Hindawi
Scientific Programming
Volume 2023, Article ID 1217898, 14 pages
https://doi.org/10.1155/2023/1217898

Research Article

Feature Dual Supervision Model for the Searches of Online Advertising Audiences

Haipeng Ni and Zhixi Wang

School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China

Correspondence should be addressed to Zhixi Wang; zhixiwang@163.com

Received 4 January 2023; Revised 26 February 2023; Accepted 21 April 2023; Published 11 May 2023

Academic Editor: Sadiq Hussain

Online advertising has become one of the most important strategies used by companies. They get the valuable results from Internet marketing and communication strategies. Therefore, it is necessary to study the click-through rate (CTR) model to search the potential audiences in online advertising. The advertisers desire to search for potential candidates through a large number of queries for audiences in programmatic advertising. Facing such a large corpus, the most common method is that using two-tower model to learn user's queries and ad representations, and then the similarity function is applied to match the feature representation to get the potential audiences related to the ad. However, in the process of feature extraction, there is a lack of information interaction between the two towers, resulting in the loss of details in the representation. In order to alleviate the lack of information interaction between the networks in the two-tower model during feature extraction. In this paper, we propose a novel model named Feature Dual Supervision Model (FDSM), which integrates by Feature Expression Unit (FEU) and Feature Supervision Unit (FSU). The FEU is used to extract ads or users features, and FSU generates a weight vector to supervise the working process of the FEU. In addition, we propose a feature cross-layer with bridge connections in FDSM to achieve effective feature interaction between ad and user representations. Finally, we conduct experiments on the Tencent Lookalike and MovieLens datasets. The experimental results indicate that the FDSM model outperforms other state-of-the-art CTR prediction models in audience expansion.

1. Introduction

With the rapid development of the Internet, online advertising provides a common marketing experience when people are accessing services using intelligent devices. Online advertising refers to advertisements displayed in media [1]. Different from traditional advertising, it has formed a crowd as the target, product-oriented technology delivery model. For given an ad and its historical audience (seed users), audience expansion aims to find potential audiences that are similar to the seed users for the ad. For example, using the user's searched keywords, topics, history of visit behaviors, interests, and so on the programmatic advertising system can accurately find the potential audiences for an ad through audience targeting technology.

Programmatic advertising (PA) refers to a kind of advertising form that applies technology to serve advertising trading and management in big data field. Advertisers can programmatically purchase media resources, applying algorithms and technologies to automatically achieve precise target audiences [2]. The technology of PA analyzes millions of ads data in real time, which enabling PA ads to accurately reflect the interests of users at the exact moment, and they are most likely to click on an ad. Therefore, programmatic advertising is a new marketing technology through the Internet and emerging technologies.

In the programmatic advertising system or recommendation system, click-through rate (CTR) is an important metric, which is defined to forecast the probability that a user will click a display ad or recommended item on web page [3]. Then the system determines whether the ad or item will

display on the user's page basing on this metric. In online advertising, the prediction result of CTR has a great influence on the effect of online advertising. Therefore, the accuracy of CTR prediction is a key factor affecting the effect of advertising, user experience and platform revenue.

In order to improve the performance of the CTR model, effective feature cross is the most commonly applied optimization method. Early studies focused on designing and utilizing effective combination features, such as FM and FFM [4]. These models utilize expert experience to explore clear interactions between features, which inevitably cost a lot of labor in the industry. However, the current large-scale recommendation contains a large number of original features and potential high-order interaction features, which makes it difficult for expert-experienced feature engineering to comprehensively cover all interaction patterns in feature space, thus limiting the application of shallow models in the industry.

In nearly a decade, a large number of CTR prediction models based on deep learning are applied to explore higher-order implicit information in feature space. In this paper, we focus on optimizing the performance of the two-tower model. This model was first applied in the domain of NLP. The typical architecture is DSSM [5]. The input of this model is a high-dimensional term vector about a query or a document. Then, the DSSM passes its input through two neural networks with two different inputs, respectively, and maps them into semantic vectors in a shared semantic space. For Web document ranking, DSSM computes the relevance score between a query and a document with the cosine similarity function and ranks documents by their similarity scores to the query.

Despite great promise, there are still some problems in two-tower model. Since the feature vectors of the user's query and ad separately are fed into two different neural networks in the online retrieval service, and generating the highly concentrated vector representation, which leads to some detail information loss and suffers from a lack of information interaction between the two towers. In order to overcome this shortcoming, we propose a Feature Dual Supervision Model (FDSM) based on the two-tower model to enhance feature extraction capability and provide more fine-grained information at the feature cross-layer. Its network structure is summarized as follows:

Users/Ads Feature Expression: During the process of extracting features in the user/ad tower, a feature expression unit is applied, which is made of multiple neural networks, and the structure among of them can be the same or different, but the dimension of the output vector should be the same. At the same time, the proposed of a feature supervision unit to monitor the process of feature extraction. Specifically, the feature vector of user/ad will feed in feature expression unit and get multiple representations correspondingly, and then the supervision unit will give a score for every representation. Finally, the unique expression feature is obtained basing on all the representations and the scores. In this paper, the fully connected network is regarded as the feature expression network.

Feature Cross-Layer: As in the ordinary two-tower model, the feature interaction between the extracted representations of user and the ad is required in this layer. In this paper, with the difference that the degree of match performed by the cosine function, it is no longer applied. A bridge connection module is proposed in this paper to combine the ad and user expression vectors, which are then fed to the network for feature interaction to perform CTR prediction.

The main contributions of this work are summarized as follows:

(i) We propose a novel Feature Dual Supervision Model (FDSM), which can enhance the feature extraction ability of users and ads information and obtain features with high performance expression.

(ii) In the feature cross-layer, a bridge connection module is proposed to connect the extracted features, which can achieve feature interaction well, so as to improve the prediction performance of CTR for FDSM.

(iii) We conduct experiments on two real datasets. The experimental results have demonstrated that with feature supervision unit and cross-layer with bridge connection module, FDSM outperforms other state-of-the-art CTR prediction models in audience expansion system.

This paper is organized as follows: Section 2 introduces the mainstream model of CTR and its development context. Section 3 illustrates the design details of FDSM model proposed in this paper. Section 4 shows the details and results of the experiment. Section 5 summarizes the paper and prospects for the future work.

2. Related Works

In this section, general models related to CTR are summarized and introduces models about semantic matching in the NLP domain and then illustrates promotion applications of the two-tower model in the system of programmatic advertising and recommendation.

Early research focused on the design and utilization of effective combinatorial features, such as FM [4] and FFM [6]. These models mainly exploit expert experience on exploring explicit interaction between features. In recent years, CTR prediction models based on deep learning have emerged to explore higher-order implicit information in feature space. Deep learning-based CTR prediction models follow the pattern of "feature embedding & feature interaction." The representative models include Wide&Deep [7], DeepFM [8], DCN [9], PIN [10], DIN [11], PNN [12], and the two-tower model [5], which jointly learn explicit and implicit feature interaction and finally output matching information.

With the application of deep learning in natural language, many neural network models have been proposed to address semantic matching problems. These approaches are divided into two categories: representation-based learning

and interaction-based learning. The models with a two-tower structure are typical characteristics of representation-based approaches, such as DSSM [5], CLSM [13], LSTM-RNN [14], and ACR-I [15]. Each tower uses a different neural network to generate a semantic representation of the query or document. A matching function, such as inner product, is then applied to measure the similarity between the metric query and the document. The interaction-based approaches learn the complicated relevance patterns between queries and documents. The mainstream models are MatchPyramid [16], Match-SRNN [17], DRMM [18], and K-NRM [19].

In the field of advertising and recommendation, the MV-DNN [20] extend the two-tower to jointly learn from features of items from different domains and user features by introducing a multiview deep learning model, which can learn the user's behavior patterns according to the rich user behavior features and improve the user experience on the web service. In advertising display system, Baidu proposed MOBIUS [21], which base on two-tower, to maximize CPM and reduce the difference between ranking and matching in the retrieval stage. However, the two-tower model suffers from a lack of information interaction between the respective towers as well as the imbalance of category data affects the performance of the model. Therefore, the DAT model not only customizes an augmented vector for each query and item to mitigate the lack of information interaction, but also proposes category alignment loss to align the item representation of uneven categories.

