The Knowledge Engineering Review (2025), **40**, e3, pp. 1–38
doi:10.1017/S0269888925000025

The Knowledge Engineering Review

CAMBRIDGE UNIVERSITY PRESS

REVIEW

A comprehensive survey on advertising click-through rate prediction algorithm

Jing Bai[1], Xinyu Geng[1], Jiaqi Deng[2], Zhen Xia[1], Hongxia Jiang[1], Guoqiang Yan[1] and Jing Liang[1]

[1]Southwest Petroleum University, Chengdu, China
[2]Southwest Institute of Electronic Technology, Chengdu, China
Corresponding author: Xinyu Geng; Email: gengxy123@126.com

Received: 22 September 2022; **Revised:** 14 March 2025; **Accepted:** 14 March 2025

Keywords: Advertising click-through rate prediction; Recommender system; Deep learning; Shallow interactive model; Review

Abstract

Advertising click-through rate (CTR) prediction is a fundamental task in recommender systems, aimed at estimating the likelihood of users interacting with advertisements based on their historical behavior. This prediction process has evolved through two main stages: from traditional shallow interaction models to more advanced deep learning approaches. Shallow models typically operate at the level of individual features, failing to fully leverage the rich, multilevel information available across different feature sets, leading to less accurate predictions. In contrast, deep learning models exhibit superior feature representation and learning capabilities, enabling a more realistic simulation of user interactions and improving the accuracy of CTR prediction. This paper provides a comprehensive overview of CTR prediction algorithms in the context of recommender systems. The algorithms are categorized into two groups: shallow interactive models and deep learning-based prediction models, including deep neural networks, convolutional neural networks, recurrent neural networks, and graph neural networks. Additionally, this paper also discusses the advantages and disadvantages of the aforementioned algorithms, as well as the benchmark datasets and model evaluation methods used for CTR prediction. Finally, it identifies potential future research directions in this rapidly advancing field.

1. Introduction

The problem addressed by recommender systems is how to effectively suggest items to users in order to enhance their click-through rate (CTR) and overall satisfaction. CTR prediction plays a crucial role in both recommender and advertising systems, as its accuracy directly impacts the performance of recommendation algorithms. The development of collaborative filtering (CF) algorithms (Koren & Bell, 2015) dates back to 1992, marking the foundation of modern recommendation models. However, CF algorithms struggle with handling sparse matrices and maintaining similarity matrices. To overcome these limitations, matrix factorization (MF) techniques (Koren *et al*., 2009) were introduced. MF represents users and items through latent vectors, enabling the extraction of underlying patterns and effectively addressing the issue of data sparsity. Research into these recommendation models has significantly contributed to the advancement of CTR prediction methodologies.

Logistic regression (LR) (Richardson *et al*., 2007) is one of the earliest and most widely used methods for CTR prediction in industry. The LR algorithm employs a shallow interaction model to integrate multiple features for recommendation, playing a key role in the early development of CTR prediction techniques. The interaction between features is critical for prediction accuracy. To address the limitation of linear models, which cannot effectively capture feature interactions, many researchers have proposed

Cite this article: J. Bai, X. Geng, J. Deng, Z. Xia, H. Jiang, G. Yan, J. Liang. A comprehensive survey on advertising click-through rate prediction algorithm. *The Knowledge Engineering Review* 40(e3): 1–38. https://doi.org/10.1017/S0269888925000025

various enhanced CTR prediction models focusing on feature engineering and interaction. CTR prediction data typically involve multiple features, with categorical features becoming highly sparse after one-hot encoding. Generalized linear models such as LR and follow-the-regularized-leader (FTRL) (McMahan *et al*., 2013) struggle to model complex feature interactions (Chapelle *et al*., 2014). To overcome this, factorization machines (FM) (Rendle, 2010; Rendle, 2012a) were introduced, utilizing the embedding of two features as an inner product to capture second-order feature interactions. FM became a mainstream recommendation model in industry between 2012 and 2014 for several reasons: (1) It significantly reduces training overhead, with complexity reduced from $O(n^2)$ in POLY2 (Chang *et al*., 2010) to the linear complexity of $O(kn)$, where k represents the length of the implicit vector. (2) FM has a relatively simple structure compared to the more complex deep learning models, making deployment and service more efficient. (3) By introducing implicit vectors, FM effectively addresses the issue of data sparsity. However, FM typically captures only pairwise feature interactions, and as the number of features increases, the model's complexity grows significantly. To capture higher-order feature interactions, Blondel *et al*. (2016) and He *et al*. (2014) have proposed various methods.

In recent years, various machine learning tasks, including object detection (Szegedy *et al*., 2013; Zhao *et al*., 2019), natural language understanding (Dahl *et al*., 2011), and speech recognition (Hinton *et al*., 2012), have been revolutionized by end-to-end deep learning paradigms. Models such as deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN), and graph neural networks (GNN) have been continuously proposed. The powerful learning capabilities of deep learning have also been applied to CTR prediction (Wang *et al*., 2018). Zhang *et al*. (2021b) explains that depth is a necessary development trend for such tasks. Given the large volumes of training data, highly sparse features, and high performance requirements often associated with CTR prediction, algorithm design is primarily focused on addressing these challenges. Shan *et al*. (2016) proposed the Deep Crossing model, based on the classic DNN architecture of ResNet, for CTR prediction. This model effectively addresses several issues in the application of deep learning to recommender systems, such as feature engineering, sparse vector densification, and optimization of multilayer neural networks for target fitting. It has laid a strong foundation for subsequent research. With the advent of Microsoft's Deep Crossing model, Google's Wide & Deep model (Cheng *et al*., 2016), and other advanced models such as Factorization Machine-based Neural Networks (FNN) (Zhang *et al*., 2016) and Product-based Neural Networks (PNN) (Qu *et al*., 2016), the field of recommender systems and computational advertising has entered the era of deep learning.

The core objective of the attention mechanism is to identify and prioritize information that is most relevant to the task at hand, allowing the model to focus on useful data while minimizing attention to noise. Given current computational resource constraints, the attention mechanism is a crucial tool for enhancing efficiency. In CTR prediction, different samples correspond to distinct scenarios, and the importance of specific features or feature combinations varies depending on the sample and application context. Vaswani *et al*. (2017) introduced the Multi-Head Attention mechanism, which has provided valuable insights into understanding user interests–specifically, what these interests are and how they evolve. Historically, many models overlooked the varying impact of different features on prediction outcomes, with fixed training weights across all features. The LS-PLM model (Gai *et al*., 2017), a traditional recommendation model, addresses this by introducing an attention mechanism that classifies samples before calculating prediction scores within each category. Since 2017, a growing body of CTR prediction research has incorporated attention networks to better capture users' latent interests. Notable models include the attentional factorization machine (AFM) (Xiao *et al*., 2017), deep interest network (DIN) (Zhou *et al*., 2018), deepinterest evolution network (DIEN) (Zhou *et al*., 2019), behavior squence transformer (BST) (Chen *et al*., 2019a), and the search-based interest model (SIM) (Pi *et al*., 2020), among others.

To summarize, this paper offers an overview for those seeking to understand the development and current state of CTR prediction research, as well as for those interested in comparing different CTR prediction models.

1.1 Our contributions

This paper makes several significant contributions, summarized as follows.

1. **Taxonomy:** We classify CTR prediction models into two categories: shallow interaction models and deep learning-based prediction models (including DNN, CNN, RNN, and GNN).
2. **Comprehensive Review:** We provide an in-depth overview of CTR prediction technologies, offering detailed descriptions of representative models for each category, making necessary comparisons, and summarizing the corresponding algorithms.
3. **Resource Compilation:** We have compiled a wealth of resources on CTR prediction models, including classic and state-of-the-art models, benchmark datasets, and evaluation metrics.
4. **Future Directions:** We analyze the limitations of existing CTR prediction methods and propose potential directions for future research.

1.2 Organization of this paper

The remainder of this paper is organized as follows: Section 2 provides a brief introduction to the CTR prediction problem, reviews related work, including fundamental concepts of CNN, RNN, Graph, and Graph Embedding, and presents symbolic definitions along with a list of commonly used notations. Sections 3 and 4 summarize CTR prediction models by category. Section 5 discusses the advantages and disadvantages of the aforementioned algorithms, as well as commonly used datasets and evaluation metrics for assessing CTR prediction performance. Section 6 outlines current research trends in this field and highlights potential directions for future exploration. Finally, Section 7 concludes the paper.

2. Related work and symbol description

In this section, we present the CTR prediction problem, review related work, introduce the fundamental concepts of CNN, RNN, graph and graph embedding, and provide a list of the common symbols used in this paper.

2.1 Related work

Advertising CTR refers to the ratio of ad clicks to ad impressions. The goal of CTR prediction is to estimate the likelihood of an advertisement being clicked based on advertising and user information. The accuracy of this prediction directly impacts the advertising revenue of internet companies (Richardson *et al*., 2007). Online advertising typically uses four billing methods (Asdemir *et al*., 2012): monthly, Cost Per Mille (CPM), Cost Per Click (CPC), and Cost Per Sale (CPS). Among these, CPC and CPS are closely related to CTR. The expected revenue R for an internet platform can be expressed as $\mathrm{R} = \mathrm{CTR} \times \mathrm{CPC}$, where CPC represents the revenue generated by a single click on an advertisement. Therefore, accurate CTR prediction is crucial for maximizing user engagement, increasing user retention, and driving significant business value for the company. The relevant theories used in the subsequent model summary are as follows:

1. **Convolutional Neural Networks:** CNN (Gu *et al*., 2018) are characterized by sparse connections and weight sharing. The overall framework of classification task based on CNN is shown in Figure 1, which can be divided into feature extraction module and classification module. The feature extraction module extracts features from input through convolution layer and pooling layer, while the classification module is based on fully connected feedforward neural network. The two modules are connected through the flattening operation to flatten the feature matrix in multiple channels obtained by the feature extraction module into a one-dimensional vector, which will be used as the input of the classification module.

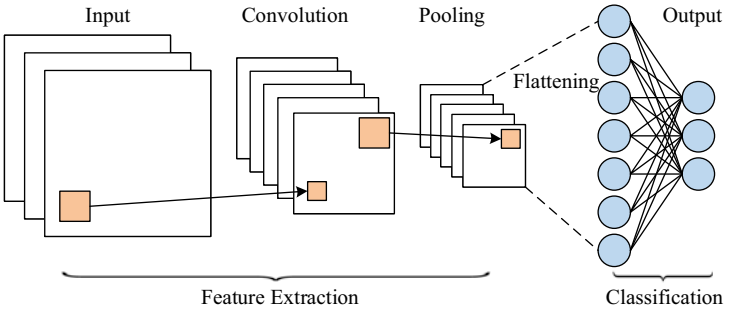

Figure 1. *Overall framework of convolutional neural network*

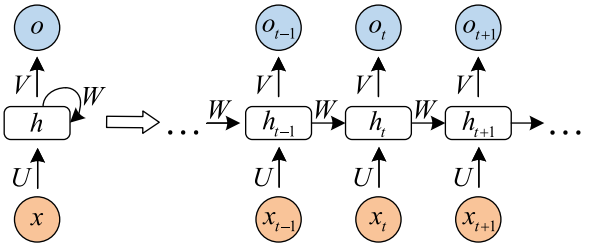

Figure 2. *Recurrent neural network*

In the convolution neural network, the feature map (Zou *et al*., 2018) is composed of multiple neurons, and each neuron is connected by the output of the upper layer neuron and the convolution kernel. Convolution kernel (Rawat & Wang, 2017) is a weight matrix of user-defined size, which acts on the local perception domains of different regions of the same image. The features of each local perception domain are extracted to generate the input value of the next layer of neurons. The convolutional layer convolves the input features, and its feature map is shown in formula (1). The pooling layer performs secondary extraction of input features through certain pooling rules, and its feature map is shown in formula (2), where H_i is the feature map of the i-th layer,

$$H_i = f\left(H_{i-1} \otimes w_i + b_i\right), \tag{1}$$

$$H_i = f\left(pooling((H_{i-1})) + b_i\right), \tag{2}$$

$f(x)$ is a nonlinear activation function, $\otimes$ represents the convolution operation of the convolution kernel and the feature map, $pooling(x)$ represents pooling rules, such as mean pooling, maximum pooling, random pooling, etc. w_i represents the weight vector of the convolution kernel of the i-th layer, b_i represents the bias term of the i-th layer.

2. **Recurrent Neural Network:** RNN (Zaremba *et al*., 2014) is a kind of neural network which is used to process time series information. In this paper, we consider discrete RNN, where the process is divided into multiple states and each state is time-stamped. Figure 2 (Wu *et al*., 2016) shows the basic idea of RNN, let x and o represent input and output, respectively, and use h to represent the values in the hidden layer, as well as three transfer matrices, U,V and W. There is a self-link in the hidden layer that indicates that it will update its value over time. Assuming that there are three states at $t-1$, t and $t+1$, $x(i)$ and $o(i)$ $(t-1 \leq t \leq t+1)$ represent the input and output in different states respectively. The hidden layer value $h\,(i)$ of state i will be updated according to the value $(h(i-1))$ of the previous state, as shown in formula (3):

$$h(i) = f(Ux(i) + Wh(i-1)), \tag{3}$$

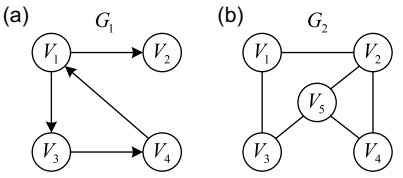

Figure 3. *Instance of graph*

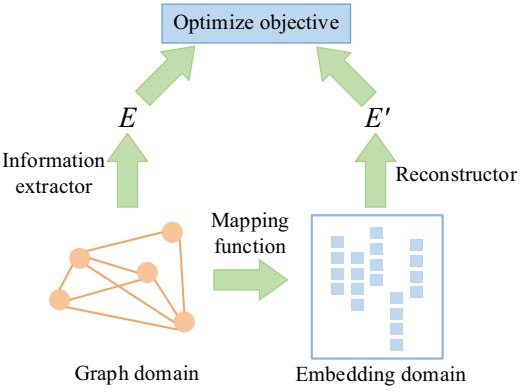

Figure 4. *A general framework for graph embedding*

where f represents nonlinear activation functions, such as tanh, ReLU and Sigmoid functions, and $o(i)$ is the predictive value of state i, which is formalized as follow:

$$o(i) = softmax(Vh(i)). \tag{4}$$

Therefore, we can get the output of each state.

3. **Graph:** Graph (Cai *et al*., 2018) can be represented as $G = (V, E)$, where V is the set of vertices and E is the set of edges (directed or undirected edges). The directed graph is shown in Figure 3(a), and the undirected graph is shown in Figure 3(b). Vertices and edges may contain additional information, collectively referred to as label information.The label of vertex v is defined as $\vec{l}_v \in \mathbb{R}^{n_v}$ and n_v is the dimension of the vertex label, usually containing the characteristics of the vertex. The label that defines the edge (v_1, v_2) is $\vec{l}_{v_1,v_2} \in \mathbb{R}^{n_E}$ and n_E is the dimension of the edge label, usually containing characteristics of the relationships between vertices. Graph structure (Scarselli *et al*., 2008) exists in various realistic scenarios, such as social network, citation network and knowledge graph, etc.
4. **Graph Embedding:** Graph embedding (Cai *et al*., 2018) aims to map each node in a given graph into a low-dimensional vector representation that typically preserves some key information of the node in the original graph. A node in a graph can be viewed from two domains: (1) the original graph domain, where nodes are connected via edges (or the graph structure) and (2) the embedding domain, where each node is represented as a continuous vector. We illustrate an overall framework of graph embedding in Figure 4, there are four key components in the general framework as: mapping function, information extractor, reconstructor, and optimize objective. In the figure, E is the extracted graph information, and $E^{'}$ is the reconstructed information. Thus, graph embedding maps graphs to low-dimensional spaces that retain graph information. Most graph analysis methods require high computational cost and space cost, but graph embedding provides an effective method to solve the problem of graph analysis.

