User Response Prediction in Online Advertising

ZHABIZ GHARIBSHAH*, Dept. of Computer & Elect. Eng. and Computer Science, Florida Atlantic University

XINGQUAN ZHU*, Dept. of Computer & Elect. Eng. and Computer Science, Florida Atlantic University

Abstract-Online advertising, as the vast market, has gained significant attentions in various platforms ranging from search engines, third-party websites, social media, and mobile apps. The prosperity of online campaigns is a challenge in online marketing and is usually evaluated by user response through different metrics, such as clicks on advertisement (ad) creatives, subscriptions to products, purchases of items, or explicit user feedback through online surveys. Recent years have witnessed a significant increase in the number of studies using computational approaches, including machine learning methods, for user response prediction. However, existing literature mainly focuses on algorithmic-driven designs to solve specific challenges, and no comprehensive review exists to answer many important questions. What are the parties involved in the online digital advertising eco-systems? What type of data are available for user response prediction? How to predict user response in a reliable and/or transparent way? In this survey, we provide a comprehensive review of user response prediction in online advertising and related recommender applications. Our essential goal is to provide a thorough understanding of online advertising platforms, stakeholders, data availability, and typical ways of user response prediction. We propose a taxonomy to categorize state-of-the-art user response prediction methods, primarily focus on the current progress of machine learning methods used in different online platforms. In addition, we also review applications of user response prediction, benchmark datasets, and open-source codes in the field.

1 INTRODUCTION

Online advertising [195], as a multi-billion dollars business, provides a common marketing experience when people are accessing to online services using electronic devices, such as desktop computers, tablets, smartphones *etc*. Using Internet as a means of advertising, different stakeholders act in the background to provide and deliver advertisements to users through numerous platforms, such as search engines, news sites, social networks, where dedicated spots of areas are used to display advertisement (ad) along with search results, posts, or page content.

Similar to the traditional media, such as printing magazines and newspapers where specific spaces are assigned to be sold for ads, a portion of online services and websites are filled with clickable components to display marketing messages. Under such circumstances, the ads to be displayed to audience (*i.e.*, users) are either pre-sold (*i.e.*, negotiated) by sellers (publishers) to buyers (advertisers) or they are dynamically selected through a real-time bidding (or auction) [151, 173]. In online advertising, advertisers are bidding an ad opportunity, but only the winner has the chance to serve their ads to users (so only the winner needs to pay to the publisher for the purchase of the auctioned ad opportunity). During the whole process, the effectiveness of the online advertising is typically evaluated through

Authors' addresses: Zhabiz Gharibshah, zgharibshah2017@fau.edu, Dept. of Computer & Elect. Eng. and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL, 33431; Xingquan Zhu, xzhu3@fau.edu, Dept. of Computer & Elect. Eng. and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL, 33431.

signals made by users towards the displayed ads. These signals are typically considered as user responses starting with a click on ads in web-pages or a tap on screen in mobile apps. Once displayed ads are clicked by users, the payment/revenue is generated between advertisers and publishers. As a result, for both advertisers and publishers, it is crucial to design a user response based pricing model.

Predicting a click, as the first measurable user response, is an important step for many digital advertising and recommendation systems to capture the user propensity to following up actions, such as purchasing a product or subscribing a service. Based on this observed feedback, these systems are tailored for user preferences to decide about the order that ads should be served to them. In the era of search engines and social websites, companies like Google introduce paid search advertising [128] via user intents recognized through the query keywords. In social media marketing, platforms like Facebook provide advertisers with user demographic information from user-generated content for viral marketing [24]. In conventional advertising in TV or printed newspapers, monitoring the effectiveness of ads is difficult. However, online advertising leverage performance metrics for targeting ad audience, so stakeholders can immediately obtain ad advertising feedback, through clicks, conversions, and other types of user response, to adjust their budget, price for bidding etc. [151].