3. Methodology

In this section, we first define the problem of audience targeting and CTR prediction, then illustrate our proposed model in detail.

3.1. Problem Formulation. Given a seed set S, and a candidate set C, audience targeting aims to extend S via selecting n users T from C (usually $|S| \ll |C|$), such that the potential users T are similar to S. In this problem, each user u is usually represented by a low-dimensional dense vector that encodes the information of users' demographic profiles and online behaviors [22]. In order to search similar users based on a seed set, we apply the CTR prediction methods.

As a binary classification task, CTR represents a probability whether a user will click an ad campaign or an item displayed online system. Specifically, for given a training set containing N samples (X, y), we indicate the input of a model as $X = \{x_1, x_2, \ldots, x_f\}$, which contains f features. X includes user features as well as ad features. All the features could be not only categorical, such as gender or occupation, but also continuous, such as the price about an item. $y \in \{0, 1\}$ is the label of a sample, where $y = 1$ indicates that the user with positive feedback for an ad campaign, such as clicking on the advertisement, purchasing the product or downloading the APP, otherwise $y = 0$. Therefore, the CTR prediction model calculates the probability $P(y = 1|X)$ for each instance X. Table 1 shows the notations in this paper.

3.2. The Details of Feature Dual Supervision Model. As shown in Figure 1, the overall framework of our proposed Feature Dual Supervision Model (FDSM) in this paper, which includes three modules: the feature embedding layer, the feature expression layer, and feature cross-layer. The embedding layer transforms the instance X into a low-dimensional dense vector. The feature expression layer extracts efficient feature representations of users and ads features, respectively. The purpose of the cross-layer is to discover relationships between features, which predict the probability of CTR about whether the user will click the ads.

3.2.1. Embedding Layer. The CTR prediction model based on deep learning follows the "feature embedding & feature interaction" paradigm [23]. The embedding module embeds each feature for an instance to a d-dimensional embedding vector. For the ith filed, the feature embedding vector can be obtained from the embedding lookup table as follows:

$$e_i = \mathbf{E}_{i,x_i}, \tag{1}$$

where e_i is the embedding vector, x_i denotes the ordinal encoding of the ith field about instance X. $\mathbf{E}_i \in R^{S_i \times d}$ is the embedding matrix, and S_i, d are the size of the lookup table for the ith field and embedding size, respectively. If the field is multivalent, the mean pooling of feature embedding as the field embedding representation:

$$e_i = \frac{\mathbf{E}_{i,x_{i1}} + \mathbf{E}_{i,x_{i2}} + \cdots + \mathbf{E}_{i,x_{\text{in}}}}{n}, \tag{2}$$

where n is the number of feature value in the ith field. Therefore, we denote the output of embedding layer for a instance X, which contains f feature fields, as the embedding matrix as follows:

$$\mathbf{E}_X = \left[e_1, e_2, \ldots, e_f\right]. \tag{3}$$

In this work, we divide an instance X into two parts according to the characteristics of user and ad, denoted as $X^u = \{x_1, x_2, \ldots, x_\mu\}$ and $X^a = \{x_{\mu+1}, x_{\mu+2}, \ldots, x_{\mu+\nu}\}$, respectively, where $\mu + \nu = f$, as shown in the left part in Figure 1. The corresponding embedding representations of user and ad are obtained through the embedding layer as

$$\begin{aligned} \mathbf{E}_{X^u} &= \left[e_1, e_2, \ldots, e_\mu\right], \\ \mathbf{E}_{X^a} &= \left[e_{\mu+1}, e_{\mu+2}, \ldots, e_{\mu+\nu}\right]. \end{aligned} \tag{4}$$

3.2.2. Feature Expression Layer. In this part, we illustrate the symbols first. The initial representations of user and ad in embedding matrix are concatenated and the mean pooling is expressed as follows:

Table 1: Notations for the proposed model FDSM.

Symbol	Definition
X	The instance contains user and ad characteristics on dataset
$\mathbf{E}_X$	The embedding matrix of instance X
e_i	The embedding vector of the ith field about instance X
X^u/X^a	The characteristics of user u's and ad a's
$\mathbf{E}_{X^u}/\mathbf{E}_{X^a}$	User u's embedding matrix and ad a's embedding matrix
e^u/e^a	User u's embedding vector and ad a's embedding vector
$e^{\overline{u}}/e^{\overline{a}}$	User u's mean pooling vector and ad a's mean pooling vector
$e^{u\overline{a}}/e^{a\overline{u}}$	Ad a's supervision vector and user u's supervision vector
$\mathbf{h}_{L_i}$	The output from the last hidden layer in the ith fully connected network
$\mathbf{A}^u/\mathbf{A}^a$	User u's multirepresentation and ad a's multirepresentation from FEU
$\mathbf{w}^u/\mathbf{w}^a$	User u's supervised weight vector and ad a's supervised weight vector
$\mathbf{I}^u/\mathbf{I}^a$	User u's feature expression vector and ad a's feature expression vector
$\mathbf{I}^b$	The vector from bridge connection unit
$\widehat{y}$	The prediction of CTR from FDSM

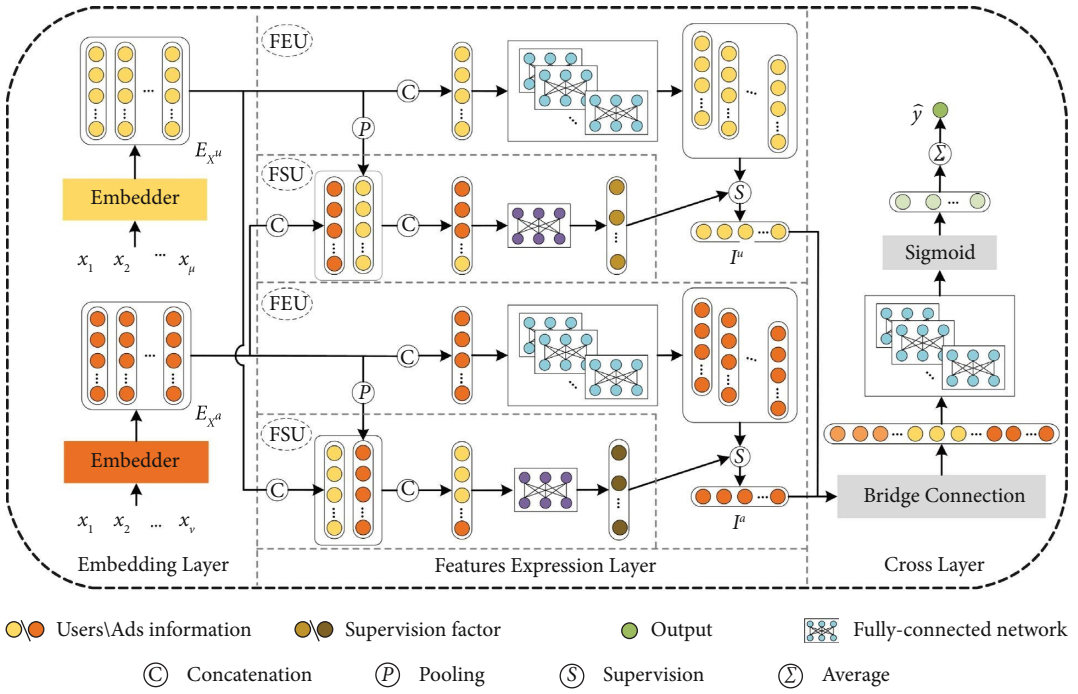

Figure 1: The overview of our proposed model FDSM with FEU and FSU.