Table 1 presents the classification of CTR prediction models discussed in this paper. We categorize these models into two groups: shallow interaction models and deep learning-based prediction models (including DNN, CNN, RNN, and GNN). In this table, "Shallow" refers to traditional CTR prediction

Table 1. *Classification and representative literature of click-through rate prediction models*

Category		Publications
Shallow		Richardson *et al*. (2007), McMahan *et al*. (2013), Rendle (2010), Chang *et al*. (2010), Blondel *et al*. (2016), He *et al*. (2014), Gai *et al*. (2017), Juan *et al*. (2016)
Deep	DNN	Shan *et al*. (2016), Cheng *et al*. (2016), Zhang *et al*. (2016), Qu *et al*. (2016), Xiao *et al*. (2017), Zhou *et al*. (2018), Pi *et al*. (2020), Guo *et al*. (2017), He and Chua (2017), Chen *et al*. (2019b), Zhao *et al*. (2021b), Zhu *et al*. (2017), Wang *et al*. (2017), Chen *et al*. (2021), Xue *et al*. (2020), Xu *et al*. (2021b), Lian *et al*. (2018), Liu *et al*. (2020a), Cao *et al*. (2021), Huang *et al*. (2021b), Zeng *et al*. (2020), Cheng and Xue (2021), Zhang *et al*. (2021a), Lu *et al*. (2021), Zhu *et al*. (2021), Zhou *et al*. (2020), Huang *et al*. (2019), Kaplan *et al*. (2021), Ouyang *et al*. (2019b), Lyu *et al*. (2020), Wu *et al*. (2020), Mishra *et al*. (2021), Cao *et al*. (2020), Ouyang *et al*. (2019a), Li *et al*. (2020b), Ouyang *et al*. (2020), Qin *et al*. (2020), Ge *et al*. (2018), Zhao *et al*. (2020), Huang *et al*. (2021a), Shi and Yang (2020), Zhu *et al*. (2020), Zhao *et al*. (2021a), Guo *et al*. (2021a)
	CNN	Liu *et al*. (2015), Chan *et al*. (2018), Chen *et al*. (2016), Shen *et al*. (2016), Zhou *et al*. (2016), Lei *et al*. (2016), Gligorijevic *et al*. (2019), Liu *et al*. (2019), Zhu (2021), Gao *et al*. (2018), Niu and Hou (2020), Edizel *et al*. (2017), Liu *et al*. (2020b), Guo *et al*. (2021c)
	RNN	Zhou *et al*. (2019), Wang *et al*. (2020a), Zhang *et al*. (2014), Feng *et al*. (2019), Xu *et al*. (2021a), Li *et al*. (2020a), Hong *et al*. (2021), Pi *et al*. (2019), Song *et al*. (2020)
	GNN	Li *et al*. (2019b), Li *et al*. (2019a), Su *et al*. (2021), Li *et al*. (2021), Feng *et al*. (2020), Chu *et al*. (2021), Guo *et al*. (2021b), Wang *et al*. (2021), Ouyang *et al*. (2021), Min *et al*. (2022), Zheng *et al*. (2022)

models based on shallow interactions, while ''Deep" encompasses models based on deep learning techniques, including deep neural networks, convolutional neural networks, recurrent neural networks, and graph neural networks. In the following two sections, we provide a brief overview of the representative models in each category.

2.2 Commonly used notations

Before formally introducing the CTR prediction model, we list the commonly used notations. Unless otherwise specified, all notations used in this paper are defined in Table 2.

3. The shallow interactive model

In internet application scenarios, the system can collect vast amounts of user and item data. The logistic regression model (Richardson *et al*., 2007) effectively leverages a variety of features and converts the problem into a binary classification task. The structure of the model is illustrated in Figure 5. However, since logistic regression lacks the ability to generate combinatorial features, its expressive power is limited.

The CTR prediction result obtained by the pattern with feature interaction is often more accurate than those without feature interaction. Hence, Chang *et al*. (2010) proposed the Poly2 model for CTR prediction, and the expression is:

Table 2. *Commonly used notations*

Notations	Descriptions
$\odot$	Element-wise product(Hadamard product)
$\sum$	Summation symbol
y	True label
x_i	Features
w, w_i	Weight of each feature
$\mathbf{W} \in \mathbb{R}^{n \times n}$	$\mathbf{W}$ belongs to n-dimensional feature space
$\mathbf{V} \in \mathbb{R}^{n \times k}$	$\mathbf{V}$ belongs to $n \times k$-dimensional feature space
w_{ij}	The weight of the intersection of the i-th feature and the j-th feature
$\langle v_i, v_j \rangle$	Inner product of v_i and v_j
k	Length of the latent vector
$\sigma(\cdot)$	The activation function
$\eta(x)$	The softmax function
D_1	The number of neurons in the hidden layer
N	The number of feature domains
M	The dimension of the embedding vector
$f_i \in \mathbb{R}^M$	f_i belongs to M-dimensional vector space
l	Layers of neural network
$x^1 \in \mathbb{R}^{n \times D \times 1}$	The input matrix of the first layer of convolution
a_{ij}	The attention score of $v_i \odot v_j$
$a(\cdot)$	The feedforward neural network
$\mathbf{c}_i^l$	The ith output of the lth pair of convolution and pooling layer
$q(\cdot)$	The pooling function
t_l	The number of feature maps in the ith layer
$C_{:,:,i}^1$	The ith feature map in the first convolutional layer
$\mathbf{h}(t)$	The hidden state in time t
$\mathbf{h}_t^i$	The tth hidden state of GRU for user i

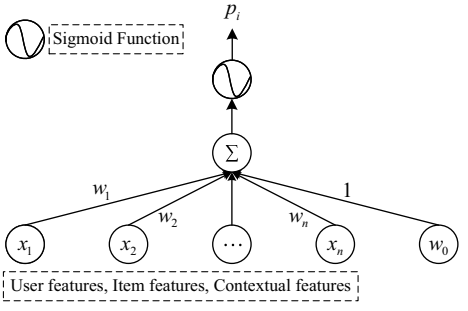

Figure 5. *Structure diagram of logistic regression model*

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j, \tag{5}$$

where $w_{ij}(i = 1, 2, \ldots, n-1; j = i+1, \ldots, n)$ is the weight of the interaction of the i-th feature and the j-th feature. The model adopts the non-selective feature interaction method to learn the second-order feature combination, which will make the originally very sparse feature vector more sparse, resulting in the lack of effective data training for the weights of most of the intersecting features, and the time complexity is $O(n^2)$.

In order to reduce the computational complexity of the model in learning second-order feature combination, Rendle (2010, 2012a) proposed the Factorization Machine (FM) model, which maps the high-dimensional sparse matrix to the low-dimensional dense vector, and learns the information of feature pairwise combination through the vector inner product. Since the features are not independent of each other, an implicit factor can be used to connect them in series. The FM introduces the idea of matrix decomposition to decompose the coefficient matrix of the cross term: $w_{ij} = \langle v_i, v_j \rangle$, The mathematical basis (Blum, 2012) is that when k is large enough, there is a real matrix $\mathbf{V} \in \mathbb{R}^{n \times k}$ for any symmetric positive definite real matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, so that $\mathbf{W} = \mathbf{V}\mathbf{V}^T$ holds. FM model can be expressed as:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j, \tag{6}$$

The cross term coefficient of the feature vector x_i and x_j is equal to the inner product of the implicit vector corresponding to x_i and the implicit vector corresponding to x_j: $\langle v_i, v_j \rangle = \sum_{t=1}^{k} v_{it} \cdot v_{jt}$, k is a hyperparameter, indicating the length of the implicit vector. In essence, it is embedding the feature, and the time complexity is $O(kn)$. FM does not consider that the implicit vector may show different distribution when combining the features of different feature fields. Therefore, Juan *et al*. (2016) introduced the concept of field-aware and proposed the field-aware factorization machine (FFM) model related to the feature field, the mathematical expression is:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle v_{i,f_j}, v_{j,f_i} \rangle x_i x_j, \tag{7}$$

although more information can be learned by using the feature field, FFM model gives a set of feature implicit vectors to individual features in each feature field, and the algorithm complexity is increased to $O(kn^2)$. Moreover, the feature interaction of FM and FFM is second-order, and at most, two features are crossed. Once there are more than two features, the complexity will become very high. Therefore, Blondel *et al*. (2016) extended the second-order FM to higher-order factorization machines (HOFM) and designed the ANOVA kernel (used when the higher-order is greater than 2) to ensure that the higher-order combination information of features can be learned when the interpretability is strong.

In order to get higher-order feature combinations, Facebook researchers (He *et al*., 2014) use the gradient boosting decision tree (GBDT) (Friedman, 2001) to extract and screen differentiated features and feature combinations and take the extracted features as the input of LR. This scheme is called GBDT+LR, which is the beginning of feature engineering modeling. Gai *et al*. (2017) put forward the large-scale piece-wise linear model (LS-PLM), also known as the mixed logistic regression (MLR), which was applied to all kinds of advertising scenes in Alibaba for a long time before the deep learning model was put forward. LS-PLM adopts the idea of divide and conquer on the basis of LR, after clustering and slicing the samples, logistic regression is applied in the sample sharding for CTR estimation. The structural characteristics of LS-PLM are similar to those of three-layer neural network (as shown in Figure 6), x is a large-scale sparse input data, the embedding operation is divided into two parts: the blue part is clustering embedding, and the green part is classification embedding, both projections are cast into the m dimensional space, and m is the number of categories. The formal expression is as follows:

$$f(x) = \sum_{i=1}^{m} \eta_i(x) \cdot \sigma_i(x) = \sum_{i=1}^{m} \frac{e^{\mu_i \cdot x}}{\sum_{j=1}^{m} e^{\mu_j \cdot x}} \cdot \frac{1}{1 + e^{-w_i \cdot x}}, \tag{8}$$

The clustering function η_i is the softmax function, which is responsible for dividing the features into different spaces of m. $\sigma_i(x)$ is the sigmoid function, which is responsible for predicting the feature fragments of m space. The space is divided into m regions for linear fitting, and finally, the results of the m regions are normalized in order to make the CTR prediction model more targeted for different user groups and different application scenarios.

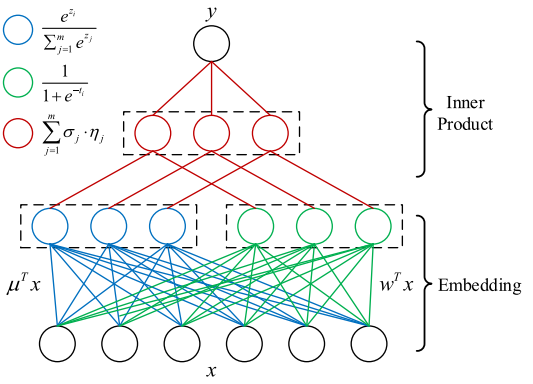

Figure 6. *Structure diagram of LS-PLM*

4. Deep learning models

Recently, recommender systems and computational advertising have entered the era of deep learning. On one hand, deep learning enables the extraction of deep feature representations for both users and items (Shaheen *et al*., 2016). On the other hand, it allows for the mapping of diverse data types to a shared latent space through automatic feature learning from multi-source heterogeneous data (Mu, 2018), thereby facilitating a unified representation of the data.

4.1 CTR prediction model based on deep neural network

Zhang *et al*. (2016) proposed factorization machine supported neural network (FNN) model, which used DNN to re-cross the second-order features of FM (Rendle, 2010) display expression, thus generating higher-order feature combinations and strengthening the learning ability of the model to data patterns. The input features are sparse after one-hot coding, which leads to the slow convergence speed of the embedding layer. FNN initializes the weight w of the embedding layer of the formally trained model with the feature implicit vectors pre-trained by the FM method. After introducing valuable prior information, the starting point of neural network training is closer to the optimal point of the target, which naturally accelerates the convergence process of neural network. However, the serial mode of FNN limits the expression ability of the whole model, which is limited to the upper limit of FM representation ability (second-order feature crossing), and only pays attention to the crossing of high-order combination features, so it is easy to lose the 'memory ability' of the model.

In the same year, Qu *et al*. (2016) put forward the product-based neural network (PNN) model to enrich the way of feature interaction. The PNN introduced the product layer and used the vector product (inner product or outer product) between features to learn feature combination information and capture cross-domain interactive information. The product layer structure of PNN is shown in Figure 7. z in the product layer is the linear operation part, $l_z = (l_z^1, l_z^2, \ldots, l_z^n, \ldots, l_z^{D_1})$, D_1 is the number of neurons in the hidden layer. The embedding vector $f = (f_1, f_2, \ldots, f_N)$ is defined as the vector $z = (z_1, z_2, \ldots, z_N)$, then the formula can be obtained:

$$l_z^n = \mathbf{W}_z^n \odot z = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\mathbf{W}_z^n\right)_{i,j} z_{i,j}, (i = 1, 2, \ldots, N), \tag{9}$$

where N is the number of feature fields, and M is the dimension of Embedding. p in the product layer corresponds to the product operation part, $l_p = (l_p^1, l_p^2, \ldots, l_p^n, \ldots, l_p^{D_1})$, divided into inner and outer modes, and $l_p^n = W_p^n \odot z$. The expressions for inner product and outer product modes are as follows:

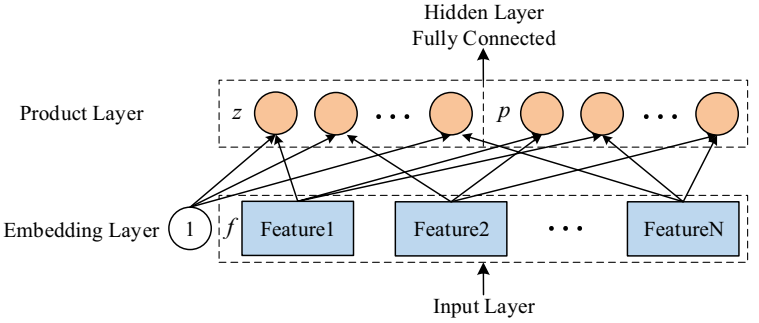

Figure 7. *Structure diagram of product layer*

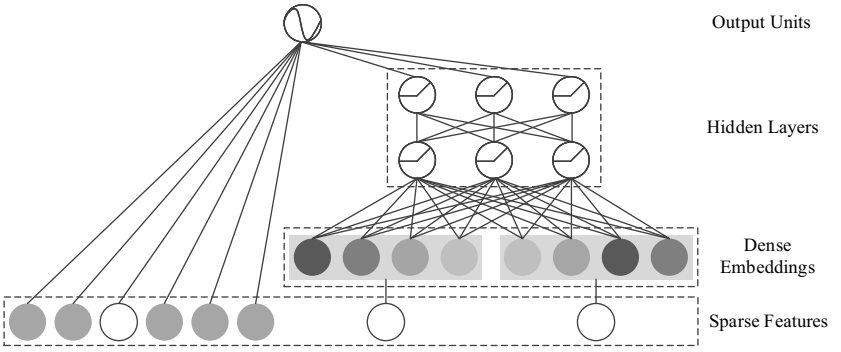

Figure 8. *Structure diagram of Wide & Deep*

1. **IPNN:** The model input is the result of the inner product between embedding vectors, and the model complexity caused by the calculation of pairwise vector product will be very high. Therefore, the weight W_p^n in the formula is decomposed by using the idea of FM: $\mathbf{W}_p^n = \theta_i^n \theta_j^n$, the formula can be transformed into:

$$l_p^n = \mathbf{W}_p^n \odot p = \sum_{i=1}^{N} \sum_{j=1}^{N} \theta_i^n \theta_j^n \langle f_i, f_j \rangle = \langle \sum_{i=1}^{N} \delta_i^n, \sum_{i=1}^{N} \delta_i^n \rangle, \tag{10}$$

   where $\delta_i^n = \theta_i^n f_i \in \mathbb{R}^M$.
2. **OPNN:** The model input is the result of the outer product between the pairwise embedding vectors. The outer product operation will increase the complexity of the problem from $O(M)$ to $O(M^2)$. In order to reduce the complexity of the model, the results of all outer product operations can be superimposed into $M \times M$, that is, p is converted into:

$$p = \sum_{i=1}^{N} \sum_{j=1}^{N} f_i f_j^T = f_M (f_M)^T, f_M = \sum_{i=1}^{N} f_i. \tag{11}$$

Deep learning networks alone can capture high-order feature interactions, but they often overlook the importance of low-order feature combinations. Both the FNN and PNN models account for high-order feature interactions, yet their memory capacity is limited due to the neglect of low-order features. In deeper network architectures, the increased depth enables more complex feature interactions, but this can lead to the loss of simpler information provided by the initial features.