The essential goal of different types of advertising systems, either traditional media based or modern online advertising based, is to find the best matching between audience (users) and ads, given contextual features in each platform. From computational perspective, this is equivalent to finding a way to accurately predict positive or negative user responses to an ad, given observed user data. It is shown that the accurate prediction of user response metrics can directly determine the revenue for both publishers and advertisers [1, 17]. The variation of the problem is defined by the availability of context in different platforms. The context in search engines are query generated by users. In display advertising, the context is considered as websites visited by users, and in-app advertising the context is the specific logical stage in mobile apps for marketing.

For years, industry and academia have developed numerous approaches to use holistic data to predict positive response of users where the positive response is typically defined in the form of the estimation of click-through rate on ads or user interactions for purchasing a product, *i.e.* a conversion. Such approaches vary from data hierarchy [1, 2, 68, 99], clustering [47, 119, 124, 158], collaborative filtering [82, 95], classification [13, 26, 53, 122, 164], to graph and network based analysis [146, 149].

As data are becoming rapidly available, machine learning based approaches have been used in nearly all domains to solve different types of challenges for knowledge discovery [194]. For online advertising, this is especially true. Since the very beginning, industry has been actively seeking effective and efficient computational methods to tackle the data volumes and real-time decision challenges. Many approaches, such as deep learning and factorization machine based methods, demonstrate a great potential to accurately estimate user responses [44, 57], but the data intensive nature and the real-time requirement have made the accurate user response prediction for online advertising extremely challenging. Here, we briefly summarize the major challenges as the following fourfold:

- **Scalability:** In real-world advertising eco-system, the number of visited web-pages is extremely large. Combining with factors like the number of unique visiting users and the amount of ads, it results in a giant dataset for analysing. In many studies [19] machine learning has been applied to predict user response and boost the personalizing of digital advertising. It is important to design solutions for large scale advertising data [17, 38, 110, 150].
- **Response rarity:** Statistics shows that the rate of click and conversion of all types of ads is not more than 2 percent over all displaying ads. Therefore, finding a way to overcome class imbalance issue and mitigate the adverse effects on prediction results is a challenge for the prediction algorithms.
- **Data sparsity:** This issue in online advertising and recommender systems stems from two factors. First, majority of input data consist of categorical features which need to use binary

representation, resulting in high dimensional vectors with very few non-zero values. Additionally, interactions between users and items follow the power law distribution, meaning majority users are interacting with a small number of items and products.

- **Cold start:** This is the common challenge for new new ads, products, and services, because no historical user information available is available to be used for estimation.

Indeed, many solutions have been proposed, but primarily focus on new methods for user response prediction. Several works propose to study current business model and technologies evolved from traditional media buying [18, 173], or review display advertising literature and new directions [23]. In [173], the authors go over the business model of real-time bidding by introducing keys actors in the market for ad delivery. From economic perspective, [23] outlines the eco-system of display ad market and non-guaranteed selling channels provided to buy and sell ads in real-time. It reviews the disciplines regarding the ad pricing decision made by different actors like advertisers and publishers and other intermediary nodes. The study [18] reviews the technologies provided for online and mobile advertising, including pricing models implemented between advertisers and publishers, inherent networking schemes by addressing the user privacy and malicious ad related activities.

Unfortunately, all existing works, including the literature review, do not provide a complete overview about types of user response and underlying technical solutions in online advertising. Answers to many key questions remain unclear for both industry and academia, especially for someone who just steps into the online advertising field. What are the main advertising platforms? What type of user response can be modeled/predicted using computational approaches? What are the features and the source of features useful for use response prediction? How to utilize features for use response prediction? What are the main types of technical solutions for user response prediction? Are there any benchmark and online resources (datasets/software) available for evaluation purposes?

In this paper, we provide a comprehensive literature review of the latest computational methods for user response prediction in online advertising, with a focus on machine learning based approaches. To the best of knowledge this is the first survey study which is focusing on computational approaches for user response prediction. Our review covers different types of user response prediction tasks ranging from click-through rate prediction to user post-click experience evaluation. Our survey includes the description of the online advertising eco-system, platforms, data sources, and early studies for user response prediction. We also consider the most recent work in this context which propose more advanced algorithmic designs and feature extraction methods.