$$e^u = e_1 \oplus e_2 \oplus \cdots \oplus e_\mu; \quad e^{\overline{u}} = \frac{1}{\mu}\sum_{i=1}^{\mu} e_i,$$
$$e^a = e_{\mu+1} \oplus e_{\mu+2} \oplus \cdots \oplus e_{\mu+\nu}; \quad e^{\overline{a}} = \frac{1}{\nu}\sum_{i=\mu+1}^{\mu+\nu} e_i, \tag{5}$$

where "$\oplus$" is vector concatenation operation, and e^u, $e^{\overline{u}}$ are the user u embedding vector and mean pooling vector, equally, e^a, $e^{\overline{a}}$ are the ad a embedding vector and mean pooling vector. Thus, the vector of e^u and $e^{\overline{a}}$ can be concatenated together as the ad supervision vector $e^{u\overline{a}}$, the vector of e^a and $e^{\overline{u}}$ can be concatenated together as the user supervision vector $e^{a\overline{u}}$, which as follows:

$$e^{u\overline{a}} = e^u \oplus e^{\overline{a}}; \quad e^{a\overline{u}} = e^a \oplus e^{\overline{u}}, \tag{6}$$

where $e^{u\overline{a}} \in R^{(\mu+1)d}$, $e^{a\overline{u}} \in R^{(\nu+1)d}$.

Users feature expression: For the part of user feature expression is shown in the middle part of Figure 1, which consists of the Feature Expression Unit (FEU) and the Feature Supervision Unit (FSU), where FEU is responsible for the extraction of user information from the user embedding vector e^u, while FSU supervises the process of user information extraction basing on the user supervision vector $e^{a\overline{u}}$. Finally, the highly condensed representation vector of user information will be obtained through both units. The details are as follows.

We can employ a unit FEU, which are made of the multiple fully connected networks; each network can extract the users' representations for the embedding vector independently. Generally, a single network only focuses on partial information during the process of extraction, which

cannot completely cover the characteristic about user. In order to address this challenge, multiple fully connected networks were applied to jointly discover users' implicit features jointly. A single deep fully connected network in FEU, with each deep layer having the following formula:

$$\mathbf{h}_l = \mathrm{ReLu}\left(\mathbf{W}_l^T \mathbf{h}_{l-1} + \mathbf{b}_l\right); \quad l = 1, 2, \cdots, \tag{7}$$

where $\mathbf{h}_{l-1} \in R^{d_{l-1}}$ and $\mathbf{h}_l \in R_l^d$ are the $(l-1)$-th and l-th hidden layer, respectively; $\mathbf{W}_l \in R^{d_{l-1} \times d_l}$ is the weight matrix for the layer from $(l-1)$th to lth; $\mathbf{b}_l \in R^{d_l}$ is bias vector for the l-th layer. In particularly, there is for the first layer, where $\mathbf{h}_0 = e^u$. Therefore, the user feature representation vector as output come from the last hidden layer in the ith fully connected network, and the matrix derive from the FEU unit with m fully connected networks can be summarized as follows:

$$\mathbf{h}_{L_i} = f\left(e^u\right), \tag{8}$$

$$\mathbf{A}^u = \left[\mathbf{h}_{L_1}, \mathbf{h}_{L_2}, \ldots, \mathbf{h}_{L_m}\right], \tag{9}$$

where $f(\cdot)$ denotes the fully connected network, the output vector of the ith network is the representation of the user's features, the subscript "L" denotes the last layer of the hidden layer in the fully connected network; and the matrix $\mathbf{A}^u \in R^{d_L \times m}$ denotes the output from the FEU with m networks.

As for the unit of FSU, which is composed of a single fully connected network, the input is the user supervision vector $e^{a\overline{u}}$. Different from the FEU, the activation function of ReLU is never applied in the last layer of a fully connected network, where the softmax activation function is applied, denoted as

$$\begin{aligned} \mathbf{h}_L &= \mathrm{softmax}\left(\mathbf{W}_L^T \mathbf{h}_{L-1} + \mathbf{b}_L\right), \\ \mathbf{w}^u &= \mathbf{h}_L = f\left(e^{a\overline{u}}\right), \end{aligned} \tag{10}$$

where the $\mathbf{w}^u \in R^m$ is the user supervised weight vector, which the dimensionality as the number of fully connected network in the FEU. The softmax activation function in the last layer normalizes the output into a probabilistic representation.

From the above exposition, it is clear that not only has the matrix $\mathbf{A}^u$, with m representation vectors, been derived from the FEU unit according to the user embedding representation, but also the m-dimensional user supervised weight vector $\mathbf{w}^u$ is obtained through the FSU unit based on the user supervised vector. Finally, the output from both units as materials, the process of supervision operation is as

$$\begin{aligned} \mathbf{I}^u &= \frac{1}{m}\sum_{i=1}^{m} \mathbf{h}_{L_i} \cdot w_i^u \\ &= \frac{1}{m}\left(\mathbf{h}_{L_1} \cdot w_1^u + \mathbf{h}_{L_2} \cdot w_2^u + \cdots + \mathbf{h}_{L_m} \cdot w_m^u\right) \\ &= \frac{1}{m}\left(\mathbf{h}_{L_1}, \mathbf{h}_{L_2}, \ldots, \mathbf{h}_{L_m}\right) \cdot \begin{pmatrix} w_1^u \\ w_2^u \\ \vdots \\ w_m^u \end{pmatrix} \\ &= \frac{1}{m}\mathbf{A}^u \mathbf{w}^u, \end{aligned} \tag{11}$$

where $\mathbf{I}^u \in R^{d_L}$ is the representation of the user's final form, in this paper, which is called the user feature expression vector after implementing supervision.

Ads feature expression: The method of extracting ad representation in this part is completely consistent with the way of user feature expression. Here, the input of FEU unit is ad embedding vector e^a, while the input of the FSU unit is an ad supervision vector $e^{u\overline{a}}$. Therefore, the output matrix, supervised weight vector, and ad's feature expression vector are as follows:

$$\begin{aligned} \mathbf{A}^a &= \left[\mathbf{h}_{L_1}, \mathbf{h}_{L_2}, \ldots, \mathbf{h}_{L_n}\right], \\ \mathbf{w}^a &= \mathbf{h}_L = f\left(e^{u\overline{a}}\right), \\ \mathbf{I}^a &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_{L_i} \cdot w_i^a = \frac{1}{n}\mathbf{A}^a \mathbf{w}^a. \end{aligned} \tag{12}$$

In this module, we set n as the number of fully connected networks in FEU unit, then the output matrix $\mathbf{A}^a \in R^{d_L \times n}$ hold n ad representation vectors, similarly, the dimension of ad supervised weight vector $\mathbf{w}^a \in R^n$ is n; $\mathbf{I}^a \in R^{d_L}$ is ad's feature expression vector.

According to the above introduction process, FEU unit is composed of multiple fully connected networks. FEU can extract features representation matrix $\mathbf{A}$ from the same user or ad based on equation (9). Different networks can focus on features of specific domains, but the number of feature tasks processed by multinetwork learning methods is limited [24, 25]. Therefore, FSU unit is used to generate supervised weight vector $\mathbf{w}$ to make comprehensive judgment of multiple feature representations. The specific calculation process is shown in equation (11). The FSU unit generates a decision weight w_i for every vector $\mathbf{h}_{L_i}$ in, and then $\mathbf{h}_{L_i} \times w_i$ is operated to obtain the evaluation vector. Finally, all vectors in $\mathbf{A}$ are operated in the same way with all supervisory factors from FSU, and the mean value of all evaluation vectors is calculated. In this way, FEU and FSU work together to enhance feature extraction and presentation in online advertising systems.