In 2016, Google researchers (Cheng *et al*., 2016) proposed the Wide&Deep model to combine linear models and deep learning models, not only considering low-level information but also learning the interactive information between features. The structure of the Wide&Deep model is shown in Figure 8, wide part is a linear model (generally LR) to provide memory for the whole model; DNN, as the deep part,

mines high-order nonlinear features to increase the generalization ability of the model. The influence of Wide & Deep is to put forward a form that can be combined, which combines the simple model with the deep neural network, so as to strengthen the memory ability and generalization ability. In view of the defect that the wide part of Wide& Deep does not have the ability of automatic feature combination, the DeepFM model proposed by Guo *et al*. (2017) uses a parallel structure to combine FM and DNN, both of which receive the same input, but learn different features (one is responsible for low-level interaction, the other is responsible for high-level interaction). As the FM part of the DeepFM is still a second-order crossover, it inevitably limits the expression ability of the model. Hence, He and Chua (2017) proposed Neural Factorization Machine (NFM) to extract the nonlinear interactive information of high-order features. The structure of NFM network is similar to that of PNN network. The structure of NFM network is similar to that of PNN network, which changes the product layer of PNN into Bi-Interaction pooling Layer to realize the seamless connection between FM and DNN.

Chen *et al*. (2019b) proposed the field-leveraged embedding network (FLEN), which uses space-time efficient methods to alleviate the widespread gradient coupling problem, mainly using field-wise bilinear interaction (FwBI) (including three parts: Linear, FM, and MF), in which the MF part is used to learn the feature interaction among the large categories of features (user, item, and context), and the FM part is used to learn the feature interaction within the large categories of features. Reference (Zhao *et al*., 2021b) proposes that Field-aware INTeraction Neural Network (FINT) for CTR prediction uses the Field-aware INTeraction layer to capture high-order feature interactions while preserving low-order field information.

In addition to the models or composite models mentioned above, there are also models that directly use multi-layer perceptron (Gardner & Dorling, 1998) to learn the interaction between features. The deep crossing model (Shan *et al*., 2016) consists of embedding layer, stacking layer, multiple residual units, and a scoring layer. Through the multilayer residual network, all dimensions of feature vectors are fully crossed and combined, so that the model can capture more nonlinear features and combined feature information, and increase the expression ability of the model. Zhu *et al*. (2017) put forward the deep embedding forest (DEF) model by replacing the residual network in the deep crossing model with the forest layer. Compared with the deep crossing, this model can effectively reduce the online prediction time.

Feature engineering plays an important role in CTR prediction accuracy, and identifying common, predictive features while exploring unseen or rare intersecting features is the key to making good predictions. Wang *et al*. (2017) proposed the Deep&Cross Network (DCN) for CTR prediction. The model consists of deep neural network and cross network, and the outputs of the two networks are combined as the input of the CTR prediction model. The purpose of designing cross network is to increase the interaction strength between features, and the time and space complexity of the network are linear. Cross network consists of multiple cross layers, assuming that the output vector of the l-th layer is x_l, then the output vector of the $l+1$-th layer is:

$$x_{l+1} = x_0 x_l^T w_l + b_l + x_l = f(x_l, w_l, b_l) + x_l, \tag{12}$$

the visualization of cross layer is shown in Figure 9 (Wang *et al*., 2017), the cross network can perform high-order feature interaction. The number of layers of the network determines the order of feature interaction, the highest cross product order corresponding to the l-th layer feature is $l+1$. It can be seen that each layer adds a n-dimensional weight vector w_l (n represents the dimension of the input vector) and retains the input vector at each layer, so the change between input and output will not be particularly obvious. Since Equation (12) is used for feature interaction learning, it can be seen that x_{l+1} is iteratively derived from x_0, so it will be more sensitive to the parameters of each layer. Due to insufficient sharing of hidden layer of DCN, and excessive network input sharing limits the expressiveness of the models. To enhance information sharing between explicit and implicit feature interactions, Chen *et al*. (2021) proposed the Enhanced Deep&Cross Network (EDCN). In EDCN, the bridge module mainly solves the problem of insufficient sharing of the hidden layer of DCN model and increases the interaction between parallel structures. and the regulation module generates different embeddings for different

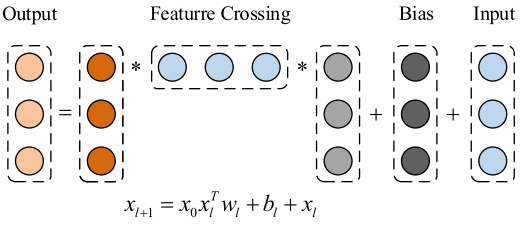

Figure 9. *Visualization of cross layer*

parallel networks, and is used again after each interaction to generate different embeddings. Explicit feature interaction modeling can help neural networks reduce the number of parameters and achieve better performance. However, because of the complexity of the calculation, the explicit feature interactions are often limited to the second order. Literature (Xue *et al*., 2020) also proposes efficient methods to express explicit higher-order feature combinations and simultaneously prune redundant features. To better model complex feature interactions, Xu *et al*. (2021b) proposed the DisentanglEd Self-atTentIve NEtwork (DESTINE) framework for CTR prediction, which explicitly separates the computation of unary feature importance from pairwise interaction.

Lian *et al*. (2018) put forward the eXtreme Deep Factorization Machine (xDeepFM) model with compressed interaction network (CIN) to learn explicit high-order interaction. The CIN module replaces the bit-wise mode of ordinary DNN with vector-wise, which retains the advantages of high-order interaction, automatic cross-multiplication, and parameter sharing of cross network. The output of the k-th layer in CIN is matrix $\mathbf{X}^k \in \mathbb{R}^{H_k \times D}$, where H_k represents the number of feature vectors in the k-th layer and let $H_0 = m$, $\mathbf{X}^k$ is calculated as follow:

$$\mathbf{X}_{h,*}^k = \sum_{i=1}^{H_{k-1}} \sum_{j=1}^{m} \mathbf{W}_{ij}^{k,h} \left(\mathbf{X}_{i,*}^{k-1} \circ \mathbf{X}_{j,*}^0 \right), \tag{13}$$

where $1 \leq h \leq H_k$, $\mathbf{W}^{k,h} \in \mathbb{R}^{H_{k-1} \times m}$ is the parameter matrix of the h-th feature vector, and $\circ$ is the Hadamard product: $\langle a_1, a_2, a_3 \rangle \circ \langle b_1, b_2, b_3 \rangle = \langle a_1 b_1, a_2 b_2, a_3 b_3 \rangle$. Finally, the linear module, CIN module, and DNN are combined to complement each other, providing low-order features, explicit high-order features and implicit high-order features, respectively, to form xDeepFM. Literature (Liu *et al*., 2020a) models automatic feature grouping of explicit high-order feature interaction in CTR prediction.

In 2017, Xiao *et al*. (2017) added the attention mechanism to the NFM (He & Chua, 2017) and proposed the attentive factorization machines (AFM) model. In the pair-wise interaction layer, the weights of the cross features of the NFM model are all 1, without considering the influence degree of different features on the results, while AFM can learn the different influence degrees of different cross features on the results. That is, an attention net is added between pair-wise interaction layer and output layer, and the formula is as follow:

$$f_{Att}\left(f_{PI}(\varepsilon)\right) = \sum_{(i,j) \in \Re_x} a_{ij} \left(v_i \odot v_j \right) x_i x_j, \tag{14}$$

where a_{ij} represents the attention score of the $v_i \odot v_j$, indicating the importance of the interaction feature to the predicted target. Intuitively, this attention score can be used as a parameter to learn by minimizing the prediction loss, but it is impossible to estimate the attention score of the interaction for features that have never been common in the training data. In order to solve the generalization problem, a multilayer perceptron is used to parameterize the attention score. The structure of the attention network is a simple single full connection layer plus softmax output layer.

AFM is a great attempt of attention in the recommender system, but it does not use specific application scenarios. Zhou *et al*. (2018) added activation unit to learn the distribution of user interest on the basis of the basic model (Embedding & MLP) to improve CTR. This model is called deep interest network (DIN), this is Alibaba's model improvement from the perspective of practical application based on

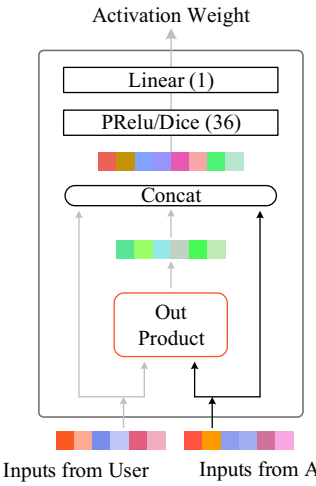

Figure 10. *Structure of activation unit*

business observation in 2018, and accords with the principle of innovation guided by actual needs. The structure of activation unit is shown in Figure 10: one of the most important features of the DIN is user behavior features, that is, the product features that the user has purchased or clicked on in the past. If many of the user's historical products are related to the current product, then the product may be in line with the user's taste, so recommend the advertisement to him. The activation unit structure makes a pairwise interaction between each record in the historical commodity and the commodity to be recommended, and calculates the correlation degree. The input of activation unit is the historical behavior commodity of each user and the current candidate commodity, and the output is the weight calculated by the correlation between the two. The user's interest is expressed as formula (15):

$$v_U(A) = f\left(v_A, e_1, e_2, \ldots, e_H\right) = \sum_{j=1}^{H} a\left(e_j, v_A\right) e_j = \sum_{j=1}^{H} w_j e_j, \tag{15}$$

where v_A is the embedding vector of the candidate advertisement A, $\{e_1, e_2, \ldots, e_H\}$ is the list of historical behavior embedding vectors of the user u, and the length is H, $a(e^j, v_A) = w^j$ indicates the weight or the correlation between the historical behavior commodity and the current advertisement A, $a(\cdot)$ is a feedforward neural network, the output is the activation weight, the input not only the historical behavior vector and candidate advertisement vector, but also their Hadamard product (the corresponding position elements are multiplied and not added) are added to the subsequent network, which is helpful to the explicit knowledge of association modeling. It should be noted that in order to retain the intensity of user interest, the attention score is taken as the final weight coefficient, and softmax normalization is not done. In recent years, some deep learning models that can automatically extract user interests from user behavior have achieved great success. In these works (Zeng *et al*., 2020; Cao *et al*., 2021; Huang *et al*., 2021b), the attention mechanism is used to select items of interest to users from historical behaviors to improve the performance of CTR predictors. Literature (Cheng & Xue, 2021) found that most CTR prediction models can be regarded as a general attention mechanism suitable for feature interaction, so attention mechanism plays a key role in CTR prediction models. Literature (Zhang *et al*., 2021a) proposes a multi-interactive attention network (MIAN) to comprehensively extract the potential relationships among various fine-grained features (such as gender, age and occupation in user profiles). The model includes a multi-interaction layer for fine-grained feature interaction learning and a Transformer-based module to extract multiple representations of user behaviors in different feature subspaces. Dual inputaware factorization machines (DIFMs) model proposed by Lu *et al*. (2021) can

adaptively learn different representations of given features according to different input examples. The automatic interaction machine (AIM) proposed in the literature (Zhu *et al*., 2021) has a similar idea.

The feature interaction method in CIN network is similar to the cross network in Deep&Cross, and each feature interaction uses input variables. Unlike FM, FM is a pairwise feature interaction of variables, and the CIN network fuses all variables into a matrix for feature interaction. The Co-Action in the Co-Action Network (CAN) proposed by Zhou *et al*. (2020) is a new feature interaction method. When there is an association between user and item, the data processed by Co-Action and the original data are simultaneously input to the depth learning models to improve CTR prediction.

Huang *et al*. (2019) pointed out that the current work of CTR prediction through feature combination mainly uses the inner product or hadamard product of feature vectors to calculate cross features. This method ignores the importance of the feature itself, and further proposes the feature importance and bilinear feature interaction network (FiBiNET) model, in which the importance of dynamic learning features using squeeze-and-excitation Nnetwork (SENET) structure and the use of a bilinear function to better establish cross features. Three kinds of bilinear functions, called Bilinear-Interaction layer, have been proposed in the literature. Taking the i-th field embedding v_i and the j-th field embedding v_j as examples, the bilinear interaction can be expressed as:

$$p_{ij} = v_i \cdot W \odot v_j, \tag{16}$$

where $W \in R^{k \times k}$, and $v_i, v_j \in R^k$ are the i-th and j-th field embedding. Literature (Kaplan *et al*., 2021) proposes dynamic length factorization machines (DLFM) for CTR prediction to dynamically optimize the user vector structure and provide better representation for each feature and each pair of features under the constraint of maximum vector length.

In 2019, the DeepMCP model proposed by Ouyang *et al*. (2019b) is different from the previous CTR prediction model. It includes three parts (a matching subnet, a correlation subnet, and a prediction subnet) to model the user-ad, ad-ad and feature-CTR relationships, respectively. Aiming at the sorting problem in CTR prediction, Lyu *et al*. (2020) combined with the idea of collaborative filtering, proposed deep match to rank (DMR) model, emphasizing the importance of capturing the correlation between users and items. Wu *et al*. (2020) proposed a tensor-based feature interaction network (TFNet) model, which introduces an operation tensor to describe the feature interaction through multi-slice matrices in multiple semantic spaces. Mishra *et al*. (2021) proposed an ad text-to-CTR prediction model based on BERT (Mozafari *et al*., 2020), which uses the Ad Text Strength Indicator of Text-to-CTR and Semantic-Ad-Similarity.

The cold-start problem (Schein *et al*., 2002) is a common and unavoidable challenge in recommender systems. Specifically, it arises when a new user is introduced, posing the question of how e-commerce platforms can personalize product recommendations, or how short video platforms can tailor video suggestions, in the absence of user data. Literature (Cao *et al*., 2020) frames cold-start click-through rate (CTR) prediction as a meta-learning problem, treating each advertisement as an individual task. An adaptive loss function is then proposed to address task diversity and distributional shifts. The ultimate aim is to enhance CTR prediction performance in cold-start scenarios. In Table 3, we summarize the key features of representative deep neural network (DNN)-based ad CTR prediction models, comparing aspects such as input sources, shallow models, attention mechanisms, auxiliary loss functions, and overall model architectures.