2 ADVERTISING ECO-SYSTEM & USER RESPONSE

In this section, we introduce key components and important concepts of online advertising eco-system. For ease of understanding, we summarize key concepts and their descriptions in Table A.1.

2.1 Online Advertising Eco-System

Online digital adverting heavily relies on real-time bidding (auction) [173] for advertisers to make decisions to display ads in online portals. In this architecture, an ad exchange network connects sellers (publishers) and buyers (advertisers), so they can negotiate to respond to ad requests in real time. In order to participate in the ad bidding, publishers and advertisers connect to the ad exchange network through SSP (Supply-Side Platform)s and DSP (Demand-Side Platform)s, respectively, to cast auctions (for SSP) and manage bids (for DSP), therefore ads are eventually delivered to different media platforms, *e.g.* a third-party website, search engine result page, or the web-page of social networks.

In Figure 1, we illustrate an online advertising eco-system. The workflow starts with an event when a user, *i.e.* an audience, launches an URL request from a publisher's web page. The ad request for ad placements is sent to SSP to trigger an ad auction call (*i.e.* an opportunity). If the requested web-page contains available ad placement, the ad call will be submitted to Ad Exchange Network, leading to

negotiations with advertisers through DSP based on bidding mechanism. The winning advertiser will insert ad script in the user requested page, so the ad is eventually delivered to the user. In the case that the displayed advertisement matches to the user preference, user response in the form of click or further user engagements, like purchasing or subscription, is generated.

The revenue of advertisers and publishers, in the online advertising, is based on the user response such as clicks or conversions. Therefore, serving users with ads best matching to their preference is of interests to both advertisers and publishers. Under such circumstances, using context to find users' preference plays an essential role for user response prediction. The information from publisher websites is usually obtained from crawling the web-pages to summarize the context. It is then complimented by online analysis of cookie data and browsing history made by users. Such information allows system to identify user interest and response regarding ad impression.
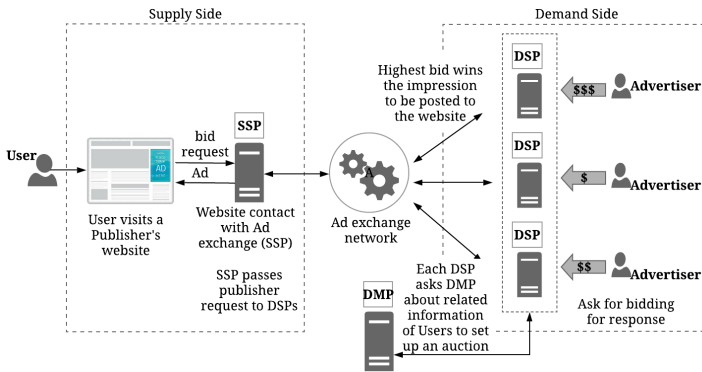

Fig. 1. Advertising Eco-system. From left to right, a process is triggered when users start to interact with online services through either visiting a web-page, searching an item, or checking the social media in publisher website. In the case that the web-page has web placement available, the publisher sends ad request through SSP node to the Ad Exchange. The bid request related to requested ads are forwarded from Ad Exchange to DSP nodes which represent advertisers. After getting relevant user information, such as user profile and their previous interaction, through DMPs, the auction is set up to gather bid among DSPs. The bidder with the highest bid wins and its ad script is forwarded to the publisher to be embedded in the page requested by users.

2.2 User Response Types

In web applications like web search engines, display advertising, recommendation systems, or e-commerce platforms, a user response to advertisements starts with a simple click on the ad or a touch on the screen in mobile app. This action is considered as an implicit positive user response which will direct users to a landing web-page. If ad content matches to user preferences, it encourages users to follow up promoted messages by generating the next clicks which can end up with desired activity such as a purchase. In online advertising the initial click or final purchase actions over displayed advertisements are considered as the critical measures to evaluate the performance of user response predictive models. Online advertising systems are generally integrated with recommendation systems in e-commerce platforms to provide users with ranked items based on explicit user's rating and implicit feedback. These feedback can be measured using different metrics to show the performance of the advertising systems. In the following subsections, we define prevalent metrics in this domain.