3.2.3. Feature Cross-Layer. Through the previous description, we obtained the feature expression vectors $\mathbf{I}^u$ and

$\mathbf{I}^a$ for user and ad. Both of them imply important information of the features, which can represent the information of user and ad more effectively. At this point, the feature cross-layer is designed to explore the relationship between a user and an ad, which plays an important part to obtained high performance of CTR prediction in ad service system. Its structure is shown in the right part of Figure 1. The expression vectors of user and ad are passed through the bridge connection module, and then the output vectors are fed to the multiple fully connected networks to achieve the prediction of user's click-through rate for a given an ad. In this paper, the operation of the bridge connection based on expressions is designed as follows:

$$\mathbf{I}^b = \left(\mathbf{I}^u \odot \mathbf{I}^a\right) \oplus \mathbf{I}^u \oplus \mathbf{I}^a, \tag{13}$$

where the operation "$\odot$" represents the Hadamard product, and the vector $\mathbf{I}^b \in R^{3d_L}$ is the output of the bridge connection unit.

Finally, we use k fully connected networks to form a feature cross-module. Similar to the feature expression, where the output of the last layer with a single neuron of each fully connected network in the feature cross-module is expressed as follows:

$$h_L = \text{Sigmoid}\left(\mathbf{W}_L^T \mathbf{h}_{L-1} + b_L\right), \tag{14}$$

where the value $h_L \in [0, 1]$ of the output node of the network represents the probability of user clicking an ad, and b_L is a bias. Therefore, the output of the ith network in the cross-layer and the combined output of k networks are

$$\begin{aligned} h_{L_i} &= f\left(\mathbf{I}^b\right), \\ \widehat{y} &= \frac{1}{k}\sum_{i=1}^{k} h_{L_i}, \end{aligned} \tag{15}$$

where $\widehat{y}$ is the prediction of CTR for the whole model through the average operation.

In the two-tower model, the cosine function was applied to calculate the CTR for the representation of the ad feature and the user feature to get the potential audience related to the ad. The feature cross-layer with bridge connection module proposed in this paper has a certain significance to improve the accuracy of the model. First, the Hadamard product is used to calculate the matching degree between feature representations, and the prerealized lower-order features are crossed. Second, the extracted user and ad feature vectors are taken as part of the input of the deep network, enabling the model to explore the higher-order implicit information among the features [7]. Finally, the result of the Hadamard product is spliced with the feature vectors of users and ads as the input of the network to form a bridge connected module, as shown in equation (13). In this way, the feature cross-layer can discover the potential relationship between low- and high-order features at the same time.

The binary cross-entropy loss is widely used in CTR prediction task, which is defined as follows:

$$\text{Logloss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\widehat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \widehat{y}_i\right)\right], \tag{16}$$

where N is the number of samples in training set. y_i and $\widehat{y}_i$ denote the ground truth and the predicted click probability, respectively. We define $\widehat{y} = \sigma(\varphi(x))$, where $\varphi(x)$ represents the model function given input features x, which contains user and ad information, and $\sigma(\cdot)$ is the sigmoid function to map $\widehat{y}$ to [0, 1]. The core of CTR prediction modeling lies in how to construct the model $\varphi(x)$ and learn its parameters from training data. In this work, the prediction $\widehat{y}$ will be compute by average operation from the multiple predictions in cross-layer.

3.3. The Discussion of Feature Dual Supervision. Multiple networks model can jointly learn from different features, so that it can result in improved accuracy for CTR prediction task [24, 26, 27]. Each fully connected network in the feature expression unit (FEU) has different ability to extract information for different features, so multiple networks are used to extract the same user or ad features to obtain multiple representations to strengthen the expression ability of the FEU. Multiple networks learning are a promising method to learn relationships among different features. However, these approaches deal with a limited number of characteristic tasks [24, 25]. Therefore, in order to alleviate the limitation and combine multiple representations, we propose a feature supervision unit (FSU). This unit consists of a single fully connected network, which gains supervised access under a supervised vector as input. In the description of the feature expression layer of the FDSM model, the user and ad feature representations are extracted in the same way. When user feature vector is extracted in the FEU unit, the inputs of the user's FSU unit are the full-volume information of the ad and the mean pooling features of the user; similarly, when the ad feature vector is extracted in the FEU, the inputs of the ad's FSU unit are the full-volume information of the user and the mean pooling features of the ad. Due to the input characteristics of the supervision unit, the operation of supervision has a two-level meaning.

Firstly, during the process of extracting the feature vector of the user, the user supervised vector contains ad full-volume information. And the input of the supervision unit contains the full-volume features of the user when the ad features are expressed, which belongs to the characteristics of the opposite side and this way means dual. It shows that the effective representation of users is influenced by the advertising information, and the effective representation of ads is influenced by the features that users care about, which is the first meaning of supervision.

Secondly, adding the same-side mean pooling feature vector for the supervised vector, the FSU unit can be used to discover the cross-information between users and ads in advance, making the supervision more sufficient, which is the second meaning of supervision. The underlying meaning of the whole is that users' behavioral decisions are made

under the information of the ad, while the extraction of effective information of the ad is expressed with the features that are concerned with the user. Therefore, the fusion process of the two levels is dual supervision for feature expression.

3.4. Audience Expanded by FDSM. There are many methods of audience targeting in online advertising, such as geotargeting. In this paper, we focus on the user profile and the abundant behaviors to expand audiences. We train the FDSM model from an advertiser's point of view through a large collection of ad campaigns that involves a large number of seed and nonseeded users. Specifically, given an ad a and a candidate pool C, the potential user set $T \subset C$ is obtained according to the click rate, and then the ad a is displayed to these users through the ad delivery system. In order to search the potential audiences of an advertising campaign more efficiently, we first train the FDSM model in all advertising campaigns to obtain the prior model. After that, rich behavioral information of users, such as keywords queried by users and behavioral interests, is collected from the online advertising system. The prior model is used to conduct microtraining on the new feedback log data of the ad a to obtain the customized model. Finally, the potential audiences of the ad are found by using the customized model. The overall process is shown in Figure 2.

The online advertising system collects a large amount of campaign data and caches it in the offline database. The offline data platform periodically processes the data generated in the past period, and then uses the data to train FDSM to obtain the prior model. The online platform is responsible for processing the data in the recent period to get the feedback data of a certain ad campaign, and then the customized model of the campaign will be obtained according to the feedback information. Finally, the customized model is used to retrieve potential audiences closely related to the campaign in the pool of candidate users in the data management platform (DMP), and the Top-N audiences are ranked according to the CTR to implement the campaign. The system transmits the users' feedback logs about the ads through the data highway to both the offline database and the online feedback database, and the whole system forms a closed-loop decision process.

4. Experiment

This section describes the experimental scheme in detail, including experimental datasets, comparison models, evaluation metrics, experimental details, comparisons results, ablation study, and discussion.

4.1. Datasets. Tencent Lookalike dataset(https://algo.qq.com/archive.html): The public dataset for Tencent Ads competitions in 2018 is based on the advertisers providing more than one hundred seed sets, which contain a large number of user characteristics and aim to expand potential

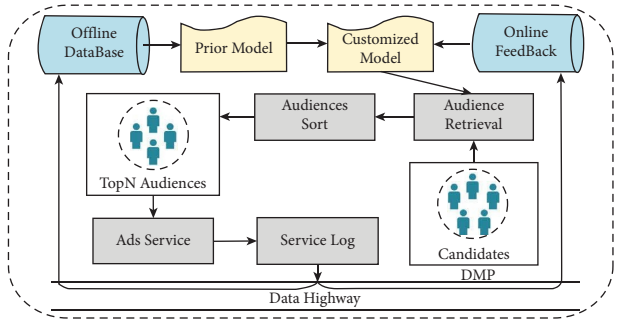

Figure 2: Audience targeting framework based on FDSM model.

audiences for these campaigns. To ensure the security of service data, all data is desensitized. The whole dataset is divided into a training set and a testing set. Each advertisement has eight categorical features: ad ID, advertiser ID, campaign ID, creative ID, creative size ID, ad category, product ID, and product type. Each user contains 19 features: including age, gender, education, carrier, consumption ability, geographical location, house, type of Internet access, five groups of interest categories, three groups of topics, and three groups of keywords.