In the common CTR prediction models, only target advertisements are used for CTR prediction. Ouyang *et al*. (2019a) use the contextual ads, clicked ads, and unclicked ads information auxiliary models of auxiliary advertisements to improve CTR. Three different processing methods are used for embedding matrix, including Pooling, Self-Attention, and Interactive Attention. Finally, three different CTR prediction models (DSTN-P, DSTN-S, DSTN-I) are obtained. Li *et al*. (2020b) proposed the Interpretable Hierarchical Attention (InterHAt) model, after embedding the layer, InterHAt joins the transformer network and uses the multi-layer attention mechanism to increase the interpretability of the network. Multi-head attention (Voita *et al*., 2019) divides the entire attention space into multiple

Table 3. *Summary of the representative DNN based ad click-through rate prediction model. Specifically, X, X_u, X_b, X_{cont}, X_t and X_n represent the input feature vector containing multiple fields, the user, the user behavior, the context, the target ad and the negative ad respectively. '+' in the Model Framework indicates that the two models are combined in parallel, and '→' indicates transmission. Missing values in the table are represented by '-'*

Approach	Inputs	Shallow	Attention	Aux	Model Framework
Shan *et al*. (2016)	X	–	–	–	Multi-Residual Units
Cheng *et al*. (2016)	X	–	–	–	LR + MLP
Qu *et al*. (2016)	X	Inner & Outer Product	–	–	shallow → MLP
Guo *et al*. (2017)	X	FM	–	–	FM + MLP
He and Chua (2017)	X	FM	–	–	FM → MLP
Chen *et al*. (2019b)	X_u, X_b, X_{cont}	FM & MF	–	–	FM + MF + MLP
Zhu *et al*. (2017)	X	Boosting (i.e. XGBoost)	–	–	Boosting→MLP
Wang *et al*. (2017)	X	Cross Network	–	–	(Cross + MLP) → MLP
Lian *et al*. (2018)	X	CIN	–	–	CIN + MLP
Zhou *et al*. (2020)	X_u, X_b, X_t, X_{cont}	CAN	–	–	(Seq-CAN + no-Seq-CAN + DIEN) → MLP
Xiao *et al*. (2017)	X	$\odot$ & FM	Att	–	FM → Att
Huang *et al*. (2019)	X	SENET & Bilinear	–	–	(Bilinear + (SENET → Bilinear)) + MLP
Ge *et al*. (2018)	X_u, X_b, X_t	Inner Product	Att	–	MLP + (Att → MLP)
Zhou *et al*. (2018)	X_u, X_b, X_t, X_{cont}	$\odot$	–	–	$\odot$ → MLP
Pi *et al*. (2020)	X_u, X_b, X_t, X_{cont}	Inner Product	Multi-Att	✓	((GSU → ESU) + DIEN) → DNN
Zhao *et al*. (2020)	X	DRM & Field-wise Module	–	–	DRM + MLP + (DRM → Field-wise Module)
Huang *et al*. (2021a)	X_u, X_b, X_t, X_{cont}	–	Att & Transf	–	(MLP + (Att → Transf)) → MLP
Qin *et al*. (2020)	X_u, X_b, X_t, X_{cont}	–	Self-Att & Att	–	(Att → MLP) → MLP
Ouyang *et al*. (2019a)	X_u, X_b, X_t, X_{cont}	–	Self-Att	–	Self-Att → MLP
Li *et al*. (2020b)	X	–	Transf & Att	–	Transf → Att $\cdots$ Att
Ouyang *et al*. (2020)	X_u, X_b, X_t, X_{cont}	–	Att	✓	(Att → Att → MLP) + (Att → MLP)
Ouyang *et al*. (2019b)	$X_u, X_b, X_t, X_{cont}, X_n$	–	–	–	MLP + MLP + MLP
Lyu *et al*. (2020)	$X_u, X_b, X_t, X_{cont}, X_n$	–	Att	✓	(Att + Att) → MLP

Table 3. *Continued*

Approach	Inputs	Shallow	Attention	Aux	Model Framework
Shi and Yang (2020)	X	–	Self-Att	–	(Self-Att →(+) Self-Att →(+) Self-Att) → MLP
Zhao *et al*. (2021a)	X	–	–	–	–
Xue *et al*. (2020)	X	$\odot$	–	–	Auto-Hash → MLP
Lu *et al*. (2021)	X	FM	M-Self-Att	–	(M-Self-Att + MLP) → FM
Xu *et al*. (2021b)	X	–	M-Self-Att	–	M-Self-Att → MLP
Huang *et al*. (2021b)	X_u, X_b, X_t, X_{cont}	–	Att	–	Att → Att → MLP
Liu *et al*. (2020a)	X	FM	–	–	AutoGroup → FM → MLP
Chen *et al*. (2021)	X	Cross Network	Self-Att	–	(Cross + MLP) → GatNet → (Cross + MLP)
Mishra *et al*. (2021)	X_t	–	–	–	BERT → MLP
Zhao *et al*. (2021b)	X	$\odot$	–	–	Multi-$\odot$ → MLP
Wu *et al*. (2020)	X	TFI	–	–	(TFI + MLP) +MLP
Kaplan *et al*. (2021)	X	DLFM	–	–	DLFM
Guo *et al*. (2021a)	X	–	–	–	AutoDis → MLP
Zhang *et al*. (2021a)	X_u, X_b, X_t, X_{cont}	–	M-Self-Att & Att	–	(M-Self-Att + Att) → MLP

attention subspaces, which has stronger expression ability. There are three ways to use multi-head attention: encoder-decoder attention, encoder self-attention and decoder masked self-attention. In principle, Multi-head is equivalent to introducing more nonlinearity to enhance the expression ability of the model under the condition that the overall calculation cost remains unchanged. A multi-head self-attention-based transformer can capture rich pairwise feature interaction and learn the diversity and polysemy of feature interaction in different semantic subspaces, that is, the diversity meaning of CTR in different click through rate contexts. Given input matrix $\mathbf{X}_0$, the potential expression $\mathbf{H}_i$ of transformer head i is

$$\mathbf{H}_i = \text{softmax}_i\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}}\right)\mathbf{V}, \tag{17}$$

$$\mathbf{Q} = \mathbf{W}_i^{(Q)}\mathbf{X}_0, \quad \mathbf{K} = \mathbf{W}_i^{(K)}\mathbf{X}_0, \quad \mathbf{V} = \mathbf{W}_i^{(V)}\mathbf{X}_0, \tag{18}$$

where matrix $\mathbf{W}_i^{(Q)} \in \mathbb{R}^{d_K \times d}$, $\mathbf{W}_i^{(K)} \in \mathbb{R}^{d_K \times d}$, and $\mathbf{W}_i^{(V)} \in \mathbb{R}^{d_K \times d}$ is the weight parameters of head i, d_K represents the dimension of $\mathbf{K}$ and $\mathbf{H}_i \in \mathbb{R}^{d_K \times m}$. Previous work mainly focused on single-domain CTR prediction, but advertisements are usually displayed as natural content, which provides opportunitys for cross-domain CTR prediction. In order to effectively use news data to predict the CTR of advertising, Ouyang *et al*. (2020) proposed a mixed interest network (MINET), which combines three types of user interests.

Shopping, looking for delicious food, etc. will use the search function. The items currently searched will be the same as those in the history. Then, through the current search, mining similar parts in the history and adding them to the recommended items will greatly improve the user experience. The Search-based Interest Model(SIM) proposed by Pi *et al*. (2020) divides the modeling of long-sequence user behavior features into two modules, namely, General Search Unit (GSU) and Exact Search Unit(ESU). GSU is responsible for screening candidate behaviors related to the current target advertisement from all user behavior queues. ESU uses the filtered information for effective modeling on this basis. The User Behavior Retrieval for CTR prediction (UBR4CTR) (Qin *et al*., 2020) model has the same purpose. UBR4CTR model retrieves a certain number of behavior sequences from the user's historical behavior according to the target predicted by CTR. The target here consists of three parts, target item, target user and other associated content context. Then the model is used to extract the features of the most relevant subsequences from the user's historical long behavior sequence, and finally these features will be used to complete the prediction task of CTR.

Over the past decade, the rapid development of e-commerce and mobile internet has led to a significant surge in the number of mobile applications. The emergence of e-commerce platforms such as Taobao, JD.com, and Douyin has introduced diverse forms of source data for advertisements, which hold considerable research significance. These platforms generate vast amounts of user interaction and behavioral data, which can be leveraged to enhance the accuracy and relevance of advertising recommendations. CTR prediction typically encompasses three primary recommendation modes, each designed to address different real-world recommendation scenarios, thereby offering tailored solutions for various types of users, content, and contextual conditions.

1. CTR prediction scenarios related to pictures

Ge *et al*. (2018) proposed the deep image CTR model (DICM) using pictures as one of the data sources. DICM uses the pictures clicked by users and the pictures in advertisements to predict CTR. When using pictures for training and predicting, it causes excessive bandwidth problems when embedding pictures, so the advanced model server (AMS) (Tusch, 2002) architecture is proposed in this paper to solve this problem. Add a learnable MLP of the compression model $\{4096 - 256 - 64 - 12\}$ for each server. When worker requests image embedding from server, the compression model on server first compresses the original 4096-dimensional image embedding to 12-dimensions, which greatly reduces the traffic. The compression model parameters on each server can be learned according to the locally stored graph data. At the end of each iteration, all server compression models need to be synchronized to ensure that the compression models on each server are consistent.

2. CTR prediction scenarios related to position

The dimension relation module (DRM) model proposed by Zhao *et al*. (2020) includes two subnetworks (Item-to-Item network and user-to-item network) and adds the location information of each behavior, which pays more attention to the recent behavior of users, so it can better predict CTR. When ordering takeout or looking for location-related services such as food on some platforms, adding location information and context information to the CTR model can greatly improve the prediction performance of the model. The common CTR prediction model uses the results obtained from the embedding layer for learning feature interaction, which will bring two shortcomings: one is that the importance of dimension in different field is not considered; the other is that the interaction between features is ignored. Therefore, Zhao *et al*. (2020) put forward field-wise and element-wise based on DRM (FED-net) to solve the shortcomings caused by the direct use of embedding. First of all, dimension relation module (DRM) is proposed in FED-net to solve the deficiency one (the importance of dimension in different field), and then the Field-wise module is designed to solve the deficiency two (interaction between features). The use of two different network structures to help solve the shortcomings caused by the direct use of the embedding layer will have a great impact on future research. Huang *et al*. (2021a) proposed that deep position-wise interaction network (DPIN) model uses multi-source data and adds attention mechanism to learn the potential interest of users' location to help the platform to better push satisfactory services to users.

3. CTR prediction scenarios related to video

The video click-through rate prediction studied in document (Wang *et al*., 2020a) solves the multi-channel problem in video CTR prediction for the first time, which is very important for the refinement of video recommendation and the revenue of video advertising. In this paper, sequential multi-fusion network (SMFN) is proposed to divide all channels into two categories: (1) the target channel to which the current candidate video belongs. (2) the context channel including all left channels. For each category, SMFN deeply fuses the two sequences through a simple but effective mechanism, and verifies that the fusion unit helps to improve the CTR prediction performance. Min *et al*. (2022) propose neighbor interaction-based CTR prediction (NI-CTR) model. The model is actually deployed to the online recommendation scene of wechat official account video. The proposed modeling neighborhood information improves the performance of CTR prediction.

The effective integration of high-level and low-level features remains an underexplored area of research. Some studies attempt to combine these features through simple summation or concatenation. However, this approach often yields suboptimal results, as it treats high-level and low-level features with equal importance, without accounting for their inherent differences in significance and abstraction. The hybrid feature fusion (HFF) model proposed by Shi and Yang (2020) is composed of feature interaction layer and feature fusion interaction. It can not only capture high-level features but also make full use of low and high level features. Model integration is a powerful means to improve the prediction accuracy. Literature (Zhu *et al*., 2020) attempts to apply knowledge distillation (KD) to ensembled CTR prediction. Zhao *et al*. (2021a) introduced reinforcement learning (Sutton & Barto, 2018) into CTR prediction model, which lays a foundation for the proposal of various evolution models later.

4.2 CTR prediction model based on convolutional neural network

Convolutional neural networks (CNNs) have demonstrated exceptional performance in processing images, videos, and other types of data, and they can also be effectively applied to click-through rate prediction tasks. CNNs are particularly well-suited for feature extraction, leveraging their hierarchical structure to capture both low-level and high-level features from raw data, the most typical of which is the convolutional click prediction model (CCPM) proposed by Liu *et al*. (2015), which calculates continuous features to obtain local features, Then, the obtained feature combination is input into the fully

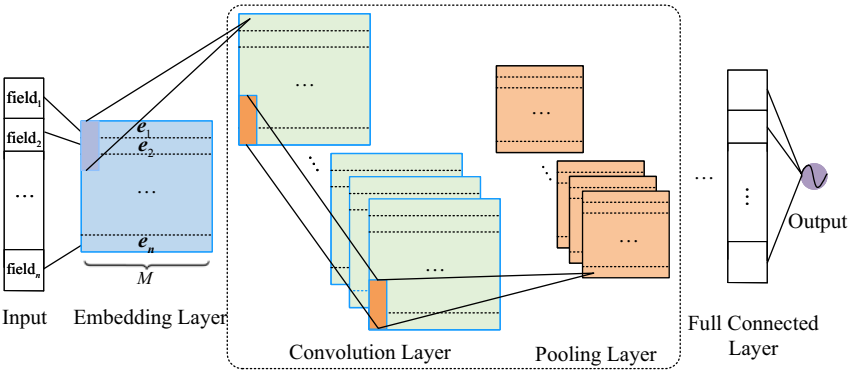

Figure 11. *Basic architecture of applying CNN to CTR prediction*

connected neural network, which improves the learning ability of the fully connected network. The basic architecture of using CNN for the CTR problem is shown in Figure 11 (Chan *et al*., 2018),

the feature field is mapped to a densely structured input space using an embedding layer, that is, the i-th feature field is mapped to e_i, where e_i represents the i-th embedding feature vector of length t, $e = [e_1, e_2, \ldots, e_n](i = 1, 2, \ldots, n)$, n is the number of feature fields. The embedding feature vector is fed into the feature learning layer, including convolution and pooling. Finally, all learned latent features are processed by fully connected layers to predict CTR. Unlike applications in image or natural language processing where the samples have natural sequences, the embedding feature vectors for CTR prediction can be arranged in any order. However, the order in which the embedding feature vectors affects the local information learned by the CNN because the convolutional and pooling layers of the CNN capture information in the local receptive fields.

The distribution of data predicted by CTR varies over time, Chan *et al*. (2018) first investigated whether and how feature sequences affect the performance of CNN-based CTR prediction methods. To learn the information provided by different sequences, two multi-sequence models are proposed: multi-sequence model with single feature learning module (MSS) and multi-sequence model with multiple feature learning modules (MSM). In the MSS model, all feature maps of the MS layer are used as the input of the first convolutional layer: $c_i^0 = [e_{s_{i1}}, e_{s_{i2}}, \ldots, e_{s_{im}}]$, usually the i-th output of the l-th pair of convolutional pooling layers $\mathbf{c}_i^l$ can be defined as Equation (19):

$$\mathbf{c}_i^l = q\left(\sigma\left(\sum_{j=1}^{t_{l-1}} \text{conv}\left(\mathbf{c}_j^{l-1}, \mathbf{w}_{ij}^l\right) + \mathbf{b}_{ij}^l\right)\right), \tag{19}$$

where $q(\cdot)$ and $\sigma(\cdot)$ are the pooling function and activation function, $\mathbf{w}_{ij}$ represents the weight of the i-th filter of the j-th input, $\mathbf{b}$ is the bias term, and t_l is the number of feature maps of the i-th layer. In the MSM model, each feature map in the MS layer is independently learned by a feature learning module. The output of the first pair of convolutional pooling layers can be defined as (20):

$$\mathbf{u}_{ij}^1 = q\left(\sigma\left(\text{conv}\left(\mathbf{c}_j^0, \mathbf{w}_{ij}^1\right) + \mathbf{b}_{ij}^1\right)\right), j = 1, 2, \ldots, n, \tag{20}$$

where $\mathbf{c}_j^0$ represents the j-th feature map of the MS layer. The MSS model first combines the information provided by the multi-sequence embedding feature vectors and is learned by a feature learning module. The time complexity of this model is low, but the feature learning module cannot learn all the information efficiently. So the MSM model is proposed so that the feature vectors embedded in each sequence are learned separately by a feature learning module, and the learned representations are merged into the fully connected layers.

Chen *et al*. (2016) proposed a DeepCTR model based on convolutional structure and multilayer perceptron structure to extract image advertisements as image features and other basic features. The image features are extracted by the convolution layer and further learned by the DNN, and then the

Table 4. *Summary of the representative CNN-based ad click-through rate prediction model. Specifically, X, X_A, X_Q, X_u, and X_{cont} represent the input feature vector containing multiple fields, the ad, the query, the user, and the context, respectively. In the Pooling column, p-max, MOR, and max & avg represent flexible p-max pooling, mean-overtime region pooling, and max and average pooling, respectively. Missing values in the table are represented by '-'*

Approach	Inputs	Conv-Kernel	Pooling	#ConvNetLayer
Liu *et al*. (2015)	X	1D-Conv	p-max	2
Chan *et al*. (2018)	X	2D-Conv	max	–
Chen *et al*. (2016)	X_A, X_Q	2D-Conv	–	17
Shen *et al*. (2016)	X	1D-Conv	MOR	2
Liu *et al*. (2019)	X	1D-Conv	max	–
Liu *et al*. (2020b)	X_A, X_u, X_{cont}	3D-Conv	max & avg	–
Gao *et al*. (2018)	X	1D-Conv	–	1
Niu and Hou (2020)	X	2D-Conv	max	2
Edizel *et al*. (2017)	X_A, X_Q	2D-Conv	max	–
Gligorijevic *et al*. (2019)	X_A, X_Q	2D-Conv	max	–

two features are normalized and input into the multi-layer perceptron. The model achieves performance improvement by combining the two types of features for prediction. Convolutional neural networks have powerful functions in extracting image features, and can also extract text features, so as to better discover latent factors. Shen *et al*. (2016) exploited convolutional neural networks to extract latent factors based on user review text data. Zhou *et al*. (2016) used convolutional neural networks to extract image advertisement features, and further considered users' visual preferences in click-through rate prediction. Lei *et al*. (2016) based on the convolutional neural network to map the latent features of the image and the user's preference features to the same latent space, discover the latent features of the image, and further generate prediction results. Literature (Gligorijevic *et al*., 2019) is the first effective attempt to use click data to learn CTR and semantic embeddings at the same time. In Table 4, we summarize the main characteristics of the representative CNN-based ad click-through rate prediction model. Input sources, the dimension of conv-kernel, the pooling method, and the number of layers of convolution network are compared among various models.