2.2.1 Click-Through Rate. Click-through rate (CTR) value is of the most important metrics to evaluate the quality of ads and the performance of campaign ads. Two elements to calculate the click-through rate values are clicks and impressions. The click-through rate is typically defined as the number of click events over impressions or the percentage of served advertisement ending up with user click events.

$$CTR = \frac{\#\ of\ Clicks}{\#\ of\ Impressions} \qquad (1)$$

The number of impressions are perceived as the number of times an ad or a promoted product is served to the users' device which is engaged to an active online platform, where the publisher can be the website of search engine, a social media, or a third-party website. A click event is an indicator of user engagement, which can be a mouse click on ad creatives on a desktop system or touching them on mobile devices. The definition of click event is extended in different applications like the number of downloads [79], or in the social media context as positive and negative actions like reply, commenting, sharing, dismiss, etc. [70]. Common issue that frequently exists regarding this metric is data class imbalance problem where the number of clicks compared to the number of impressions is very few. Some studies [66, 187] suggest that relying on this metric to evaluate the performance of e-commerce search results can be noisy and generate misleading outcomes.

2.2.2 Conversion Rate. In order to evaluate user experience and activities after the click, metrics are introduced to evaluate ad campaigns following cost-per-conversion business model. The desired actions for advertisers like purchases, subscription of service, registrations, and installation of a software, are considered examples of conversion events.

Conversion rate is simply defined as the proportion of users who visited ad creative in online portal and chose to take any above mentioned actions after opening the landing website.

$$CVR = \frac{\#\ of\ Conversions}{\#\ of\ Impressions} \qquad (2)$$

A conversion is generally considered as a user response following a temporal order of events starting with the page visit, ad display, click, to the conversion. In the case that the sequence of conversion, click, and visit of ads are all available, the prediction of conversion rate is defined as a post-click conversion prediction. The essential goal of the problem is to estimate the probability of a conversion event given clicks and the context [88, 160].

2.2.3 User Engagement. Recommender systems have been commonly used in different application platforms range from social networks and news feed services to e-commerce portal and entertainment data stream services. The common problem in these system is the overload of information that users are confronted with the high volume of items being overwhelming to browse. The priority of these systems is to attract more users by replying their requests with a relevant list of items matched with their preferences. So, the common objective is to recommend a small set of items which includes promoted ones to get immediate implicit user feedback (e.g. CTR) while keeps users activating. User engagement objectives have been studies differently in prior research. There are some studies which model active users by following churn-rate and dwell time analysis. Recent studies have modeled user engagement using multi-objective optimizations. So two recommending and online advertising are optimized together to satisfy user experience in the long-term [185, 186].

With the advent of smartphones and the increase in their popularity among users, there is a surge of interest in developing softwares operating on this platform. As a result, a new online advertising, called in-app advertising, emerges where specific spots on screen before completing a transition in the app are designed for commercial ads. In this context, some studies proposed to provide personalized ads [66, 163] which are evaluated by studying different users activity patterns to model users' engagement. Because smartphone platforms are personalized with respect to individual users, user response can be extended to the user engagement concept with the general questions to learn the factors which can (1) retain user being active to use an online service, like streaming providers and (2) also help gain revenue through directing people to take a desirable action with regard to ads. Therefore, several researches [85, 127] investigate features leading to user engagement with regard to mobile apps, and a recent work [8] proposes to study factors which are resulting in users being disengaged from mobile apps through hierarchical clustering models.

In video streaming platformss like YouTube, video ads have become a modern effective way of conveying commercial messages via telling a story to users. In this context, video completion rate value is a metric designed to evaluate the effectiveness of video advertisements and user engagements[1]. As it is shown in [63] the content, position and length of video ad along with the length and the provider of host videos in addition to user connection information (geography and connection devices) are key factors to impact video completion rates and evaluating the effectiveness of video ads.

