target $v \in \{item, brand, shop, category\}$. The value of each behavior $b^i_{t,v}$ is the id of its behavior target, e.g., item id. The sequence is sorted by occurrence time of $b^i_{t,v}$. The time window is 3 days and the length does not exceed 100. (4) Long-term user behavior: The representation is the same as that of short-term one. The time window can be several months and the length can be tens of thousands.

Embedding layer. Embedding layer is a common operation to transform high-dimensional sparse features to low-dimensional dense embedding. For sparse feature f_i, the corresponding embedding dictionary is denoted as $E_i = [e^1_i, ..., e^j_i, ...e^{N_i}_i] \in R^{D \times N_i}$, where $e^j_i \in R^D$ is an embedding vector, D is the embedding dimension, N_i is the number of sparse features. Embedding layer looks up the embedding table and produces a list of embedding vectors. If f_i is a one-hot vector with the j-th element $f_i[j] = 1$, the embedding vector list is $\{e^j_i\}$. If f_i is a multi-hot vector with $f_i[j] = 1$ for $j \in \{i_1, i_2, ..., i_k\}$, the embedding vector list is $\{e^{i_1}_i, e^{i_2}_i, ..., e^{i_k}_i\}$.

Pooling layer. The embedding vector list is fed into pooling layer to get a fixed-length embedding vector. Sum pooling is adopted in our work.

MLP layer. All the embedding vectors of input features are concatenated and fed into MLP layer. MLP layer is used to learn the nonlinear interaction between features. The output of MLP layer is the probability of target item being clicked.

Loss function. Negative log-likelihood function is used:

$$L = -\frac{1}{N} \sum_{(x,y) \in S} (ylogp(x) + (1-y)log(1-p(x))) \quad (1)$$

where S is the training set of size N, x is input features, $y \in \{0, 1\}$ is click label and $p(x)$ is the predicted CTR.

2.2 The ADFM model

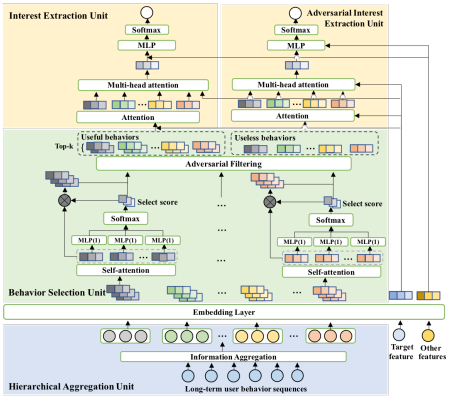

Figure 1: The structure of ADFM. i) Hierarchical aggregation unit aggregates raw sequences to remove duplicate behaviors. ii) Behavior selection unit refines each behavior and separates useful and useless behaviors. iii) Interest extraction unit and adversarial interest extraction unit capture user interests from useful and useless behaviors respectively.

The aim of ADFM is to identify duplicate and useless behaviors from the long-term user behavior sequence and retain the most useful k behaviors, where k is a hyper-parameter. Based on the filtered sequences, user's long-term interests are extracted and fed into the CTR model. The model structure of ADFM is shown in Figure 1 and the main modules are as follows:

Hierarchical Aggregation Unit (HAU). HAU uses a hierarchical aggregation representation to compress duplicate behaviors in raw sequences. For each sequence, it groups behaviors by their value (e.g., item id) and only keeps the unique behaviors. The representation is essentially a set instead of a sequence. Keeping unique behaviors compresses raw sequences significantly whereas it may cause the loss of sequential information and can not distinguish between the behaviors occurred once and multiple times. To preserve the original information as much as possible, we add two aggregation statistics (i.e., occurrence number and the maximum timestamp of behavior) and transform them into

sparse features by binning. Each behavior is represented as a tuple $< behavior\_id, number, timestamp >$. Take a sequence with behavior type *click* and behavior target *shop* for example, the aggregated sequence can be:

$$s^u_{click,shop} = [< shop\_id_1, click\_num_1, timestamp_1 > , ..., < shop\_id_m, click\_num_m, timestamp_m >] \quad (2)$$

Behavior Selection Unit (BSU). The sequences aggregated by HAU will be fed into embedding layer and output the embedding vectors of the sequences $A^u_{t,v} = (e^1_{t,v}, ..., e^i_{t,v}, ..., e^m_{t,v})$, where m is the length of aggregated sequences. Here we stack $e^i_{t,v} \in R^D$ together into a matrix $E^u_{t,v} \in R^{m \times D}$. There have been no duplicate behaviors in aggregated sequences but still have a lot of useless behaviors. BSU aims to select top-k useful behaviors from aggregated sequences.