The primary challenge in click-through rate prediction is effectively modeling feature interactions. Many researchers have proposed deep learning models to capture both low-order and high-order feature interactions from raw features. However, many of these meaningful features are sparse, and while manual feature engineering can improve model performance in real-world scenarios, it is often costly and requires extensive domain knowledge. Consequently, there is a need for methods that can automatically expand the feature space, reducing the reliance on manual intervention. Liu *et al*. (2019) proposed a new feature generation model based on convolutional neural network (FGCNN), which consists of two parts: feature generation and deep classifier. Figure 12 is a general framework for automatic feature generation, the raw features are input into the machine learning model (model section in Figure 12) to identify and generate feature interactions between the raw features. The original features are then combined with the new generated features and fed into the deep neural network. In CNN, the design of weight sharing and pooling mechanism greatly reduces the number of parameters required to find important local patterns, and eases the optimization difficulty of the later MLP structure. Assuming that the output of the first convolutional layer is $C^1 \in R^{n_f \times k \times m_c^1}$, the convolutional layer can be expressed as Equation (21):

$$C^1_{p,q,i} = \tanh\left(\sum_{m=1}^{1}\sum_{j=1}^{h^1} E^1_{p+j-1,q,m}\,\mathbb{W}\mathbb{C}^1_{j,1,1,i}\right), \tag{21}$$

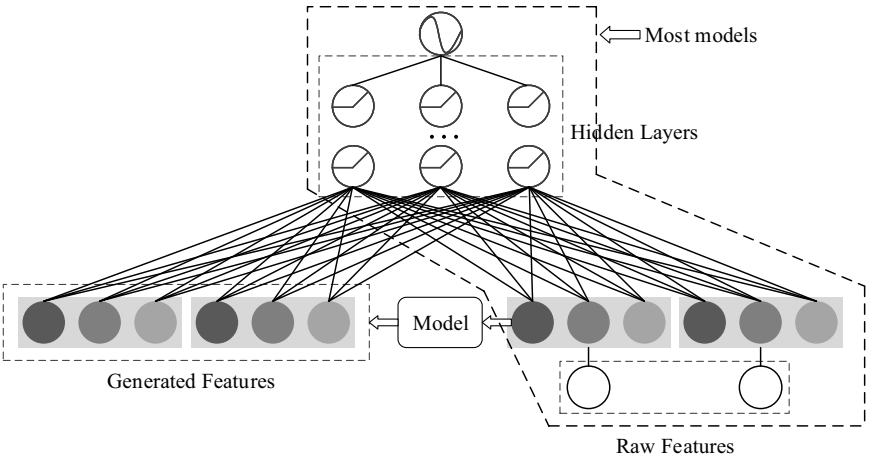

Figure 12. *General framework of FGCNN model*

where $C^1_{p,q,i}$ represents the i-th feature map of the first convolutional layer, and p,q are the row and column indices of the i-th feature map. If only CNN is used, many useful global feature interactions will be lost. Therefore, a complementary approach of CNN and MIP is adopted to extract cross features that are difficult to get from DNN, and then the generated new features and old features are spliced together and input into any other classifier (FM, DNN, IPNN, DeepFM, etc.) to improve the effect.

Most existing studies only focus on user-level CTR prediction, and advertiser-level CTR prediction also plays an important role, because return on investment (ROI) is closely related to advertiser-level click-through rate forecasting. The CTR prediction of advertiser-level can be described as a time series prediction problem based on historical click through records. Literature (Zhu, 2021) proposes a CNN-LSTM convolution hybrid neural network algorithm to predict advertising click through rate. In the modeling process, effective features and combined features are extracted, and prediction and analysis are performed according to the LSTM neural network time series features. Gao *et al*. (2018) proposed a context-aware attention convolutional neural network (CACNN) to capture the highly nonlinear and local information of time series and the potential correlation between CTR time series and context information, so as to obtain more accurate prediction. Literature (Niu & Hou, 2020) proposes a new input instance representation method based on density matrix, which can contain the interaction information of global second-order features. Then, combining the advantages of density matrix and convolutional neural network, a density matrix based convolutional neural network (DMCNN) is proposed, which can capture more feature interactions than other models. Literature (Edizel *et al*., 2017) proposes two new content-based click-through rate prediction model for sponsored search. Both models are based on convolutional neural network structure, which can significantly improve the accuracy and calibration of the model in production.

Attention mechanism is an important feature selection method, which can help CNN to highlight important parts in feature maps and suppress unimportant parts. Many previous works (such as CBAM Woo *et al*., 2018 and GSoP Gao *et al*., 2019) have attempted to learn attention weights from feature maps, called self-attention. A large part of the advertisements of e-commerce application scenarios are displayed in the form of images, existing algorithms usually use CNN to extract visual features and fuse visual and non-visual features together to finally predict CTR. Liu *et al*. (2020b) proposed a new visual embedding module category-specific CNN (CSCNN) for CTR prediction. The core idea is to perform category-specific channel and spatial self-attention to emphasize important and category-related features. CSCNN early combined category knowledge with a lightweight attention module on each convolutional layer. This enables CSCNN to extract expressive class-specific visual patterns that are beneficial for CTR prediction. Literature (Guo *et al*., 2021c) proposes two multi-interest extractors

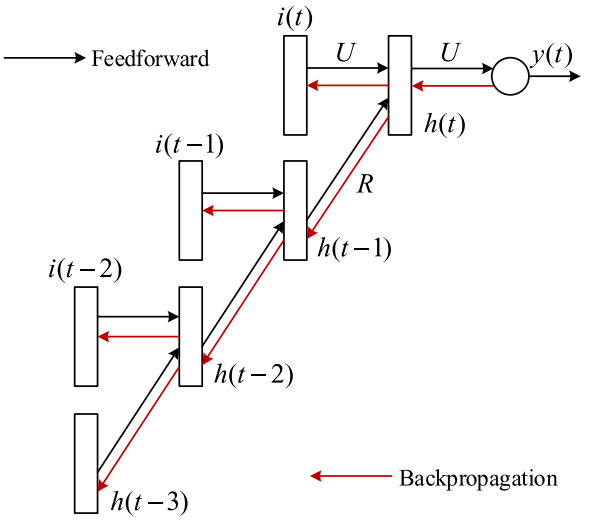

Figure 13. *RNN training process with BPTT algorithm*

based on CNN, which fully consider different interest representation, interest dependence and interest correlation.

4.3 CTR prediction model based on recurrent neural network

Personalization is a key factor in enhancing user experience for click-through rate prediction models. Personalized information is inherently embedded in a user's past behavior. As a result, many models aim to learn a user's current interests by incorporating their behavioral sequence into the modeling process. A user's decision to click on an advertisement is often influenced by a series of prior behaviors, such as previous searches, content clicks, and the time spent on landing pages (Zhang *et al*., 2014). For instance, if a user clicks on an advertisement and quickly closes the landing page, the likelihood of them clicking on an advertisement in the future is significantly reduced. Conversely, if a user searches for flight booking keywords, the probability of them clicking on a flight booking advertisement is much higher. In comparison to shallow models and traditional deep learning approaches, recurrent neural networks (RNNs) are particularly effective in capturing the impact of a user's browsing sequence on CTR prediction (Gan & Xiao, 2019). RNNs excel at identifying latent interests behind user behavior and can track the dynamic evolution of these interests over time.

The sequence of user behavior is ignored in most CTR prediction models. Zhang *et al*. (2014) used a recurrent neural network to model sequential dependencies in predicting ad click probabilities. They treat each user's ad viewing history as a sequence that generates internal dependencies. During the training of the RNN model, the features of each ad impression are fed into the hidden layers along with the previously accumulated hidden states, and order dependencies are incorporated to improve the accuracy of click predictions. The RNN training process of this model adopts the BPTT algorithm (De Jesus & Hagan, 2007), the expansion step is set to 3, and the structure is shown in Figure 13, the network consists of input layer i, output unit, hidden layer h and internal weight matrix. Here, we use $t \in \mathbf{N}$ to denote the timestamp and use $\mathbf{h}(t)$ to denote the hidden state at time t. Specifically, the recurrent connection $\mathbf{R}$ between $\mathbf{h}(t-1)$ and $\mathbf{h}(t)$ can propagate sequential signals. The input layer consists of the vector $\mathbf{i}(t)$ representing the current user behavior characteristics, and the vector $\mathbf{h}(t-1)$ represents the value in the hidden layer calculated from the previous step. The activation values of the hidden layer and output layer are calculated as Equations (22) and (23):

Table 5. *Summary of the representative RNN based ad click-through rate prediction model. Specifically, X, X_u, X_b, X_t, and X_{cont} represent the input feature vector containing multiple fields, the user features, the user behavior features, the target ad features, and the context features, respectively. Att, Multi-Att, and M-Self-Att in the Attention indicate that attention, multi-head attention, and multi-head self-attention, respectively. Missing values in the table are represented by '-'*

Approach	Inputs	RNN-type	Attention Mechanism	Aux
Zhou *et al*. (2019)	X_u, X_b, X_t, X_{cont}	GRU & AUGRU	Att	✓
Pi *et al*. (2019)	X_b, X_t, X_{cont}	GRU	Att	–
Zhang *et al*. (2014)	X	RNN	Att	–
Feng *et al*. (2019)	X	Bi-LSTM	M-Self-Att	–
Song *et al*. (2020)	X	–	–	–
Li *et al*. (2020a)	X_b, X_t, X_{cont}	GRU	Att & Multi-Att	–
Wang *et al*. (2020a)	X_b, X_t, X_{cont}	GRU	–	–
Xu *et al*. (2021a)	X_u, X_b, X_t, X_{cont}	GRU & AUGRU	Att	✓
Hong *et al*. (2021)	X_b, X_t	GRU	Att	✓

$$\mathbf{h}(t) = f\left(\mathbf{i}(t)\mathbf{U}^T + \mathbf{h}(t-1)\mathbf{R}^T\right), \tag{22}$$

$$y(t) = \sigma\left(\mathbf{h}(t)\mathbf{V}^T\right), \tag{23}$$

where $f(\cdot)$ represents the tanh function for nonlinear activation, and $\sigma(\cdot)$ represents the sigmoid function. $\mathbf{i}(t)$ represents the features related to the user's current behavior, and $\mathbf{h}(t)$ represents the sequence information of the user's previous behavior. The prediction results not only depend on the current input features but also on continuous historical information.

Currently, user historical data are a time series, so the recurrent neural network can be used to learn user interests. Sequences of user historical behaviors may contain multiple concurrent interests, and the rapid jumps and abrupt ends of these interests cause the sequence data of user behaviors to be noisy. The deep interest evolution network (DIEN) (Zhou *et al*., 2019) proposed in 2019 is an evolution of the DIN (Zhou *et al*., 2018). Based on the DIN model, a recurrent neural network is introduced to capture sequence information. DIEN utilizes RNN with two layers of gated recurrent unit (GRU) to learn user interests. The first layer is the interest extractor layer, which learns the sequence dependencies between historical sequence behaviors by simulating the user's interest migration process. A GRU training loss, an auxiliary loss, is introduced to supervise the training process of each intermediate hidden state of the GRU, as shown in Equation (24):

$$L_{aux} = -\frac{1}{N}\left(\sum_{i=1}^{N}\sum_{t}\log\sigma(\mathbf{h}_t^i, \mathbf{e}_b^i[t+1]) + \log\left(1-\sigma(\mathbf{h}_t^i, \hat{\mathbf{e}}_b^i[t+1])\right)\right), \tag{24}$$

where $\sigma\left(\mathbf{x}_1, \mathbf{x}_2\right) = \frac{1}{1+\exp(-[\mathbf{x}_1, \mathbf{x}_2])}$ is the sigmoid activation function, $\mathbf{h}_t^i$ represents the t-th hidden state of the GRU of user i. Then the output of the first layer is used as the input of the attentional up-date gate (AUGRU) of the second layer (interest evolving layer) and combined with the attention network to simulate the user's interest migration process related to the target advertisement. The attention network is used to control the update gate of AUGRU in the second layer to make it more targeted to simulate the interest evolution path related to the target advertisement. Finally, the last state in the second layer is input into the DNN as the user's interest to predict the user's CTR. In Table 5, we summarize the main characteristics of the representative RNN based ad click-through rate prediction model. Input sources, the type of RNN, the attention mechanism, and the auxiliary loss function are compared among various models.

Transformer is a feature extractor based on attention mechanism. The transformer architecture includes two parts: encoder and decoder, which can extract the features of sequences instead of CNN

and RNN. Transformer has the following advantages over the recurrent neural network: (1) Long distance dependencies in sequences can be captured directly. (2) The model has high parallelism, which greatly reduces the training time. Most works ignore the inherent structure of user behavior sequences, user behavior sequences are composed of multiple sessions (Hidasi *et al*., 2015), and the sessions are distinguished by the user's click time. A user has a definite and separate need to purchase items within the same session, but his interests will change once a new session is opened. Based on this observation, Feng *et al*. (2019) proposed the deep session interest network (DSIN) that utilizes multiple historical sessions of users to simulate user sequence behavior in CTR prediction tasks. The key part of the DSIN is to model the user behavior sequence, which is divided into four layers from bottom to top: (1) session division layer: divide the user's behavior sequence into multiple sessions according to the click time; (2) session interest extraction layer: for each session, the multi-head self-attention mechanism in transformer is used to extract the interest features of the user session and capture the internal relationship between actions; (3) session interest interacting layer: adopts Bi-LSTM (Huang *et al*., 2015) captures the interaction and evolution of users' interests across multiple historical sessions; and (4) session interest activating layer: applies a local activation unit to the user's session interest about the item. Finally, the output of the session interest activating layer, along with the user portrait embedding and the item portrait embedding, is input into the fully connected layer for final prediction.

The aforementioned studies demonstrate that researchers have long acknowledged the importance of extracting user interests in CTR prediction tasks. Many of these studies treat interactions between users and items as sequential data and apply recurrent neural networks (RNNs) to effectively capture and model user interests. However, these solutions cannot handle relatively long sequence lengths due to the vanishing gradient problem (Hochreiter, 1998) of RNNs. Therefore, Xu *et al*. (2021a) proposed a new core interest network (CIN) to alleviate the long sequence problem of the CTR prediction task for sequence data. The main idea of the model is to extract users' core interests first and the refined data is then used as input for the following learning tasks. The model divides a long sequence into multiple subsequences and extracts the user's core interest in each subsequence and also uses the auxiliary loss shown in Equation (24) to supervise the training process of each intermediate hidden state of the GRU. The core interests extracted from each subsequence are passed to the next subsequence and finally the learning of user interests in the whole long sequence is completed. Li *et al*. (2020a) proposed the deep time-aware item evolution network (TIEN), the mentioned time-aware item behavior extends traditional user behavior, and helps indicates user interest drift and item popularity over time. Reference (Hong *et al*., 2021) proposes a recommendation model that is closer to real recommender scenarios by jointly learning the current and comprehensive interests of users.

Simple recurrent neural networks (RNNs) often struggle to learn from long sequence data. To address this, attention mechanisms can be introduced to enhance the model's expressive power by compressing relevant information from sequential data into fixed-length vectors. However, the computational cost of attention mechanisms grows with the length of the action sequence. Moreover, the hidden state in an RNN does not retain all information from the past sequence; instead, it tends to focus more on the prediction target, potentially overlooking important historical context. Drawing on the idea of neural turing machine (NTM) (Graves *et al*., 2014), Pi *et al*. (2019) proposed the multichannel user interest memory network (MIMN) to deal with long sequences of user behaviors in CTR prediction. MIMN designs an independent user interest center (UIC) module, which separates the bulk user interest computation from the entire CTR prediction process. UIC stores MIMN's external storage information, updates it for each user's behavior, and UIC gradually captures the user's interest from the user's behavior sequence. The core idea of the model is to adopt two designs: (1) Increase a memory utilization regularization to improve the expressiveness of memory tensors in UIC by improving memory utilization. (2) Use a memory induction unit to help capture higher-order information. Reference (Song *et al*., 2020) conducted a preliminary study of automatically designing the architecture for the CTR prediction task.