In the context of e-commerce system, use engagement is generally evaluated by ranking metrics used in information retrieval systems. The performance of ranking in the produced ordered lists is established by considering a samples of users who have positive interaction with items. Generally, there is a chance that items preferred by users are missing in the list. Mean average precision at rank K ($MAP@K$), mean average recall at rank K ($MAR@K$) and Normalized Discounted Cumulative Gain at rank K, are the frequent metrics which give more details about the ranking performance. The $MAP@K$ assesses how much system can incorporate relevant items in the list. The second one $MAR@K$ checks how well model can create a list from all available items being relevant to user preferences. Due to the fact that the relevancy of items to user preferences are not the same, $NDCG@K$ consider the performance to put the more relevant items before the others in the recommended list S. It gives more significance to hit rates happened at higher ranks of the recommended list. According to Eq. (4), for each item in the recommended list, $rel_{s,r} = 1$ represents a case that the item s ranked at r is matched with the ground-truth otherwise it would be $rel_{s,r} = 0$. A log factor is used to assign a penalty with regard to position of items in the list.

$$MAP@K = \frac{\# \, of \, recommended \, items \, being \, relevant}{\# \, of \, recommended \, items} \qquad MAR@K = \frac{\# \, of \, recommended \, items \, being \, relevant}{\# \, of \, relevant \, items} \qquad (3)$$

$$NDCG@K = \frac{\sum_{s \in S} \sum_{r=1}^{K} \frac{rel_{s,r}}{\log_2(r+1)}}{\# \, of \, recommended \, items} \qquad (4)$$

2.2.4 *Explicit user feedback.* In contrast to implicit user feedback, explicit rating score information allows users to express their interests or opinions through online methods like surveys. Compared to implicit user feedback, this information are scarce since they require users to provide additional input with regard to items via surveys and online forms. In addition, they may come with bias in user's opinions. Implicit feedback are frequently analyzed through models for classification tasks where explicit user responses are adopted for regression tasks such as rating prediction so that user rating score with regard to new items are estimated by the system.

Table 1 summarizes characteristics and challenges of different user response types.

Table 1. The characteristics of different user response types.

Metric	Abundance	Accuracy	User feedback		Illustration of user preferences	Description
			Implicit	Explicit		
Click-Through Rate	High	Low	✓		Positive	Often not the final goal
Conversion Rate	Low	High	✓		Positive	Needs a domain specific definition
User Engagement	High	High	✓		Positive	Assumes a direct trend between retention & engagement
						Takes short-term or sequential user behavior and intents
User rating scores	Low	High		✓	Pos. and Neg.	Sparse data

3 FEATURES FOR USER RESPONSE PREDICTION

User response prediction plays an essential role for online advertising and recommender systems [68], where the prediction is typically defined as the probability of users making a positive response on promoted item in a marketplace, ad, or news article in online platforms [95, 135, 151]. The performance-based advertising is the paradigm mainly followed in online advertising systems, where the predicted

---

[1]It is defined as the percentage that videos are watched to the end. The more time the video ad is watched by users, the higher the chance it may influence users to take follow-up actions.

probability is not only used as an indicator to present user preferences, it is also involved in bidding strategies to determine the revenue of advertiser and publishers [125].

Figure 2 shows the workflow of typical user response prediction models consisting of two main stages. The first stage is related to data collected from different data sources (Users, Advertisers and Publishers) in online advertising systems. After the pre-processing and labeling steps, data samples are described with series of features (fields) along with label (class) values which are normally specified as binary user response value such as 1 for click, conversion, purchasing, etc. and 0 otherwise. For recommendation systems, the output in Figure 2 is an ordered list of promoted/recommended products. For the prediction task, it will output probability of users making an interaction (*e.g.* a click) on items in the list. Like typical machine learning problems, the input data should be described through feature vectors to capture the class correlation, meaning that features need to be discriminative for the prediction task. Therefore, during the second (learning) phase, features are extracted using different approaches, such as (1) using data fields to represent users, pages, etc. and create sparse features; or (2) using embedding based approaches to create dense features.
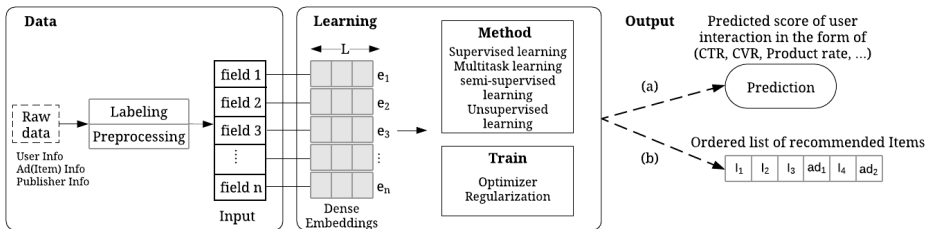