MovieLens dataset (https://grouplens.org/datasets/movielens/): The public dataset contains 6,040 users; each of them consists of user ID, gender, age, occupation, and zip code, and holds 3,883 movies, each movie including movie ID, title, and genres. And it was rated by user with a score that among of 5 scale, and recorded timestamp for the rating behavior. In this study, in order to fit the audience targeting, we firstly according to the movie genres and the normalized years, the years were extracted from the movie title, to cluster all the movies into 50 groups thought k-means method. Each group was regarded as an ad group, and target audiences were found for each ad group. Meanwhile, in order to make the sample data suitable for CTR prediction task, we converted the rating data into a binary classification data [11]. Specifically, we label the original rating with 4 and 5 to be seed users (labeled as 1), and the rest are recorded as nonseed users (labeled as 0). Finally, based on the sequence of the timestamps of each movie being rated by users, the sample of 80% rating numbers with the top time is used as the training set, and the rest is used as the testing set. This results in 80% training data and 20% testing data for each ad group. Training data in all ad groups are taken as training sets, and all test data constitute testing sets.

Statistics about the datasets are shown in Table 2. The ratio of positive to negative samples in the Tencent Lookalike dataset is 1 : 20, while the ratio of positive to negative samples in the MovieLens dataset after processing is close to 1 : 1. The training set and testing set corresponding to each ad/ad group contain both seed and nonseed users. The users in the testing set are regarded as candidates for testing. In the system of audience targeting proposed in this paper, in order to obtain a prior model in the offline data, as shown in Figure 2, we take the training set of Tencent Lookalike according to 50% of the positive and negative samples of each seed set as the offline data and the rest data as online

Table 2: Statistics of the experimental datasets.

Dataset	Lookalike	MovieLens
Ads/movies	173/*	50/3883
Seeds	421961	461206
Nonseeds	8376853	337438
Candidates	2265989	201565

data. In this method, the offline data and online data are obtained to simulate the whole audience targeting process. For the training set of MovieLens, we consider it as both offline and online data.

In the two-tower model, the initial application field is natural language processing. Tencent Lookalike dataset not only contains user profile information, but also text information such as search keywords, favorite topics, and interests of users. FDSM model is improved on the basis of the two-tower model, so it can be used as the data set of experiments. The FDSM model proposed in this paper belongs to the CTR model, so it is necessary to add more data for experiments to show the advantages of this model. MovieLens is a dataset frequently used by the CTR model, and it is very convincing to use this dataset for experiments.

4.2. Baselines. In online advertising audience targeting, we compare our proposed model FDSM with the following baseline methods.

FM [4] combines the advantages of support vector machine and factorization model, which has demonstrated its effectiveness in many CTR prediction tasks.

MLP is popular structural model that embed each feature for a sample into an embedded vector, then obtains a dense embedding representation through concatenation operation, and feeds it into a fully connected network to automatically learn the CTR prediction.

DeepFM [8] adds deep neural network as the deep part on the basis of FM model, so that the model can learn higher-order feature interactions. The interaction terms of FM and the output of deep network will be model for CTR prediction.

PNN [12] model applies a product layer after embedding layer and multiple fully connected layers to explore the high-order feature interactions.

Two-tower [5] is a popular model in retrieval tasks. In this paper, the user features are input into the user tower and the ad features are input into the ad tower, by which the user and advertisement features are mapped into a shared semantic space. The cosine function is applied to calculate the matching scores by the extracted expression vectors of user and ad.

DCN [9], which proposed a deep cross-network to perform high-order feature interactions in an explicit way. In addition, it integrated a deep neural network. The output from both networks to achieve CTR prediction task together.

Wide&Deep [7], which differs from the DCN model, it adds a "wide" part on top of DNN. As a general learning framework that combines a wide network and deep neural network to achieve the advantage of both. The output of the last layer of DNN and "wide" part are inputted a linear model to complete the CTR prediction task.

AFN+ [28], AFN applies logarithmic transformation layer to learn adaptive-order feature interactions. AFN+ further integrates AFN with a deep network.

4.3. Evaluation Metrics. In this study, we use four metrics to measure the performance among of models. In the field of CTR prediction, AUC (Area under ROC Curve) is a widely used metric [29], which reflects the ranking quality of the prediction sample, and a higher AUC indicates a better CTR prediction performance. In this paper, according to the meaning of audience targeting, we calculate the AUC score for each ad in the testing set, and then calculate the average AUC of all ads, denoted as GAUC. In addition, we use the equation (16) to calculate the loss of audience prediction for each ad in the testing set; the smaller loss means better model performance. The average value is the final test result, which is denoted as Logloss. We also apply another two metrics: Precision@K% and Recall@K% [22], which indicates that after retrieval from candidate users, the Top-N candidates is selected as the target user to calculate precision and recall rate of the model. They are defined as follows:

$$\begin{aligned} \text{Precision@K\%} &= \frac{|S \cap T|}{|T|}, \\ \text{Recall@K\%} &= \frac{|S \cap T|}{|S|}, \end{aligned} \tag{17}$$

where S denotes the set of the seed of a certain ad campaign, and T denotes the set of the top targeting audiences after predicted with number of $K\% \times |S|$ by the audience expansion model for the ad. In this study, according to the approximate ratio of positive and negative samples in the whole dataset, we set K for Tencent Lookalike dataset and MovieLens dataset as 5 and 50, respectively.

4.4. Implementation Details. In this section, we will introduce the experimental environment and parameter details. As for the experimental parameter details, for the sake of fairness, we set the offline learning rate parameter λ_{off} and the online learning rate parameter λ_{on} for the training process of prior model and the customized model in the audience targeting system. Both of the learning rates of offline and online stage about Tencent Lookalike, which are tuned from 0.00002 to 0.0002 and the step is 0.00002. While for the MovieLens, the offline learning rate parameter is the same as the Tencent dataset, the online learning rate ranges from 0.0001 to 0.001 with a step of 0.0001. In the FDSM proposed in this paper, the parameters $[m, n, k]$ of Feature Expression Unit (FEU) and the number of fully connect networks in the cross-layer are set as [6, 6, 5] in Tencent

Lookalike and [8, 4, 8] in MovieLens, and the hidden layer structure of all fully connect networks networks in FSU and cross-layer is [128, 64]. For all other models, the hidden layer structure of all networks is [256, 128, 64]. The dimension of the embedding vector is 64. In the offline experiment, the offline full data is used for training, 8 ad campaigns are sampled and the minimum sample size is 512 for every ad in each generation, and the training times of the prior model is one epoch. In the online process, according to the order of the online training data of an ad campaign, 512 samples are applied in each iteration to train the prior model with three epochs to get the customized model. The Adam [30] is applied as the optimizer to optimize network weight parameters both online and offline. In the experiment, all models were coded in Python language on PyTorch l.6.0. We conduct our experiments with platform is CPU version of Intel Xeon Silver 4210 with 2.2 GHz. The memory of the device is 32 GB, and GPU version is independent graphics card GTX2080Ti.

4.5. Model Comparisons. In this part, we will analyze the experimental results from Table 3, Figures 3 and 4. Where in Table 3 are shown the ten prior models obtained for each model trained with all offline learning rate parameters λ_{off}, and each prior model obtains ten training customized models through different online learning rate parameters λ_{on}. Finally, the optimal results are obtained for each model tested with one hundred custom models. Each coordinate of Figures 3 and 4 represents the average value that the ten customized models are obtained by training all prior models of each method under one λ_{on}, then ten groups of test indicators under all parameters were calculated. From the results, we can summarize as follows:

(1) As shown in Table 3, among the performance results of all CTR prediction models, the best performances are highlighted in bold, and the best baseline results are highlighted in the underline. As can be observed from the table, the proposed FDSM model has been improved in different degrees in Lookalike and MovieLens datasets. For example, on Lookalike dataset, FDSM surpasses the suboptimal MLP model over 1.03% on GAUC metric, and improves 2.83% and 2.86% for precision and recall, respectively, with lower Logloss metric values than the other baseline models. On the MovieLens dataset, the FDSM model outperforms the PNN model by 0.80% on the GAUC metric, comparing with the AFN+model, the corresponding precision and recall rates are improved by 0.62% and 0.67%, and the Logloss metric values are also lower than those of other baseline models. This demonstrates that the FDSM model can extract more fine-grained feature expression of the users and ads after supervised operation, so as to find accurate matching patterns between feature information in feature cross-layer. And it also shows the effectiveness of the FDSM model in CTR prediction tasks and audience prediction.