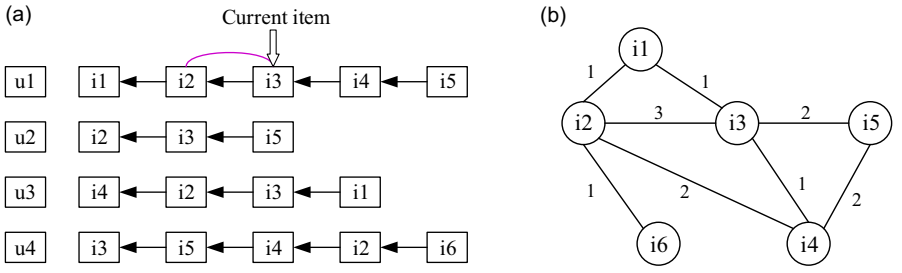

Figure 14. *Structure of co-occurrence commodity graph*

4.4 CTR prediction model based on graph neural network

In recent years, graph neural networks (GNNs) (Scarselli *et al*., 2008) have gained widespread adoption as a deep learning-based approach for processing graph-structured data, owing to their powerful ability to model complex relationships within graph structures. In 2019, Li *et al*. (2019b) pioneered the use of GNNs for modeling intricate interactions between features and proposed the feature interaction graph neural network (Fi-GNN) for CTR prediction. The basic idea is to use a graph structure called feature graph to represent multi-field features. The feature is used as a node of the graph, there is an edge between two nodes, and the weight on the edge represents the importance of feature interaction, so as to transform the complex interaction between features into the interaction between the nodes of the feature graph. In the embedding layer, the model uses a multi-head self-attention network layer to obtain a new field embedding, which contains the high-level feature interaction between the field and other feature fields, and the output feature map is used as the input of the Fi-GNN. The Fi-GNN consists of multiple steps, and each step updates the nodes. The information of neighbor nodes is aggregated using an attention network, and then a GRU unit is used to make state updates for the nodes. In Fi-GNN, each node updates its own state in a cyclic manner by exchanging state information with neighbor nodes, so the number of steps updated on the graph network is equivalent to the order of feature interaction. According to the powerful representation ability of the graph, the model not only can flexibly and explicitly model complex feature interactions but also provides more understandable model interpretation for CTR prediction.

The accuracy of CTR prediction in sponsored search has a key impact on improving business revenue and user experience. Li *et al*. (2019a) proposed a graph intention network (GIN) based on co-occurrence commodity graph to mine user intentions. In previous models, user intentions were mostly extracted based on their historical click behaviors, there will be problems such as user behavior sparsity, weak generalization, and so on. First, the GIN method enriches user behaviors through the multilayer graph diffusion of historical behaviors and solves the problem of sparse user behaviors; Second, by introducing commodity co-occurrence relationships, it explores users' potential preferences and alleviates the weak generalization problem. The construction of the co-occurring commodity graph based on historical behavior is shown in Figure 14, each row in Figure 14(a) represents a user's click sequence, when the window size is 1, the black arrows represent the behavior direction, and the red arrows represent the edges of the graph. In the undirected co-occurrence commodity graph of Figure 14(b), the nodes represent the clicked commodities, and the edge weights represent the number of co-occurrence clicks. By performing multilayer neighborhood diffusion on the graph for each item in the user click sequence, then an attention mechanism is applied to aggregate the tree-like intents. Finally, through end-to-end joint training, the intent mining method based on co-occurrence commodity graph is combined with the CTR prediction task. In Table 6, we summarize the main characteristics of the representative GNN based ad click-through rate prediction model. Input sources, graph info and the attention mechanism are compared among various models.

Feature interaction is critical for achieving high-accuracy recommendations in recommender systems. Graphs provide a more effective data structure for addressing combinatorial problems, making them particularly well-suited for modeling complex interactions between features. In order to make full

Table 6. *Summary of the representative GNN based ad click-through rate prediction model. Specifically, X, X_u, X_b, X_t and X_{cont} represent the input feature vector containing multiple fields, the user features, the user behavior features, the target ad features and the context features, respectively. Att and M-Self-Att in the Attention Mechanism indicates that attention and multihead self-attention, respectively. Missing values in the table are represented by '-'*

Approach	Inputs	Graph info	Attention mechanism
Li *et al*. (2019b)	X	GRU	M-Self-Att & Att
Li *et al*. (2019a)	X_u, X_b, X_t, X_{cont}	Att	Att
Li *et al*. (2021)	X_u, X_b, X_t, X_{cont}	GraphSAGE	–
Guo *et al*. (2021b)	X_u, X_b, X_t	GCN	Att
Feng *et al*. (2020)	X_u, X_b, X_t	Bi-LSTM	
Ouyang *et al*. (2021)	X_u, X_b, X_t	GAT	Att

use of feature interaction, Su *et al*. (2021) proposed a recommendation model based on graph neural network—L_0-SIGN, which detects beneficial feature interaction through graph neural network and L_0 regularization, and only uses beneficial feature interaction for recommendation. L_0-SIGN also constructs a feature graph, where each data sample is treated as a graph, features are nodes, feature interactions are edges, and the weight of edge represent the importance of feature interactions. This is the first time that the problem of detecting beneficial feature interactions in recommender system is proposed and elaborated, and an edge prediction model with L_0 activation regularization is also proposed to automatically detect those beneficial feature interactions in recommendation accuracy, thereby filtering out feature interactions that bring noise. Specifically, the model consists of two components: one component is the L_0 edge prediction model, which detects the most beneficial feature interactions by predicting the presence of edges between nodes. Another component is the graph classification model, called the statistical interaction graph neural network (SIGN). SIGN takes nodes (features) and detected edges (beneficial feature interactions) as input graphs, and outputs predictions by efficiently modeling and aggregating pairs of nodes connected by edges. The general form of the SIGN prediction function is

$$y = f_S\left(G_n\left(X_n, E_n'\right); \boldsymbol{\theta}\right), \tag{25}$$

where θ is the parameter of SIGN, and y is the graph classification result. So the L_0-SIGN prediction function f_{LS} is:

$$f_{LS}\left(G_n\left(X_n, \emptyset\right); \boldsymbol{\theta}, \boldsymbol{\omega}\right) = f_S\left(G_n\left(X_n, F_{ep}\left(X_n; \boldsymbol{\omega}\right)\right); \boldsymbol{\theta}\right). \tag{26}$$

Different from the end-to-end modeling of Fi-GNN, the PCF-GNN proposed by Li *et al*. (2021) is a two-stage model. The first stage is the pre-training of GNN: build a GNN based on feature co-occurrence relationship, nodes represent each feature, and edge weights are feature co-occurrence degrees. During pre-training, the multi-head attention mechanism is not used to learn the initial representation of nodes like Fi-GNN, but the interaction relationship with the output feature is explicitly predicted, and the prediction can also be generalized for new interactions that have not appeared before. The second stage is the downstream application: GNN can use fixed parameters as interactive feature extractor, the value of the interaction feature is first inferred in the application stage, and then spliced together with the remaining features as the input of the subsequent DNN. Or a pre-training paradigm can be used to fine-tune the GNN during the downstream CTR model training process to update the representation of each feature.

Modeling user behavior sequences has attracted a lot of attention, and many existing methods ignore the underlying reasons that drive users to click on target items. Feng *et al*. (2020) proposed a novel Multiplex Target-Behavior Relation enhanced Network (MTBRN) framework to enhance CTR prediction using multiple relationships between user behavior and target items. Multiple relationships contain

semantics that enable better understanding of user interests from different perspectives. MTBRN combines various graphs such as knowledge graph (Wang *et al*., 2014) and item-item similarity graph to build multiple relational paths between user behaviors and target items. Chu *et al*. (2021) put forward Dynamic Sequential Graph Learning (DSGL) method, which enhances the representation of users or items by using the collaboration information in local sub-graphs associated with users or items. In traditional methods, item attributes are regarded as ID features, thus ignoring the dependency between structural information and attributes. In addition, when mining user interests from user product interactions, the current model ignores user intentions and product intentions with different attributes. Zheng *et al*. (2022) proposed hierarchical intention embedding network (HIEN), which considers attribute dependency based on bottom-up tree aggregation in the constructed attribute graph. The hierarchical attention mechanism captures both user and product intentions across different attributes. It represents the relationship between attributes and products (users) using graph and tree structures, exploring attribute dependencies through aggregation methods. Additionally, the attention mechanism integrates with the hierarchy to uncover user and product intentions based on varying attributes.

In recent years, two prevalent techniques for CTR prediction are feature interaction modeling and user interest mining. However, these approaches encounter key challenges: (1) Feature sparsity arises as many features occur infrequently, and feature interaction models rely heavily on feature co-occurrence and (2) user interest mining requires extensive behavioral data to capture diverse interests, but many users have short behavior sequences, leading to sparse behavior data. To address these issues, Guo *et al*. (2021b) proposed the dual graph enhanced embedding module compatible with various CTR prediction models to alleviate these two issues. And further propose dual graph enhanced embedding neural network (DG-ENN) for CTR prediction. A user (item) attribute graph and a collaborative graph are proposed in DG-ENN to alleviate the feature sparsity and behavior sparsity problems. To efficiently learn these graphs, the embeddings are optimized through two well-designed learning strategies: divide-and-conquer and curriculum-learning-inspired organized learning. Literature (Wang *et al*., 2021) puts forward the dependency-aware multi-interest network (Deminet), which explicitly models multiple user interests in CTR prediction task. In order to reduce the noise signal in the behavior sequence, we carry out multi-dependency-aware heterogeneous attention and self-supervised interest learning.

5. Discussions

5.1 Comparison of advantages and disadvantages of algorithms

Different ad click-through rate prediction algorithms exhibit unique advantages and challenges. Table 7 offers a comprehensive summary of the advantages and disadvantages of CTR prediction algorithms based on shallow interactive model, DNN, CNN, RNN, and GNN.

This comparative overview offers valuable insights into the strengths and weaknesses of these models, guiding researchers in selecting the most appropriate approach for CTR prediction tasks.

5.2 Datasets

In the existing literature, ad click-through rate prediction models are often evaluated using various datasets. Table 8 summarizes the datasets used in several studies. It is evident that Criteo, Avazu, and Amazon are the most commonly used public datasets. Proprietary datasets, on the other hand, are sourced from advertising platforms such as Alibaba Cloud, as well as social media platforms (e.g., Tencent, Facebook) and e-commerce sites (e.g., Alibaba, Taobao). Public datasets tend to be more widely used than proprietary ones, likely due to their greater accessibility.

5.3 Model evaluation indicators

Many studies have proposed a range of evaluation indicators for evaluating CTR prediction models. Table 9 presents various evaluation indicators along with the corresponding research references.

Table 7. *Advantages and disadvantages of CTR prediction algorithms based on shallow interactive model, DNN, CNN, RNN, and GNN*

Model Type	Advantages	Disadvantages
The shallow interactive model	1. Simple, efficient, and fast to train, making it suitable for resource-constrained problems. 2. Interpretable, offering a clear linear relationship between features and predictions. 3. Effective in scenarios with simple feature interactions, capturable by linear models	1. Struggles to capture complex feature interactions, limiting performance on high-dimensional datasets. 2. Sensitive to feature scaling, necessitating extensive preprocessing. 3. Underperform in modeling non-linear relationships or higher-order interactions
DNN (Deep Neural Network)	1. Effectively models complex feature interactions, making it suitable for large-scale datasets. 2. Highly flexible, capable of handling various input data types. 3. End-to-end training eliminates the need for manual feature engineering	1. Requires extensive training data and may underperform with sparse data. 2. Lacks interpretability in its internal workings. 3. Computationally intensive, particularly with large datasets
CNN (Convolutional Neural Network)	1. Suitable for data with local structures, where convolutional layers reveal complex patterns. 2. Reduces reliance on manual feature engineering, showing strong adaptability. 3. FGCNN Liu *et al*. (2019) automatically generates expressive features, capturing local dependencies and feature interactions	1. Requires high-quality input data and may need additional preprocessing. 2. Less effective on small or sparse datasets compared to DNN. 3. May fail to capture complex global relationships, focusing primarily on local patterns
RNN (Recurrent Neural Network)	1. Well-suited for sequential data, capturing long-term dependencies, such as DIN (Zhou *et al*., 2018). 2. Effective for modeling user behavior sequences, particularly in CTR prediction with time-dependent features. 3. Adaptable to real-time data, handling temporal variations in user behavior.	1. Susceptible to vanishing/exploding gradient issues. 2. Demands significant computational resources and lengthy training times, especially for large datasets. 3. Struggles to model complex feature interactions, particularly with diverse user behavior sequences
GNN (Graph Neural Network)	1. Effectively models complex feature dependencies, making it suitable for representing interactions as graphs. Addressing cold-start issues and enhancing accuracy for new ads. 2. Well-suited for sparse and high-dimensional data, particularly in CTR prediction tasks. Fi-GNN Li *et al*. (2019b) improves CTR prediction accuracy by modeling feature interactions within a graph structure.	1. High computational complexity, especially with large-scale graph data, resulting in significant resource consumption. 2. Requires careful design of the graph structure, as different construction methods may impact performance. 3. Limited interpretability due to complex dependencies between nodes

Table 8. *The summary of datasets for advertising click-through rate prediction model*

Dataset	Outline		Ciation
	#Instances	#Fields	
Criteo[1]	45M	39	Qu *et al*. (2016), Guo *et al*. (2017), Wang *et al*. (2017), Lian *et al*. (2018), Huang *et al*. (2019), Liu *et al*. (2019), Li *et al*. (2019b), Zhao *et al*. (2020), Li *et al*. (2020b), Juan *et al*. (2016), Song *et al*. (2020), Wu *et al*. (2020), Xue *et al*. (2020), Lu *et al*. (2021), Xu *et al*. (2021b), Liu *et al*. (2020a), Chen *et al*. (2021), Zhao *et al*. (2021b), Zhu *et al*. (2020), Cheng and Xue (2021), Guo *et al*. (2021a), Niu and Hou (2020), Zhu *et al*. (2021)
Avazu[2]	40M	24	Chen *et al*. (2019b), Zhou *et al*. (2020), Huang *et al*. (2019), Liu *et al*. (2015), Chan *et al*. (2018), Liu *et al*. (2019), Li *et al*. (2019b), Zhao *et al*. (2020), Li *et al*. (2020b), Song *et al*. (2020), Wu *et al*. (2020), Xue *et al*. (2020), Lu *et al*. (2021), Xu *et al*. (2021b), Liu *et al*. (2020a), Chen *et al*. (2021), Zhao *et al*. (2021b), Zhu *et al*. (2020), Cheng and Xue (2021), Zhao *et al*. (2021a), Niu and Hou (2020), Zhu *et al*. (2021)
Amazon-books[3]	22W	4	Zhou *et al*. (2020), Shen *et al*. (2016), Zhou *et al*. (2019), Pi *et al*. (2020), Ouyang *et al*. (2020), Pi *et al*. (2019), Cao *et al*. (2021), Hong *et al*. (2021), Xu *et al*. (2021a), Zhang *et al*. (2021a), Wang *et al*. (2021), Guo *et al*. (2021c)
Amazon-Electronics[3]	7.8W	–	Zhou *et al*. (2019), Zhou *et al*. (2018), Cao *et al*. (2020), Cao *et al*. (2021), Xu *et al*. (2021a), Zhang *et al*. (2021a)
Amazon-Other[3]	–	–	Ouyang *et al*. (2020), Zeng *et al*. (2020), Li *et al*. (2020a), Huang *et al*. (2021b), Liu *et al*. (2020b), Cao *et al*. (2021), Chu *et al*. (2021), Guo *et al*. (2021c)
MovieLens Dataset[4]	1M	12	Zhou *et al*. (2018), Cao *et al*. (2020), Shi and Yang (2020), Cao *et al*. (2021), Hong *et al*. (2021), Zeng *et al*. (2020)
KDDCup 2012-track2[5]	200M	–	Song *et al*. (2020), Zhao *et al*. (2021b), Rendle (2012b), Shi and Yang (2020)
iPinYou[6]	19M	–	Zhang *et al*. (2016), Qu *et al*. (2016), Xue *et al*. (2020), Liu *et al*. (2020a)
Avito[7]	2M	27	Ouyang *et al*. (2019a), Ouyang *et al*. (2019b)
Huawei[8]	–	–	Zhu *et al*. (2021)