Firstly, to capture the interactions between behaviors in aggregated sequence, we use the self-attention mechanism [3, 13] to obtain a new embedding vector for each behavior:

$$\begin{aligned} E^{u,att}_{t,v} &= Attention(E^u_{t,v}W^Q, E^u_{t,v}W^K, E^u_{t,v}W^V) \\ &= Softmax(\frac{E^u_{t,v}W^Q(E^u_{t,v}W^K)^T}{\sqrt{|D|}})E^u_{t,v}W^V \end{aligned} \quad (3)$$

where W^Q, W^K and W^V are linear matrices.

$E^u_{t,v}$ is then fed into a scoring gate and output selection scores $G_{t,v}$, which reflects the importance of sequence behaviors.

$$G_{t,v} = F(E^{u,att}_{t,v}) \quad (4)$$

where $F(.)$ is implemented as a feed forward neural network which is jointly trained with CTR model. We multiply $E^{u,att}_{t,v}$ by $G_{t,v}$ to adjust embedding vector of each behavior and obtain the embedding vectors of selected behaviors by:

$$E^{u,s}_{t,v} = Filter(G_{t,v},\ G_{t,v}E^{u,att}_{t,v},\ k), \quad (5)$$

where the function $Filter(score, embedding, k)$ sorts sequence behaviors by score and selects top-k behaviors.

Interest Extraction Unit (IEU). IEU adopts multi-head attention [3, 13] to extract user interest from the top-k behaviors:

$$head_i = Softmax(\frac{E_{target}W^C(E^{u,s}_{t,v}W^S)^T}{\sqrt{|D|}})E^{u,s}_{t,v}W^E \quad (6)$$

where W^C, W^S, W^E are linear matrices, E_{target} represents the embedding of target item and $head_i$ is the ith head in multi-head attention. The long-term user interest is represented as $concat(head_1; ...; head_n)$ and then fed into MLP layer for CTR prediction. The CTR loss based on the top-k behaviors is denoted as $Loss_s$ and calculated by Eq. (1).

Ideally, CTR model can propagate gradient back to BSU and adjust selection score, which can guide the training of BSU. However, the optimization goal of model loss is the accuracy of CTR prediction, not the usefulness of the behaviors selected by BSU. It is hard to guarantee that the selected behaviors are top k most useful. To supervise the learning of BSU more directly, we propose an adversarial filtering mechanism, which consists of an Adversarial Interest Extraction Unit (AIEU) and an adversarial training schema.

Adversarial Interest Extraction Unit. The input of this module is the remaining m-k behaviors, and the model structure is the same as that of IEU. Since the distribution of useful behaviors and useless behaviors are quite different, the parameters of AIEU are not shared with IEU. AIEU aims to identify potential user interests from the remaining m-k actions and feed the interests into MLP layer for CTR prediction. The CTR loss based on the remaining m-k behaviors is denoted as $Loss_{\overline{s}}$. Notice that AIEU is only used in offline training while IEU is used in both training and inference.

If the top-k behaviors selected by BSU are always useful, $Loss_s$ should be small and $Loss_{\overline{s}}$ should be large. We optimize the following minimization problem:

$$\min\ loss_s - loss_{\overline{s}}. \quad (7)$$

The loss will force BSU to select the top-k useful behaviors while the relatively useless m-k behaviors flow into AIEU.

There is a problem in directly optimizing this loss: When the parameters of BSU update during model training, the distribution of the top-k and the remaining m-k behaviors may change drastically, which makes it difficult for IEU and AIEU to converge. The convergence of the CTR loss becomes difficult meanwhile. Inspired by the success of adversarial learning [2, 4, 6], we propose an adversarial training schema to accelerate the optimization of ADFM.