(2) In Figures 3 and 4, they present the average performance of the GAUC and Logloss metrics for all models in the online stage on both datasets. Figure 3 represents the results of the tests on the Lookalike dataset. In this figure, we can clearly observe that the FDSM model outperforms the other baseline models in both metrics on average under all parameters of online learning rate. Figure 4 shows the online test results of all models on the MovieLens dataset. We can see that the average performance of the FDSM model on GAUC and Logloss does not reach the optimum when the online learning rate parameter is below 0.0002, but after 0.0002, the average performance outperforms the other baseline models and is able to achieve the global optimum average performance on the GAUC metric. As for the Logloss metric, the performance reaches the optimum after 0.0005. It further illustrates that the FDSM model outperforms than other baseline models in average performance.

(3) On the Lookalike dataset, through the experimental results of Table 3, we can observe that the optimal performance of the FM model is lower than the other models, and the average performance reflected from Figure 3 is also the worst, due to the fact that FM is a shallow structure, which can only be limited to second-order interaction. Other models all involve deep networks; second-order and higher-order feature interaction can be found at the same time. Therefore, the performance of them exceeds FM. The optimal and average performance of the PNN model is also unsatisfactory, and one possible reason is that the model applies the inner product of features as part of the deep network input, and it is the same as the FM model, where the second-order feature interaction affects the deep network finds higher-order information. The performance of the AFN+model is close to PNN, and its logarithmic transformation layer is also applied to the advance feature interaction, only with an additional exponential factor, so its performance on this dataset is close to that of the PNN.

DeepFM differs from PNN in that the results of the second-order interaction do not participate in the deep network, but enter the linear model together with the output of the deep network and finally get the prediction results, so this model performs better than PNN, but the experimental results are lower than Wide&Deep, MLP, and DCN models. Wide&Deep combines the output of a shallow structure "Wide" and a deep network, and the prediction results are obtained through the linear model, while DCN is combined with a deep cross-network and a deep network. For MLP, the high optimal performance in the baseline model is obtained by using only the deep network, which shows the powerful capability of the model. Two-tower's performance is still lower, probably due to the use of

Table 3: Results of comparison experiment on Lookalike and MovieLens.

Datasets	Lookalike				MovieLens			
Model	GAUC	Logloss	Precision@5%	Recall@5%	GAUC	Logloss	Precision@50%	Recall@50%
FM	0.7076	0.1661	0.2225	0.2295	0.6974	0.6171	0.7188	0.6214
MLP	*0.7291*	*0.1616*	*0.2403*	*0.2479*	0.7003	0.6119	0.7193	0.6211
DeepFM	0.7272	0.1620	0.2385	0.2462	0.7001	0.6094	0.7192	0.6210
PNN	0.7165	0.1642	0.2280	0.2351	*0.7016*	0.6112	0.7205	0.6218
Wide&Deep	0.7285	0.1617	0.2399	0.2478	0.6975	0.6110	0.7171	0.6183
Two-tower	0.7221	0.1725	0.2393	0.2470	0.7011	*0.6091*	0.7191	0.6208
DCN	0.7287	0.1617	0.2401	0.2477	0.6982	0.6162	0.7184	0.6199
AFN+	0.7170	0.1648	0.2214	0.2282	0.7002	0.6097	*0.7209*	*0.6228*
FDSM	**0.7366**	**0.1601**	**0.2471**	**0.2550**	**0.7072**	**0.6072**	**0.7254**	**0.6270**

The best performances are in bold.

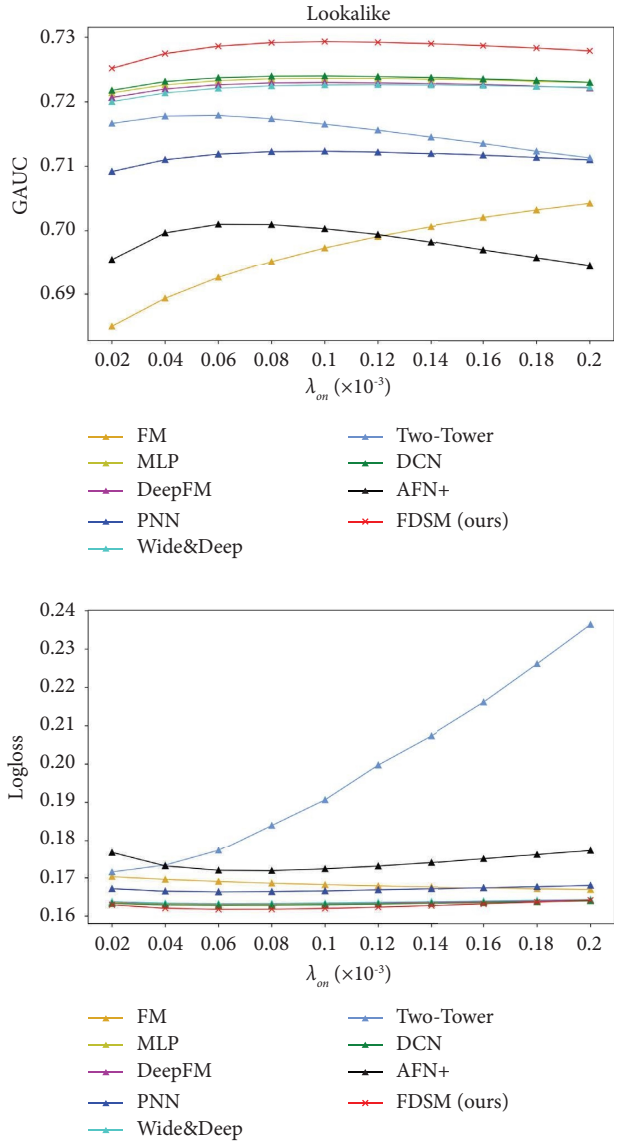

Figure 3: The average of GAUC and Logloss of different models on Loolalike dataset during online stage.

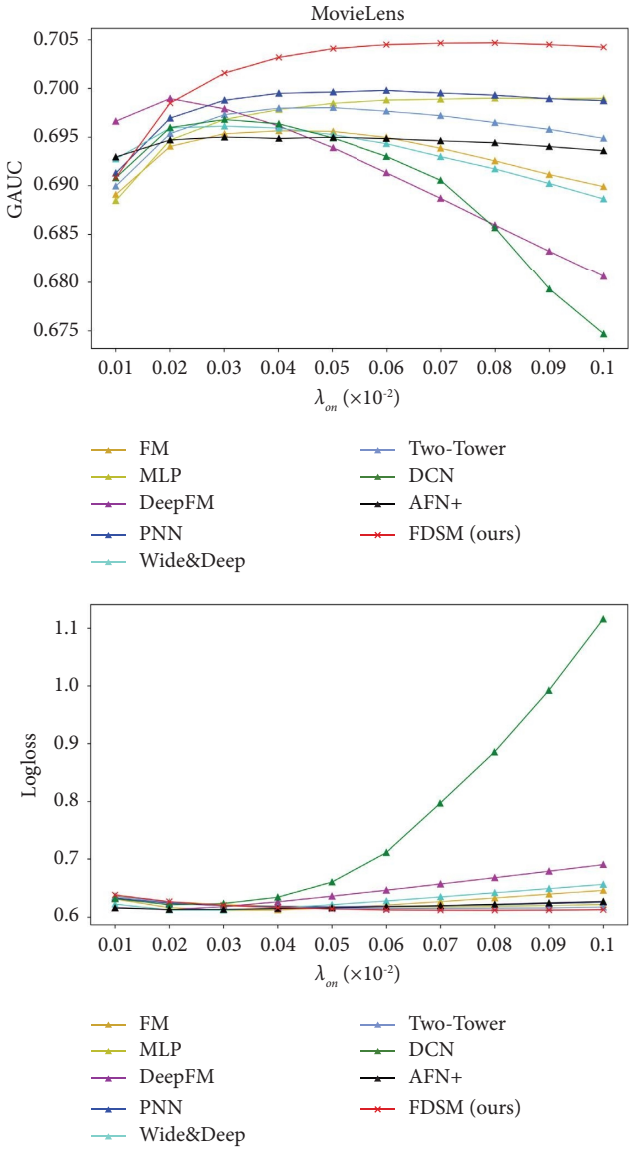

Figure 4: The average of GAUC and Logloss of different models on MovieLens dataset during online stage.