Fig. 2. The schema of user response prediction workflow. Embedding layer is the common paradigm to deal with high dimensional binary representation in user response prediction. They can either be set by pre-defined values or be trained as internal parameters in end-to-end models like deep learning methods. The output can be considered as two types of user responses a) a scalar value of predicted score for an interaction between given user u_i and item I_j b) a ranked list of regular and promoted items ordered by predicted user response scores.

3.1 Type of Features

In order to accurately predict user response, it is important to train models using discriminative features. In the following subsection, we will discuss features studied by various methods.

3.1.1 Multi-field categorical features. The typical input data fed into online advertising systems are generally formed as multi-field categorical values. Contrary to continuous features which are generally found when dealing with images or audios, the input data contains an array of categorical fields including Gender, City, Age, Id, $\cdots$ and device type and ad category, $\cdots$, to describe users and ads or the other related objects in the system. An event representing the user interaction with online advertising includes features from different actors like users, publishers, advertisers and the context in online advertising systems. A representative list of categorical features corresponding to user profile and behavior, advertisement and publisher's web-page is provided in Table 2. The one-hot encoding is the conventional approach to deal with this type of data [48]. As shown in Figure 3, each field is shown as a binary vector. The dimension of vector is determined by the number of unique values which are taken in the field in which one entry is set as one while the remaining as zero. In this example, fields like gender has the length of 2 and the length of weekday is 7. The simple way to represent features is the concatenation of these vectors which typically creates a high dimensional sparse binary vector. In the mathematical way, considering the input data with n feature fields and x_i is the hot-encoded vector of the field feature i with dimension of K_i where $\sum_{j=1}^{n} |x_i| = k$. In the case $k = 1$, we have one-hot-encoded vectors while $k > 1$ refers to multi-hot-encoding [39, 97, 190] that feature field is represented by more than one value entries. To handle the high dimensionality issue,

the common approach for many classification based methods is employing the embedding step to generate condensed embedding vectors. These vectors can be concatenated like $x = [x_1, ..., x_i, ..., x_n]$ to create input layer of different user response predicting models.

field 1=Mobile	field 2=Tuesday	field 3=Male	field 4=Berlin
user_device_type	weekday	gender	city
[1,0,0,0]	[0,1,0,0,0,0]	[1,0]	[0,0,1,0,...,0]

Fig. 3. The characteristics of multi-field categorical features as the input to user response prediction models. The binary representation of multi-categorical features is created using one-hot-encoding

Object	Features
User	Id, location(Area, city, country), IP, Network Spec, Browser cookie, Gender, Age , Date
Advertiser	Ad id, Ad group Id, Campaign Id, Ad Category, Bid, Ad Size, Creative, Creative Type, Advertiser Network
Publisher	Publisher(Id), Site, Section, Ad Placement, Content Category, Publisher Network, Device type, Page Referrer
Context	Serve time, Response Time

Table 2. Representative categorical features corresponding to the main online advertising objects

3.1.2 Textual features. In search advertising, ads are displayed in the search result pages, incorporating textual data such as headline, relevant keywords and the body to highlight the details of promoted products. Many research proposes to treat click-through rate prediction task as the similarity learning between users' query keywords and keywords of ads using their proposed text based similarity. For example, keywords in the title and body of advertisements [6, 33] and keywords in user queries are considered as two sources of data to extract textual features in many designed models [32, 41, 147]. Relying only on ad textual content and user query at character- and word-level, a deep CTR prediction model [32] collects data from textual letters of query along with the title, the description, and the ad URL. They are organized to feed into system as a one-hot-encoded matrix.