We first fix BSU and learn IEU and AIEU with the goal of extracting user interests from both the top-k behaviors and the remaining m-k behaviors. The optimization problem is defined as follows:

$$\min_{IEU,AIEU}\ Loss_s + Loss_{\overline{s}}, \quad (8)$$

Then we fix IEU and AIEU and optimize BSU with the following minimization problem:

$$\min_{BSU}\ Loss_s - Loss_{\overline{s}}. \quad (9)$$

The training process loops until it satisfies the stopping criterion (e.g., the changes of $Loss_s$ and $Loss_{\overline{s}}$ are small).

3 EXPERIMENTS

In this Section we present the experimental setup and conduct experiments to evaluate the performance and cost of our model.

3.1 Experimental setup

Datasets. Two public datasets and an industrial dataset are used:

Amazon dataset[1]. It is a commonly used benchmark, which is composed of product reviews and meta-data from Amazon [7]. We use the Books category of Amazon dataset, including 51 million records, 1.5 million users and 2.9 million items from 1252 categories. We treat review as click behavior. For long-term use behavior model, we filter the samples whose sequence length is shorter than 20.

Taobao dataset[2]. It is a collection of user behaviors from Taobao's recommender system [18] and contains 89 million records, 1 million users and 4 million items from 9407 categories. We only use click behaviors of each user.

Industrial dataset. It is collected from our online advertising system in April 2021. The data collector has signed a contract with each user and obtained their consent to use the data. The dataset contains 4.3 billion records, 11.5 million users, 16.4 million items

[1] https://nijianmo.github.io/amazon/index.html
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

Table 1: Results on public datasets and industrial dataset

Models	Amazon	Taobao	Industrial dataset		
	AUC	*AUC*	*AUC*	$GAUC_{user}$	$GAUC_{req}$
DNN	0.8355	0.8714	0.7001	0.6354	0.6083
DIN	0.8386	0.8804	0.7019	0.6367	0.6094
DIEN	0.8442	0.9032	0.7034	0.6379	0.6106
ComicRec	0.8458	0.9012	0.7037	0.6381	0.6109
MIMN	0.8494	0.9144	0.7046	0.6392	0.6131
SIM	0.8461	0.9360	0.7047	0.6390	0.6128
ADFM	**0.8574**	**0.9462**	**0.7094**	**0.6428**	**0.6161**

Table 2: Ablation study

Model	AUC
DNN	0.7001
DNN_{long}	0.7028
$DNN_{long} + HAU$	0.7035
$DNN_{long} + HAU + BSU$	0.7061
$DNN_{long} + HAU + BSU + adversarial\ learning(ADFM)$	**0.7094**
$DNN_{long} + HAU + ComicRec$	0.7074

from 5000 categories. Each record contains historical behavior sequences with several behavior types (e.g., impression, click, add to chart and pay) from the preceding 180 days.

Competitors. We compare ADFM with these competitors:

- **DNN** is a base DNN model presented in Section 2.1.
- **DIN** [16] proposes an attention mechanism to represent the user interests w.r.t. candidates.
- **DIEN** [15] uses GRU to model user interest evolution.
- **ComicRec** [1] proposes a comprehensive framework which integrates the controllability and multi-interest components.
- **MIMN** [8] proposes a memory network-based model to capture multiple channels of user interest drifting for long-term user behavior modeling.
- **SIM** [9] is a search-based user interest model for long-term user behavior modeling. We will only compare the hard-search strategy as it is adopted in their online system.

Among these models, DNN, DIN, DIEN, ComicRec are designed for short-term user behavior, and MIMN, SIM are for long-term.

Parameter Configuration. For short-term user behavior models, the maximum sequence length is set to 20 and 100 in public datasets and industrial dataset respectively. For long-term ones, all historical behaviors are used. For ADFM, the number of selected behaviors is 20 and 100 in public datasets and industrial dataset respectively. MLP shape is 256 * 128 * 64 and embedding dimension D is 16. . The head number in multi-head attention is set to 2. Optimization algorithm is Adam [5] with learning rate 0.001.