cosine function in cross-layer, and the low accuracy of matching between users and ads. Through analysis, it can be seen that the performance of the model applied to explicit second-order feature interaction is low, such as FM, PNN, and DeepFM. The reason may be that listing all cross-features, including irrelevant ones, will introduce noise feature combinations and reduce model performance [28]. However, the performance is better for models that do not involve explicit second-order feature interaction on Lookalike data, such as Wide&Deep, MLP, and DCN models.

(4) On the MovieLens dataset, FM, Wide&Deep, DCN, and DeepFM have lower both optimal performance in Table 3 and average performance presented in Figure 4. PNN, AFN+, MLP, and tow-tower have higher optimal and average performances. From the above analysis, it is clear that FM is limited to second-order interaction and therefore has lower performance. While Wide&Deep, DCN, and DeepFM models all use a combination of the outputs of nondeep network and deep network, compared with PNN, MLP, and two-tower models that only use deep networks, their ability to match information is weaker in the shallow structure. The AFN+model applies adaptive-order feature interaction as the shallow structure, its coefficient factors can be automatically adjusted and weaken the ability of the shallow structure; and it cooperates with the deep network to achieve CTR prediction, which indicates the strong performance of deep networks together.

(5) Why do all the models perform so differently on the two datasets? In the previous dataset introduction, we know that the Lookalike dataset contains not only the user profile information, but also the keywords and topic features of the user's query, which belongs to the text corpus, and if these features are not extracted, it will be difficult to directly perform feature interaction to discover the relationship between features, and even affect the ability of other structures, such as FM, PNN, and DeepFM. As for MovieLens dataset, there is only the user profile information and rating, and no text information. Therefore, feature interaction can be performed in advance and will not affect the performance of deep networks, such as PNN and AFN+ model. Through the above, it can be seen that the models applied to deep networks can all achieve feature extraction and higher-order interaction of features at the same time. In the FDSM model, we propose the FEU, FSU units and cross-layer with bridge connection based on the two-tower model, all of which use the deep network, which not only strengthens the feature extraction ability, but also enables more efficient features interaction.

4.6. Ablation Study. In ablation study, basing on the two-tower model, we will conduct experiment to analyze the influence on the number of fully connected networks in Feature Expression Units (FEUs) and feature cross-layer in the FDSM model, the Feature Supervision Units (FSU) and the cross-layer with bridge connection.

4.6.1. Ablation Experiment Setup. As described the implementation details, we still conduct experiments under the same setting of learning rate parameters, and each group of experimental achieves the optimal performance among 100 groups of results. It is mainly divided into two aspects; the specific arrangement is as follows.

Firstly, in order to explore the influence of the number of feature networks in the two-tower model, we set the structure parameters of $[m, n]$ as [6, 6] on Lookalike dataset, while [8, 4] on MovieLens dataset. The cosine function on the feature cross layer was still used, which is denoted as

Table 4: Results of ablation study.

Datasets	Lookalike				MovieLens			
Model	GAUC	Logloss	Precision@5%	Recall@5%	GAUC	Logloss	Precision@50%	Recall@50%
MNTT	0.7193	0.1678	0.2280	0.2244	0.7026	0.6082	0.7206	0.6221
$FDSM_a$	0.7268	0.1635	0.2401	0.2471	0.7042	0.6100	0.7245	0.6263
$FDSM_b$	0.7289	0.1644	0.2403	0.2472	0.7047	0.6075	0.7235	0.6254
Two-tower	0.7221	0.1725	0.2393	0.2470	0.7011	0.6091	0.7191	0.6208
TT_a	0.7283	0.1619	0.2412	0.2489	0.7011	0.6112	0.7192	0.6209
TT_b	0.7288	0.1619	0.2414	0.2490	0.7013	0.6089	0.7197	0.6214
FDSM	**0.7366**	**0.1601**	**0.2471**	**0.2550**	**0.7072**	**0.6072**	**0.7254**	**0.6270**

The best performances are in bold.

multinetwork two-tower (MNTT). To study the influence of Feature Supervision Unit (FSU) on the basis of MNTT, as introduced in the discussion of feature dual supervision, two levels of supervision need to be set, so the input of the supervision unit is replaced. In the first method, the user supervision vector is set to e^a, and the ad supervision vector is set to e^u. This method is denoted as $FDSM_a$. In the second method, we set the user supervision vector to $e^{a\bar{u}}$ and the ad supervision vector to $e^{u\bar{a}}$, which is denoted as $FDSM_b$.

Secondly, to explore the effect of the proposed bridge connection unit in the feature cross-layer, we replaced the cosine function with k fully connected networks in the two-tower model, where k is set to 5 on Lookalike dataset and 8 on MovieLens dataset. For the bridge connection in feature cross-layer, we will rebuild of the bridge vector $\mathbf{I}^b$, which was proposed in Section 3. In the first method, the concatenation operation is applied to combine features expression vector of users and ads from their tower, respectively, that is $\mathbf{I}^b = \mathbf{I}^u \oplus \mathbf{I}^a$, which is fed into k fully connected networks, denoted as TT_a In the second method, the input vector of the fully connected networks through the bridge unit proposed in this paper, that is $\mathbf{I}^b = (\mathbf{I}^u \odot \mathbf{I}^a) \oplus \mathbf{I}^u \oplus \mathbf{I}^a$, which denoted as TT_b Through these two methods, the effectiveness of the proposed feature cross-layer with bridge structure will be verified.

4.6.2. The Impact of the Supervision Unit. In order to explore the impact of FSU, we set the $FDSM_a$ and $FDSM_b$ models based on MNTT for the study. The reason for using the MNTT model instead of the two-tower model is to avoid the influence of the number of different networks in the process of feature extraction. Another reason is that the softmax function is applied in the output of the FSU for the supervised weight factor, so the FSU unit fails when $[m, n]$ is set to [1, 1], which makes it impossible to make a fair comparison. As shown in Table 4, the performance of the $FDSM_a$ and $FDSM_b$ models exceeds that of MNTT on both datasets. It illustrates that the supervision unit can enhance the feature extraction capability of the FEU. It can also be found that the performance of supervision in the $FDSM_b$ method is significantly better than that in the $FDSM_a$ method, which indicates that the same-side information can discover the correlation between users and ads information in advance, and finally obtain more fine-grained users and ads information in the feature extraction layer. Finally, through the above analysis, it

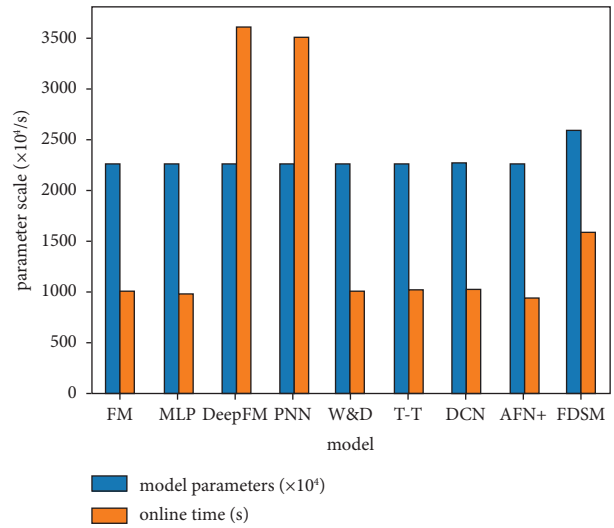

Figure 5: Comparison of online time and parameters for all models.

can be clearly found that the multinetwork feature extraction unit and feature supervision unit proposed in this paper not only have strong feature extraction ability, but may also be able to adapt to different datasets.