Table 8. *Continued*

Dataset	Outline		Ciation
	#Instances	#Fields	
Proprietary dataset	–	–	Richardson *et al*. (2007), Chang *et al*. (2010), He *et al*. (2014), Gai *et al*. (2017), Xiao *et al*. (2017), Zhou *et al*. (2018), Zhou *et al*. (2019), Pi *et al*. (2020), Ge *et al*. (2018), Chan *et al*. (2018), Chen *et al*. (2016), Zhou *et al*. (2016), Lei *et al*. (2016), Zhang *et al*. (2014), Li *et al*. (2019a), Su *et al*. (2021), Li *et al*. (2021), Guo *et al*. (2021b), Feng *et al*. (2019), Huang *et al*. (2021a), Qin *et al*. (2020), Ouyang *et al*. (2019a), Zhao *et al*. (2021a), Lyu *et al*. (2020), Feng *et al*. (2020), Li *et al*. (2020a), Huang *et al*. (2021b), Wang *et al*. (2020a), Mishra *et al*. (2021), Ouyang *et al*. (2021), Zeng *et al*. (2020), Liu *et al*. (2020b), Kaplan *et al*. (2021), Gao *et al*. (2018), Edizel *et al*. (2017), Zhu (2021), Gligorijevic *et al*. (2019), Wang *et al*. (2021), Chu *et al*. (2021), Guo *et al*. (2021c), Min *et al*. (2022), Zheng *et al*. (2022)

[1] https://www.kaggle.com/c/criteo-display-ad-challenge.
[2] https://www.kaggle.com/c/avazu-ctr-prediction/dat.
[3] http://jmcauley.ucsd.edu/data/amazon/.
[4] https://grouplens.org/datasets/movielens/20m/.
[5] https://www.kaggle.com/c/kddcup2012-track2.
[6] http://contest.ipinyou.com/.
[7] https://www.kaggle.com/c/avito-context-ad-clicks/data.
[8] https://www.kaggle.com/louischen7/2020-digix-advertisement-ctr-prediction.

Table 9. *Evaluation metrics for CTR prediction model*

Evaluation metrics	Citation
AUC	Blondel *et al*. (2016), Shan *et al*. (2016), Cheng *et al*. (2016), Zhang *et al*. (2016), Qu *et al*. (2016), Gai *et al*. (2017), Zhou *et al*. (2018), Zhou *et al*. (2019), Pi *et al*. (2020), McMahan *et al*. (2013), Guo *et al*. (2017), Chen *et al*. (2019b), Zhu *et al*. (2017), Lian *et al*. (2018), Zhou *et al*. (2020), Huang *et al*. (2019), Ge *et al*. (2018), Chan *et al*. (2018), Chen *et al*. (2016), Liu *et al*. (2019), Zhang *et al*. (2014), Li *et al*. (2019b), Li *et al*. (2019a), Su *et al*. (2021), Li *et al*. (2021), Guo *et al*. (2021b), Feng *et al*. (2019), Zhao *et al*. (2020), Huang *et al*. (2021a), Qin *et al*. (2020), Ouyang *et al*. (2019a), Li *et al*. (2020b), Song *et al*. (2020), Wu *et al*. (2020), Xue *et al*. (2020), Lu *et al*. (2021), Xu *et al*. (2021b), Liu *et al*. (2020a), Chen *et al*. (2021), Zhao *et al*. (2021b), Zhu *et al*. (2020), Cheng and Xue (2021), Guo *et al*. (2021a), Zhao *et al*. (2021a), Shi and Yang (2020), Ouyang *et al*. (2019b), Ouyang *et al*. (2020), Pi *et al*. (2019), Lyu *et al*. (2020), Cao *et al*. (2020), Feng *et al*. (2020), Li *et al*. (2020a), Cao *et al*. (2021), Hong *et al*. (2021), Huang *et al*. (2021b), Wang *et al*. (2020a), Mishra *et al*. (2021), Ouyang *et al*. (2021), Xu *et al*. (2021a), Zeng *et al*. (2020), Liu *et al*. (2020b), Kaplan *et al*. (2021), Zhang *et al*. (2021a), Niu and Hou (2020), Edizel *et al*. (2017), Gligorijevic *et al*. (2019), Wang *et al*. (2021), Chu *et al*. (2021), Zhu *et al*. (2021), Guo *et al*. (2021c), Min *et al*. (2022), Zheng *et al*. (2022)
Logloss	Shan *et al*. (2016), Qu *et al*. (2016), Juan *et al*. (2016), Guo *et al*. (2017), Chen *et al*. (2019b), Zhu *et al*. (2017), Wang *et al*. (2017), Lian *et al*. (2018), Huang *et al*. (2019), Liu *et al*. (2015), Chen *et al*. (2016), Liu *et al*. (2019), Li *et al*. (2019b), Guo *et al*. (2021b), Zhao *et al*. (2020), Qin *et al*. (2020), Ouyang *et al*. (2019a), Li *et al*. (2020b), Song *et al*. (2020), Xue *et al*. (2020), Lu *et al*. (2021), Xu *et al*. (2021b), Liu *et al*. (2020a), Chen *et al*. (2021), Zhao *et al*. (2021b), Zhu *et al*. (2020), Cheng and Xue (2021), Guo *et al*. (2021a), Shi and Yang (2020), Ouyang *et al*. (2019b), Ouyang *et al*. (2020), Cao *et al*. (2020), Feng *et al*. (2020), Li *et al*. (2020a), Huang *et al*. (2021b), Ouyang *et al*. (2021), Kaplan *et al*. (2021), Zhang *et al*. (2021a), Niu and Hou (2020), Wang *et al*. (2021), Chu *et al*. (2021), Zhu *et al*. (2021), Guo *et al*. (2021c), Min *et al*. (2022), Zheng *et al*. (2022)
Relalmpr	Zhou *et al*. (2018), Chan *et al*. (2018), Liu *et al*. (2020a), Zhao *et al*. (2021a), Lyu *et al*. (2020), Cao *et al*. (2021), Wu *et al*. (2020), Mishra *et al*. (2021), Wang *et al*. (2020a), Zhu *et al*. (2021)
RIG	He *et al*. (2014), Qu *et al*. (2016), Zhang *et al*. (2014)
MSE	Richardson *et al*. (2007), Shen *et al*. (2016)
RMSE	Rendle (2010), Qu *et al*. (2016), Xiao *et al*. (2017), He and Chua (2017), Gao *et al*. (2018), Zhu (2021)
Accuracy	Chang *et al*. (2010), Zhou *et al*. (2016), Su *et al*. (2021), Gligorijevic *et al*. (2019)
Precision	Shen *et al*. (2016), Lei *et al*. (2016)
Recall	Lei *et al*. (2016)
F1-score	Li *et al*. (2020a), Zeng *et al*. (2020)

Among these, the most commonly used evaluation indicators are AUC and Logloss. A brief overview of some of these metrics is provided below:

1. **AUC:** AUC measures the probability that a randomly selected positive item ranks higher than a randomly selected negative item. It is the area under the ROC curve (Narkhede 2018). It only

considers the order of prediction instances, which is not sensitive to the problem of imbalance of class. AUC's upper boundary is 1, the bigger the better.

2. **Logloss:** Logloss (Vovk 2015) measures the distance between the predicted scores of each instance and the real label. The lower limit of the Logloss is 0, which means that the two distributions match exactly, the smaller the value, the better the performance. The expression of Logloss with regularization term is as shown in the formula (27),

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) + (1 - y_i)\log\left(1 - \hat{y}_i\right) + \lambda\|\Theta\|_2, \tag{27}$$

where y_i and $\hat{y}_i$ are the true label and estimation value of the i-th instance, respectively. N is the total number of training instances, λ is the weight of L_2 regularization, and Θ is a model parameter set.

3. **RelaImpr:** The introduction of RelaImpr is to estimate the relative improvement of online performance on the basis of offline performance. As shown in the formula (28), the model is compared well with the baseline model. RelaImpr is also known as RI-AUC, the value of AUC for random guess is 0.5, and RelaImpr can be expressed as follows:

$$RelaImpr = [\frac{AUC(\text{model}) - 0.5}{\text{AUC(baseline)} - 0.5} - 1] \times 100\%. \tag{28}$$

4. **Relative Information Gain (RIG):** $RIG = 1 - NE$, where NE is normalized cross entropy, which is expressed as

$$NE = \frac{-\frac{1}{N}\sum_{i=1}^{n}\left(\frac{1+y_i}{2}\log\left(p_i\right) + \frac{1-y_i}{2}\log\left(1 - p_i\right)\right)}{-(p * \log\left(p\right) + (1 - p) * \log\left(1 - p\right))}, \tag{29}$$

where p_i represents the click-through rate estimation value, p is the average experience CTR value.

6. The future research directions

1. **Attention mechanism:** In practical applications, not all feature interactions contribute to improved model performance, and CTR predictions that cannot be explained are often unreliable. Models that automatically capture high-order feature interactions, such as DeepFM (Guo *et al*., 2017) and xDeepFM (Lian *et al*., 2018), require more robust theoretical support. Future research should explore the full potential of attention and pooling methods to analyze the importance of combined features. In the context of online advertising, a deeper understanding of user behavior can significantly enhance CTR prediction. Attention-based CTR models are particularly effective in capturing user interests by leveraging sequential behavioral data. However, since user interests are dynamic and prone to drift, more sophisticated predictive models are needed to better capture the evolving relationship between user behavior and click-through rate.
2. **Graph neural network:** In recent years, several studies have explored the use of GNNs for CTR prediction, primarily employing graph representations for simple feature interactions. GNNs can integrate more powerful feature interactions, such as those found in models like FwFM (Pan *et al*., 2018), FmFM (Sun *et al*., 2021) and AOAFM (Wang *et al*., 2020b), into the graph structure, and apply various aggregation strategies to achieve better CTR prediction performance. Researchers have also focused on developing explicit higher-order models for CTR prediction, such as the deep and cross network (Wang *et al*., 2017) and compressed interaction network (CIN) (Lian *et al*., 2018). However, while explicit representations and high interpretability are valuable, they can sometimes limit the predictive performance of these models. Another promising approach is to leverage GNNs to represent feature interactions directly within the graph structure (Li *et al*., 2019b; Su *et al*., 2021), transforming complex interactions

into node-to-node relationships. This approach suggests that GNNs hold significant potential for advancing the exploration of explicit higher-order models in CTR prediction.

3. **Cold start:** For newly launched advertisements, there is often insufficient historical data to predict clicks effectively. Deep learning models struggle to generate accurate embedding vectors for new ads or ads with limited training samples. To address this, Pan *et al*. (2019) proposed a meta-embedding model that leverages attributes related to new advertisements to mitigate the cold start problem. However, this approach may overlook other valuable information. GNNs offer a solution to the cold start issue by constructing graphs that link various advertisements, enabling the extraction of useful information from adjacent ads. This approach can enhance the click-through rate prediction performance for new advertisements (Ouyang *et al*., 2021). Effectively addressing the cold start problem could provide valuable insights for developing highly interpretable CTR prediction models.
4. **Embedding of numerical features:** Most modeling frameworks for CTR prediction primarily focus on capturing interactions between categorical features, while the embedding of numerical features is often overlooked. The GBDT+LR model (He *et al*., 2014) addresses this by converting numerical features into categorical values using a tree model and then searches for embedding dimensions to obtain their representations. To better handle numerical features in CTR prediction, Guo *et al*. (2021a) proposed the AutoDis framework, a pluggable embedding learning approach for numerical features. AutoDis boasts high model capacity and generates unique representations with a controlled number of parameters in an end-to-end manner.

7. Conclusion

This paper provides a comprehensive overview of ad click-through rate prediction models. We classify CTR prediction models into two main categories: shallow interaction models and deep learning-based CTR prediction models (including DNN, CNN, RNN, and GNN). First, we trace the evolution of classical CTR prediction models in recommender systems, with a focused discussion on representative models from each category. Next, we summarize the advantages and disadvantages of the aforementioned algorithms as well as commonly used datasets and evaluation metrics for assessing the performance of CTR prediction models. Finally, we explore the current research trends in this field and highlight potential directions for future exploration. This paper aims to offer foundational knowledge and identify key areas for further research for scholars interested in CTR prediction.

Author contributions. J. Bai was involved in conceptualization, literature reviews and formal analysis, division and classification, comparison and summary, and writing the original draft. X. Geng took part in conceptualization, project administration and resources, methodology, investigation, writing, reviewing, and editing. J. Deng and Z. Xia took part in conceptualization, resource acquisition, literature reviews, document statistics, reviewing, and editing. H. Jiang, G. Yan, and J. Liang was involved in resource acquisition and formal analysis, reviewing, and editing. All authors have read and agreed to the published version of the manuscript.

Competing interests. All authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Asdemir, K., Kumar, N. & Jacob, V.S. 2012. Pricing models for online advertising: Cpm vs. cpc. *Information Systems Research* **23**(3-part-1), 804–822

Blondel, M., Fujino, A., Ueda, N. & Ishihata, M. 2016. Higher-order factorization machines. In *Advances in Neural Information Processing Systems 29*.

Blum, K. 2012. *Density Matrix Theory and Applications*, **64**. Springer Science & Business Media

Cai, H., Zheng, V. W. & Chang, K. C. C. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* **30**(9), 1616–1637.

Cao, T., Xu, Q., Yang, Z. & Huang, Q. 2020. Task-distribution-aware meta-learning for cold-start ctr prediction. In *Proceedings of the 28th ACM International Conference on Multimedia*, 3514–3522.
Cao, T., Xu, Q., Yang, Z. & Huang, Q. 2021. Meta-wrapper: Differentiable wrapping operator for user interest selection in ctr prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
Chan, P. P., Hu, X., Zhao, L., Yeung, D. S., Liu, D. & Xiao, L. 2018. Convolutional neural networks based click-through rate prediction with multiple feature sequences. *In IJCAI*, 2007–2013.
Chang, Y. W., Hsieh, C. J., Chang, K. W., Ringgaard, M. & Lin, C. J. 2010. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research* **11**(4).
Chapelle, O., Manavoglu, E. & Rosales, R. 2014 Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)* **5**(4), 1–34.
Chen, B., Wang, Y., Liu, Z., Tang, R., Guo, W., Zheng, H., Yao, W., Zhang, M. & He, X. 2021. 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*, 3757–3766.
Chen, J., Sun, B., Li, H., Lu, H. & Hua, X. S. 2016. Deep ctr prediction in display advertising. In *Proceedings of the 24th ACM International Conference on Multimedia*, 811–820.
Chen, Q., Zhao, H., Li, W., Huang, P. & Ou, W. 2019a. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, 1–4.
Chen, W., Zhan, L., Ci, Y., Yang, M., Lin, C. & Liu, D. 2019b. Flen: Leveraging field for scalable ctr prediction. arXiv preprint arXiv:191104690.
Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 7–10.
Cheng, Y. & Xue, Y. 2021. Looking at CTR prediction again: Is attention all you need? In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1279–1287.
Chu, Y., Chang, X., Jia, K., Zhou, J. & Yang, H. 2021. Dynamic sequential graph learning for click-through rate prediction. arXiv preprint arXiv:210912541.
Dahl, G. E., Yu, D., Deng, L. & Acero, A. 2011. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing* **20**(1), 30–42.
De Jesus, O. & Hagan, M. T. 2007. Backpropagation algorithms for a broad class of dynamic networks. *IEEE Transactions on Neural Networks* **18**(1), 14–27.
Edizel, B., Mantrach, A. & Bai, X. 2017. Deep character-level click-through rate prediction for sponsored search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 305–314.
Feng, Y., Lv, F., Hu, B., Sun, F., Kuang, K., Liu, Y., Liu, Q. & Ou, W. 2020. Mtbrn: Multiplex target-behavior relation enhanced network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2421–2428.
Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y. & Yang, K. 2019. Deep session interest network for click-through rate prediction. arXiv preprint arXiv:190506482.
Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.
Gai, K., Zhu, X., Li, H., Liu, K. & Wang, Z. 2017. Learning piece-wise linear models from large scale data for ad click prediction. arXiv preprint arXiv:170405194.
Gan, M. & Xiao, K. 2019. R-rnn: Extracting user recent behavior sequence for click-through rate prediction. *IEEE Access* **7**, 111767–111777.
Gao, H., Kong, D., Lu, M., Bai, X. & Yang, J. 2018. Attention convolutional neural network for advertiser-level click-through rate forecasting. In *Proceedings of the 2018 World Wide Web Conference*, 1855–1864.
Gao, Z., Xie, J., Wang, Q. & Li, P. 2019. Global second-order pooling convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3024–3033.
Gardner, M. W. & Dorling, S. 1998. Artificial neural networks (the multilayer perceptron)–a review of applications in the atmospheric sciences. *Atmospheric Environment* **32**(14–15), 2627–2636.
Ge, T., Zhao, L., Zhou, G., Chen, K., Liu, S., Yi, H., Hu, Z., Liu, B., Sun, P., Liu, H., et al. 2018. Image matters: Visually modeling user behaviors using advanced model server. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2087–2095.
Gligorijevic, J., Gligorijevic, D., Stojkovic, I., Bai, X., Goyal, A. & Obradovic, Z. 2019. Deeply supervised model for click-through rate prediction in sponsored search. *Data Mining and Knowledge Discovery* **33**(5), 1446–1467.
Graves, A., Wayne, G. & Danihelka, I. 2014. Neural turing machines. arXiv preprint arXiv:14105401.
Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. 2018. Recent advances in convolutional neural networks. *Pattern Recognition* **77**, 354–377.
Guo, H., Chen, B., Tang, R., Zhang, W., Li, Z. & He, X. 2021a. An embedding learning framework for numerical features in ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2910–2918.
Guo, H., Tang, R., Ye, Y., Li, Z. & He, X. 2017. Deepfm: A factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:170304247.
Guo, W., Su, R., Tan, R., Guo, H., Zhang, Y., Liu, Z., Tang, R. & He, X. 2021b. Dual graph enhanced embedding neural network for ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 496–504.