3.1.3 Visual features. E-commerce platforms, which are available through web portals and mobile apps, are hosts of many categorical ads and items. Each item is generally described by texts and images, acting as visual features to attract users' attention. Categorical features are generally used for model user behavior history. However, the data sparsity issue in categorical data encourages to consider intrinsic visual information in images for the development of further methods [19, 38]. There is also an increasing interest to develop video ads for digital streaming platforms in which user responses generally happen by clicking on the image section [134]. Very recently, user facial information along with user behavior history is proposed to use for modeling user purchase interest. Analyzing this type data can provide an estimation of some user profile information such as gender, age and ethnicity which in turn draw inferences about user background and status and their manner for purchasing[86].

3.2 Organization of Features

Earlier studies to analyze user responses in online advertising mainly use one type of multi-field, visual or textual features for model designing, mainly because of their transparency and easy to interpret. Advanced models are later studied to extract complex features for better prediction accuracy. In the following section, we will go over a couple of models which take advantage of two important layout of features such as sequential and hybrid features to improve the performance of predictive models.

3.2.1 Temporal and sequential (user behavior) features. Users activities are commonly recorded as data logs available in many online data provider services. Considering the sequence of user actions *w.r.t.* different types of ads are valuable features for analyzing user response prediction [115, 170, 189]. The majority of proposed methods are categorized into recurrent neural network based and network based models (detailed in Sections 4.3 and 4.4.2). Some studies in literature showed that the history of previous visited pages, clicked ads [163], not-clicked ads [100] sorted by time in system can be leveraged to

model sequential dependency between input features. Several studies have shown the importance of these features to enhance the performance of various user response prediction tasks [36, 40, 189, 190].

In [190] user behavior features are represented as a list of visiting events of ads, each of which are described by categorical features about goods, shop, and page categories of past user-defined time points. Each time point is described using multi-hot-encoding.

Sessions are also used to represent user's behavior history [36]. Separated by occurring time, user activities are considered homogeneous in very short time slots within sessions but different with regard to other sessions. User interaction patterns over time are evolving in short-term and long-term trends. A common scenario is to deal with virtually short-term sequences of user behaviors when user profile (*e.g.* ID) of active users is not accessible via log-in to system. So, the task of predicting user responses is defined to extract relevant patterns based on limited actions of anonymous users. In this case, the complete user interaction histories are also organized into sessions [115, 183]. The long-term user interaction can also be studied to create user profile behavior. It can not only provide indication of user intent change over time which can be used to improve the predication user responses, the popularity pattern regarding products can also be identified to remind users about a product according their previous interactions. In [40] authors consider a sequence of user activity events before and after the ad click and corresponding passed time-slot to investigate the potential user conversion intents. They analyzed the effect of elapsed time as a feature for conversion rate prediction and using targeting and retargeting[2] paradigm for different users in online advertising systems.

3.2.2 Hybrid Features. Combining different types of features are also studied to enhance user response prediction. Some studies use textual features along with multi-field categorical features to improve the performance of recommendation systems in e-commerce platforms [97] and the sponsored search marketing [25]. Some research consider the compounds of categorical features and image data [19, 38] and the combination of categorical features with video data [134] to improve predictions. The combination of different features in modeling lead to various compound embedding layer for input data to generate a condensed feature representation, with pooling being employed to reduce parameters and cope with over-fitting. Max and sum pooling are also studied as the aggregation mechanism in some studies [25, 38]. The concatenation of feature embedding vectors is a straightforward approach commonly used in many studies [78, 117, 130, 179]. Recently, an adaptive approach to combine most relevant features from different feature types is employed based on attentive mechanism [38, 190].