4.6.3. The Impact of Bridge Connection. In order to explore the impact of multi-network with bridge connection in cross-layer, we will analyze the experimental results of two-tower, TT_a, and TT_b. According to the results in Table 4, the performance of TT_a and TT_b are higher than that of Two-Tower, which indicates that multi-network is more accurate than cosine function in calculating the matching degree between user and ad. The performance of TT_a is lower than TT_b, indicating that the bridge connection with the Hadamard product can better calculate the matching degree between two expression vectors. Therefore, the feature cross-layer with bridge connection proposed in this paper has excellent information matching ability.

Through the introduction of this part, it is shown that the proposed FSU and feature cross-network with bridge-connected module have high performance in CTR prediction. By summarizing all the results in Table 4, we can find that the performance of the FDSM model is better than that of the other CTR models, which indicates that the method of

combining the FSU and the feature cross-network with bridge connection has higher performance than either part.

4.7. Complexity Analysis Experiment. In this study, experimental tests were conducted on Lookalike dataset to calculate the number of parameters and online running time of each model, as shown in Figure 5. In this picture, the x-axis represents the model and the y-axis represents the parameter scale. The $(\times 10^4)$ describes the unit of the number of model parameters and (s) describes the online running time of the model, whose unit is second. To clarify, W&D stands for Wide&Deep model and T-T stands for Two-Tower model.

First, it illustrates that the FDSM model proposed in this paper has the largest number of parameters because it applies multiple networks, while other models are all single network. Therefore, the FDSM model consumes more memory. Second, compared with baseline models, the online running time of FDSM is less than that of DeepFM and PNN, but more than other models. In general, in terms of the number of parameters and online running time, the model does not have an absolute advantage, but reaches closely the average performance. However, according to the results from Table 3, FDSM can achieve the best performance at the expense of certain memory and time. Hence, with the continuous improvement of hardware performance, FDSM model can be better applied to online advertising system.

5. Conclusion

In this paper, we propose a novel CTR model called the Feature Dual Supervision Model (FDSM) for advertising audience targeting system. This model is based on the two-tower model, aiming at the shortcoming of the two-tower model in the process of feature extraction, the lack of information interaction between the towers leads to the loss of details in the feature. In the FDSM model, Feature Expression Unit (FEU) and Feature Supervision Unit (FSU) are designed. The FEU unit is used to extract features from users or ads information to obtain a representation matrix with multiple feature representations. And the supervised weight vector is generated by the FSU unit. Then the supervised weight vector is applied to achieve supervision of the representation matrix to obtain a unique representation. In addition, we propose feature interaction with bridge connection to find more efficient matching patterns for user and ad representation. Finally, we conducted a large number of experiments, and through comparative experiments, it is shown that our proposed FDSM model surpasses many classical CTR models, and it is also found that the FDSM model may be adapted to more different contexts. The effectiveness of the proposed FEU, FSU, and bridge-connected cross-networks are illustrated by the ablation experiments.

In future work, we will further study the effects brought by different neural networks of FEU and FSU units, such as convolution neural networks (CNN) or adding attention mechanism in the networks. In addition, different designs will be made for the feature cross layer. By studying different bridge connection modules and exploring different network influences, such as setting residual network (ResNet), so that the more efficient feature input is explored to achieve efficient matching of information in the feature cross-layer.

Data Availability

The data can be obtained from the following link https://github.com/haipengni/DataForAds. The data are also available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Z. Gharibshah and X. Zhu, "User response prediction in online advertising," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–43, 2021.

[2] P. Palos-Sanchez, J. R. Saura, and F. Martin-Velicia, "A study of the effects of programmatic advertising on users' concerns about privacy overtime," *Journal of Business Research*, vol. 96, pp. 61–72, 2019.

[3] Z. Zhao, S. Yang, G. Liu, D. Feng, and K. Xu, "Fint: field-aware interaction neural network for ctr prediction," 2021, https://arxiv.org/arXiv%20preprint%20arXiv:2107.01999.

[4] S. Rendle, "Factorization machines," in *Proceedings of the 2010 IEEE International conference on data mining*, pp. 995–1000, IEEE, Sydney, NSW, Australia, December 2010.

[5] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 2333–2338, San Francisco, CF, USA, October 2013.

[6] Y. Juan, D. Lefortier, and C. Olivier, "Field-aware factorization machines in a real-world online advertising system," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 680–688, Perth, Australia, April 2017.

[7] H.-T. Cheng, L. Koc, J. Harmsen et al., "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10, Boston, MA, USA, September 2016.

[8] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," 2017, https://arxiv.org/abs/1703.04247.

[9] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, pp. 1–7, Halifax, NS, Canada, August 2017.

[10] Y. Qu, B. Fang, W. Zhang et al., "Product-based neural networks for user response prediction over multifield categorical data," *ACM Transactions on Information Systems*, vol. 37, no. 1, pp. 1–35, 2018.

[11] G. Zhou, X. Zhu, C. Song et al., "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1059–1068, London, United Kingdom, July 2018.

[12] Y. Qu, C. Han, K. Ren et al., "Product-based neural networks for user response prediction," in *Proceedings of the 2016 IEEE*

16th International Conference on Data Mining (ICDM), pp. 1149–1154, IEEE, Barcelona, Spain, December 2016.

[13] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 101–110, Shanghai, China, November 2014.

[14] H. Palangi, L. Deng, Y. Shen et al., "Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[15] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[16] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[17] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Matchsrnn: modeling the recursive matching structure with spatial rnn," 2016, https://arxiv.org/abs/1604.04378.

[18] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, pp. 55–64, Indianapolis, Indiana, USA, October 2016.

[19] C. Xiong, Z. Dai, J. Callan, Z. Liu, and P. Russell, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieva*, pp. 55–64, Shinjuku, Tokyo, Japan, August 2017.

[20] A. M. Elkahky, Y. Song, and X. He, "A multiview deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th international conference on world wide web*, pp. 278–288, Florence, Italy, May 2015.

[21] M. Fan, J. Guo, S. Zhu, S. Miao, M. Sun, and P. Li, "Mobius: towards the next generation of query-ad matching in Baidu's sponsored search," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2509–2517, Anchorage, AK, USA, July 2019.

[22] Z. Liu, X.-F. Niu, C. Zhuang et al., "Two-stage audience expansion for financial targeting in marketing," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2629–2636, Virtual Event, Ireland, October 2020.

[23] B. Chen, Y. Wang, Z. Liu et al., "Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3757–3766, Virtual Event, Queensland, Australia, October 2021.

[24] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multitask learning with multigate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1930–1939, London, United Kingdom, July 2018.

[25] J. Zhao, B. Du, L. Sun, F. Zhuang, W. Lv, and H. Xiong, "Multiple relational attention network for multitask learning," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & Data Mining*, pp. 1123–1131, Anchorage, AK, USA, July 2019.

[26] Z. Qin, Y. Cheng, Z. Zhao, Z. Chen, D. Metzler, and J. Qin, "Multitask mixture of sequential experts for user activity streams," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3083–3091, Virtual Event, CA, USA, August 2020.

[27] D. Xi, Z. Chen, Y. Peng et al., "Modeling the sequential dependence among audience multistep conversions with multitask learning in targeted display advertising," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3745–3755, Virtual Event, Singapore, August 2021.

[28] W. Cheng, Y. Shen, and L. Huang, "Adaptive factorization network: learning adaptive-order feature interactions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3609–3616, 2020.

[29] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[30] D. P. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.