Evaluation Metrics. We use AUC, $GAUC_{user}$ and $GAUC_{req}$ to evaluate these models. AUC is a widely used metric to measure model effectiveness. $GAUC_{user}$ and $GAUC_{req}$ is group AUC [17] calculated at the level of user and request respectively. We only use GAUC in industrial dataset since the number of groups is relatively small in public datasets.

3.2 Performance evaluation

As shown in Table 1, MIMN, SIM and ADFM outperform DNN, DIN, DIEN and ComicRec since long-term user behavior sequence brings much new information. ADFM outperforms the other long-term user behavior models with a significant AUC gain of at least 0.1 in Amazon and Taobao dataset. ADFM also achieves an AUC gain of 0.0047, a $GAUC_{user}$ gain of 0.0038 and a $GAUC_{req}$ gain of 0.0033 over the best competitor SIM in industrial dataset. In online advertisement systems with huge traffic, even 0.001 absolute AUC gain is a significant improvement [16].

We conduct online A/B testing experiments to evaluate our model in our advertising system. The experiment lasts for 12 days and ADFM achieves 4.7% CTR and 3.1% RPM gain compared to SIM. To deploy ADFM, we keep the top-k behaviors selected by BSU and upload them to online cache. Online system fetches the selected sequences per user and feeds them to ADFM for CTR prediction. For SIM, we store user-category level behavior sequences in online cache and look up one sequence for each candidate item.

3.3 Storage and latency cost

During online inference, ADFM only stores the top-k behaviors online and storage cost is proportional to k. We analyze the contribution of HAU and BSU on cutting storage cost in industrial dataset. HAU compresses sequence length by 60% and BSU reduces length by 27% further. For online latency, MIMN maintains a fixed-length memory and the latency is nearly the same as that of ADFM if memory size equals to k. As sequence length increases, the latency of ADFM remains constant whereas SIM continues to increase.

3.4 Ablation study

We conduct an ablation study on industrial dataset to evaluate the effect of key modules of ADFM. As shown in Table 2), DNN_{long} is feeding long-term sequences into base CTR model and processing by sum-pooling. It is no surprise that it outperforms DNN by introducing new information. $DNN_{long} + HAU$ performs slightly better than DNN_{long} and demonstrates the effectiveness of hierarchical aggregation representation. Introducing BSU for selecting useful behaviors brings great improvements and adversarial learning strengthens the power of BSU by providing more supervision information for the learning of BSU. In addition, we evaluate the performance of *ComicRec* on top of $DNN_{long} + HAU$. The AUC is higher than that of $DNN_{long} + HAU + BSU$ but less than that of ADFM, which indicates that BSU may be hard to optimize and adversarial learning alleviates the problem.

4 CONCLUSION

In this paper, we propose a novel adversarial filtering model on long-term user behavior sequences. We use a hierarchical aggregation representation to remove duplicate behaviors. To filter useless behaviors, our model scores and selects top-k useful behaviors with the help of an adversarial filtering mechanism. The results of offline and online A/B experiments demonstrate that our model achieves significant improvements over the competitors.

REFERENCES

[1] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 2942–2951.

[2] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2019. A game theoretic approach to class-wise selective rationalization. In *Advances in Neural Information Processing Systems*. 10055–10065.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/ARXIV.1810.04805

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc.

[5] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[6] Jian Liang, Bing Bai, Yuren Cao, Kun Bai, and Fei Wang. 2020. Adversarial infidelity learning for model interpretation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 286–296.

[7] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 188–197.

[8] 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*. Association for Computing Machinery, New York, NY, USA, 2671–2679.

[9] 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*. Association for Computing Machinery, 2685–2692.

[10] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User behavior retrieval for click-through rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 2347–2356.

[11] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. 2019. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 565–574.

[12] 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)*. Association for Computing Machinery, New York, NY, USA, 1441–1450.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[14] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4695–4703. https://doi.org/10.24963/ijcai.2021/636 Survey Track.

[15] 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.

[16] 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. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[17] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized cost per click in Taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 2191–2200.

[18] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 1079–1088.