Guo, W., Zhang, C., He, Z., Qin, J., Guo, H., Chen, B., Tang, R., He, X. & Zhang, R. 2021c. Miss: Multi-interest self-supervised learning framework for click-through rate prediction. arXiv preprint arXiv:211115068.

He, X. & Chua, T. S. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 355–364.

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 1–9.

Hidasi, B., Karatzoglou, A., Baltrunas, L. & Tikk, D. 2015. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:151106939.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, Ar., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* **29**(6), 82–97.

Hochreiter, S. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**(02), 107–116.

Hong, W., Xiong, Z., You, J., Wu, X. & Xia, M. 2021. Cpin: Comprehensive present-interest network for CTR prediction. *Expert Systems with Applications* **168**, 114469.

Huang, J., Hu, K., Tang, Q., Chen, M., Qi, Y., Cheng, J. & Lei, J. 2021a. Deep position-wise interaction network for CTR prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1885–1889.

Huang, T., Zhang, Z. & Zhang, J. 2019. Fibinet: Combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 169–177.

Huang, Z., Tao, M. & Zhang, B. 2021b. Deep user match network for click-through rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1890–1894.

Huang, Z., Xu, W. & Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:150801991.

Juan, Y., Zhuang, Y., Chin, W. S. & Lin, C. J. 2016. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 43–50.

Kaplan, Y., Koren, Y., Leibovits, R. & Somekh, O. 2021. Dynamic length factorization machines for CTR prediction. In *2021 IEEE International Conference on Big Data (Big Data)*, 1950–1959. IEEE.

Koren, Y. & Bell, R. 2015. Advances in collaborative filtering. In *Recommender Systems Handbook*, 77–118.

Koren, Y., Bell, R. & Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37.

Lei, C., Liu, D., Li, W., Zha, Z. J. & Li, H. 2016. Comparative deep learning of hybrid representations for image recommendations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2545–2553.

Li, F., Chen, Z., Wang, P., Ren, Y., Zhang, D. & Zhu, X. 2019a. Graph intention network for click-through rate prediction in sponsored search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 961–964.

Li, F., Yan, B., Long, Q., Wang, P., Lin, W., Xu, J. & Zheng, B. 2021. Explicit semantic cross feature learning via pre-trained graph neural networks for ctr prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2161–2165.

Li, X., Wang, C., Tong, B., Tan, J., Zeng, X. & Zhuang, T. 2020a. Deep time-aware item evolution network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 785–794.

Li, Z., Cheng, W., Chen, Y., Chen, H. & Wang, W. 2020b. Interpretable click-through rate prediction through hierarchical attention. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 313–321.

Li, Z., Cui, Z., Wu, S., Zhang, X. & Wang, L. 2019b. FI-GNN: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 539–548.

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X. & Sun, G. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1754–1763.

Liu, B., Tang, R., Chen, Y., Yu, J., Guo, H. & Zhang, Y. 2019. Feature generation by convolutional neural network for click-through rate prediction. In *The World Wide Web Conference*, 1119–1129.

Liu, B., Xue, N., Guo, H., Tang, R., Zafeiriou, S., He, X. & Li, Z. 2020a. Autogroup: Automatic feature grouping for modelling explicit high-order feature interactions in CTR prediction. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 199–208.

Liu, H., Lu, J., Yang, H., Zhao, X., Xu, S., Peng, H., Zhang, Z., Niu, W., Zhu, X., Bao, Y., et al. 2020b. Category-specific cnn for visual-aware CTR prediction at jd. com. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2686–2696.

Liu, Q., Yu, F., Wu, S. & Wang, L. 2015. A convolutional click prediction model. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1743–1746.

Lu, W., Yu, Y., Chang, Y., Wang, Z., Li, C. & Yuan, B. 2021. A dual input-aware factorization machine for ctr prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 3139–3145.

Lyu, Z., Dong, Y., Huo, C. & Ren, W. 2020. Deep match to rank model for personalized click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 156–163.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al. 2013. Ad click prediction: A view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1222–1230.

Min, E., Rong, Y., Xu, T., Bian, Y., Luo, D., Lin, K., Huang, J., Ananiadou, S. & Zhao, P. 2022. Neighbour interaction based click-through rate prediction via graph-masked transformer. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 353–362.

Mishra, S., Hu, C., Verma, M., Yen, K., Hu, Y. & Sviridenko, M. 2021. Tsi: An ad text strength indicator using text-to-ctr and semantic-ad-similarity. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 4036–4045.

Mozafari, M., Farahbakhsh, R. & Crespi, N. 2020. Hate speech detection and racial bias mitigation in social media based on bert model. *PloS One* **15**(8), e0237861.

Mu, R. 2018. A survey of recommender systems based on deep learning. *IEEE Access* **6**, 69009–69022.

Narkhede, S. 2018. Understanding auc-roc curve. *Towards Data Science* **26**(1), 220–227.

Niu, T. & Hou, Y. 2020. Density matrix based convolutional neural network for click-through rate prediction. In *2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 46–50. IEEE.

Ouyang, W., Zhang, X., Li, L., Zou, H., Xing, X., Liu, Z. & Du, Y. 2019a. Deep spatio-temporal neural networks for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2078–2086.

Ouyang, W., Zhang, X., Ren, S., Li, L., Zhang, K., Luo, J., Liu, Z. & Du, Y. 2021. Learning graph meta embeddings for cold-start ads in click-through rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1157–1166.

Ouyang, W., Zhang, X., Ren, S., Qi, C., Liu, Z. & Du, Y. 2019b. Representation learning-assisted click-through rate prediction. arXiv preprint arXiv:190604365.

Ouyang, W., Zhang, X., Zhao, L., Luo, J., Zhang, Y., Zou, H., Liu, Z. & Du, Y. 2020. Minet: Mixed interest network for cross-domain click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2669–2676.

Pan, F., Li, S., Ao, X., Tang, P. & He, Q. 2019 Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 695–704.

Pan, J., Xu, J., Ruiz, A. L., Zhao, W., Pan, S., Sun, Y. & Lu, Q. 2018. Field-weighted factorization machines for click-through rate prediction in display advertising. In *Proceedings of the 2018 World Wide Web Conference*, 1349–1357.

Pi, Q., Bian, W., Zhou, G., Zhu, X. & Gai, K. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2671–2679.

Pi, Q., Zhou, G., Zhang, Y., Wang, Z., Ren, L., Fan, Y., Zhu, X. & Gai, K. 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*, 2685–2692.

Qin, J., Zhang, W., Wu, X., Jin, J., Fang, Y. & Yu, Y. 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*, 2347–2356.

Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y. & Wang, J. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining*, 1149–1154. IEEE.

Rawat, W. & Wang, Z. 2017. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation* **29**(9), 2352–2449.

Rendle, S. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*, 995–1000. IEEE.

Rendle, S. 2012a. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* **3**(3), 1–22.

Rendle, S. 2012b. Social network and click-through prediction with factorization machines. In *KDD Cup*.

Richardson, M., Dominowska, E. & Ragno, R. 2007. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, 521–530.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., Monfardini, G. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* **20**(1), 61–80.

Schein, A. I., Popescul, A., Ungar, L. H. & Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 253–260.

Shaheen, F., Verma, B. & Asafuddoula, M. 2016. Impact of automatic feature extraction in deep learning architecture. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1–8. IEEE.

Shan, Y., Hoens, T. R., Jiao, J., Wang, H., Yu, D. & Mao, J. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 255–262.

Shen, X., Yi, B., Zhang, Z., Shu, J. & Liu, H. 2016. Automatic recommendation technology for learning resources with convolutional neural network. In *2016 international symposium on educational technology (ISET)*, 30–34. IEEE.

Shi, Y. & Yang, Y. 2020. Hff: Hybrid feature fusion model for click-through rate prediction. In *International Conference on Cognitive Computing*, 3–14. Springer.

Song, Q., Cheng, D., Zhou, H., Yang, J., Tian, Y. & Hu, X. 2020. Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 945–955.

Su, Y., Zhang, R., Erfani, S. & Xu, Z. 2021. Detecting beneficial feature interactions for recommender systems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*.

Sun, Y., Pan, J., Zhang, A. & Flores, A. 2021. Fm2: Field-matrixed factorization machines for recommender systems. In *Proceedings of the Web Conference 2021*, 2828–2837.

Sutton, R. S. & Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Szegedy, C., Toshev, A. & Erhan, D. 2013. Deep neural networks for object detection. In Advances in Neural Information Processing Systems *26*.

Tusch, R. 2002. *AMS: An adaptive multimedia server architecture*. Inst. of Information Technology, Univ. Klagenfurt.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R. & Titov, I. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. arXiv preprint arXiv:190509418.

Vovk, V. 2015. The fundamental nature of the log loss function. In *Fields of Logic and Computation II*, 307–318. Springer.

Wang, Q., Xing, S., Zhao, X., Li, T., et al. 2018. Research on ctr prediction based on deep learning. *IEEE Access* **7**, 12779–12789.

Wang, R., Fu, B., Fu, G. & Wang, M. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, 1–7.

Wang, W., Zhang, W., Feng, W. & Zha, H. 2020a. Sequential multi-fusion network for multi-channel video ctr prediction. In *International Conference on Database Systems for Advanced Applications*, 3–18. Springer.

Wang, Y., Luo, Q., Ding, Y., Wang, D. & Deng, H. 2021. Deminet: Dependency-aware multi-interest network with self-supervised graph learning for click-through rate prediction. arXiv preprint arXiv:210912512.

Wang, Z., Ma, J., Zhang, Y., Wang, Q., Ren, J. & Sun, P. 2020b. Attention-over-attention field-aware factorization machine. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6323–6330.

Wang, Z., Zhang, J., Feng, J. & Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Woo, S., Park, J., Lee, J. Y. & Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Wu, S., Ren, W., Yu, C., Chen, G., Zhang, D. & Zhu, J. 2016. Personal recommendation using deep recurrent neural networks in netease. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp 1218–1229. IEEE.

Wu, S., Yu, F., Yu, X., Liu, Q., Wang, L., Tan, T., Shao, J. & Huang, F. 2020. Tfnet: Multi-semantic feature interaction for ctr prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1885–1888.

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F. & Chua, T. S. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. arXiv preprint arXiv:170804617.

Xu, E., Yu, Z., Guo, B. & Cui, H. 2021a. Core interest network for click-through rate prediction. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **15**(2), 1–16.

Xu, Y., Zhu, Y., Yu, F., Liu, Q. & Wu, S. 2021b. Disentangled self-attentive neural networks for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3553–3557.

Xue, N., Liu, B., Guo, H., Tang, R., Zhou, F., Zafeiriou, S. P., Zhang, Y., Wang, J. & Li, Z. 2020. Autohash: Learning higher-order feature interactions for deep CTR prediction. IEEE Transactions on Knowledge and Data Engineering.

Zaremba, W., Sutskever, I. & Vinyals, O. 2014. Recurrent neural network regularization. arXiv preprint arXiv:14092329.

Zeng, J., Chen, Y., Zhu, H., Tian, F., Miao, K., Liu, Y. & Zheng, Q. 2020. User sequential behavior classification for click-through rate prediction. In *International Conference on Database Systems for Advanced Applications*, 267–280. Springer.

Zhang, K., Qian, H., Cui, Q., Liu, Q., Li, L., Zhou, J., Ma, J. & Chen, E. 2021a. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 984–992.

Zhang, W., Du, T. & Wang, J. 2016. Deep learning over multi-field categorical data. In *European Conference on Information Retrieval*, 45–57. Springer.

Zhang, W., Qin, J., Guo, W., Tang, R. & He, X. 2021b. Deep learning for click-through rate estimation. arXiv preprint arXiv:210410584.

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B. & Liu, T. Y. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhao, P., Luo, C., Zhou, C., Qiao, B., He, J., Zhang, L. & Lin, Q. 2021a. Rlnf: Reinforcement learning based noise filtering for click-through rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2268–2272.

Zhao, Z., Fang, Z., Li, Y., Peng, C., Bao, Y. & Yan, W. 2020. Dimension relation modeling for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2333–2336.

Zhao, Z., Yang, S., Liu, G., Feng, D. & Xu, K. 2021b. Fint: Field-aware interaction neural network for ctr prediction. arXiv preprint arXiv:210701999.

Zhao, Z. Q., Zheng, P., Xu, St. & Wu, X. 2019. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems* **30**(11), 3212–3232.

Zheng, Z., Zhang, C., Gao, X. & Chen, G. 2022. Hien: Hierarchical intention embedding network for click-through rate prediction. arXiv preprint arXiv:220600510.

Zhou, G., Bian, W, Wu, K., Ren, L., Pi, Q., Zhang, Y., Xiao, C., Sheng, X. R., Mou, N., Luo, X., et al. 2020. Can: Revisiting feature co-action for click-through rate prediction. arXiv preprint arXiv:201105625.

Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X. & Gai, K. 2019. Deep interest evolution network for click-through rate prediction. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 5941–5948.

Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H. & Gai, K. 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.

Zhou, J., Albatal, R. & Gurrin, C. 2016. Applying visual user interest profiles for recommendation and personalisation. In *International Conference on Multimedia Modeling*, 361–366. Springer.

Zhu, C., Chen, B., Zhang, W., Lai, J., Tang, R., He, X., Li, Z. & Yu, Y. 2021. Aim: Automatic interaction machine for click-through rate prediction. IEEE Transactions on Knowledge and Data Engineering.

Zhu, D. 2021. Advertising click-through rate prediction based on cnn-lstm neural network. In Computational Intelligence and Neuroscience *2021*.

Zhu, J., Liu, J., Li, W., Lai, J., He, X., Chen, L. & Zheng, Z. 2020. Ensembled CTR prediction via knowledge distillation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2941–2958.

Zhu, J., Shan, Y., Mao, J., Yu, D., Rahmanian, H. & Zhang, Y. 2017. Deep embedding forest: Forest-based serving with deep embedding features. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1703–1711.

Zou, J., Rui, T., Zhou, Y., Yang, C. & Zhang, S. 2018. Convolutional neural network simplification via feature map pruning. *Computers & Electrical Engineering* **70**, 950–958.