4 USER RESPONSE PREDICTION FRAMEWORKS

For years, user response prediction, in online advertising, has been continuously evolving. Early approaches usually reply on hand crafted features to dissect data into different segments, where each segment contains users with similar response. Therefore, the click-through rate values or conversion rate values estimated on each segment can be used to estimate future (new) users' CTR values. Following similar approaches, clustering or collaborative filter based approaches are also proposed to recommend ads to users. In the context of recommendation systems, the ordered list of items including promoted products are proposed to users by predicting how likely the list contains items matching to user preferences. The evaluation of these systems are examined with different ranking and regression metrics (Detailed in Section 2.2). Typical types of recommender systems have analyzed past user interactions to detect a connection between users and products either through studying users with same tastes or similar items visited by different users. Recently, machine learning, especially deep learning, based approaches, are becoming increasingly popular for user response prediction, mainly because these approaches can simultaneously accommodate a large number of features, and learn to create new features, for accurate user response prediction.

---

[2]A cookie based advertising which tracks users clicking or visiting ad in a website who have not taken further actions against promoted products. Using this paradigm advertising systems remind users their previous interest about a promoted product.

Category			Publication
Data Hierarchy Based			[1, 2, 15, 62, 68, 99]
Collaborative-Filtering Based	Hierarchy Based		[82, 95]
	Network Based		[50, 61, 146, 149, 156]
	Hybrid Methods		[20, 51, 161]
Supervised Learning	Predictive Models	Logistic Regression	[5, 17, 27, 48, 65, 68, 121, 145, 180, 193]
		Factorization Machines	[21, 53, 57, 84, 99, 99, 106, 107, 112, 118, 122, 177, 179]
		Deep Learning Methods	[12, 15, 15, 19, 25, 32, 36, 38, 41, 51, 78, 83, 97, 100, 110, 130, 137, 163, 166, 189–191]
		Hybrid Methods	[22, 44, 117]
	Ensemble Methods	Cascading	[179]
		Stacking	[5]
		Boosting	[87]
		Mixed	[159]
Un-supervised & Semi-supervised Learning	Network Based	Network Embedding: *(Node Embedding, GNN Based methods, User Intention Network Modeling)*	[71, 74, 115, 150, 169, 170]
		Knowledge Network Based: *(Node Embedding, Meta-path Based methods, GNN Based methods)*	[35, 50, 116, 147, 148, 156, 157, 174]
	Clustering Based		[47, 119]
Stream-Based Data			[9, 29, 64, 67, 70, 126, 192]

Table 3. A taxonomy of user response prediction in online advertising, along with representative publications

In Table 3, we propose a taxonomy of user response prediction, which includes hierarchical based methods, collaborative filtering based approaches, supervised, semi-supervised and unsupervised learning studies. In the case that labeled data are available, supervised learning algorithms leverage the label in the definition of the loss function in their learning procedure. Unsupervised learning rely on unlabeled data for the loss function optimization. Semi-supervised models are between supervised and unsupervised models where their objective functions are optimized considering both data with and without labels. The supervised methods can be further categorized into basic predictive models and ensembles ones whereas semi-supervised and unsupervised category consists of network-based and clustering based methods. The last category in the taxonomy includes stream-based methods. Following subsections will discuss and review representative methods in each category.

4.1 Data Hierarchy Based Approaches

Using unstructured input features, data sparsity and cold start are common issues in online advertising and recommender systems. Data hierarchy based methods refer to approaches that organize data in a hierarchical format [81]. The motivation is to build a tree structured hierarchy, using some selected features, such that each leaf nodes represents a user groups sharing similar response. This hierarchy provides valuable information to show correlation between user responses at different level of granularity, which alleviates the adverse effect of limited historical information about users.

4.1.1 Data Hierarchy. As the first attempt to cope with data sparsity and limited historical data in online advertising, hierarchical structures of publisher web pages, ads and end-users are commonly used to address correlation between input features [1, 2, 68, 81, 99]. In this case, users, web-pages and ads are grouped based on different factors, such as demographic or geographic information about users, domain and content of web pages and the context and campaign of ads. An example of the data hierarchy is shown in Figure 4. From advertiser perspective, the hierarchy can be created by classifying ads based on campaign, content type, and advertisers. For publishers, web-pages can be grouped using simply URL path or the content category. Users can also be organized as hierarchical data using third party information like user geographic, ad and web-page visit history *etc.* Studies show that data hierarchy for ads, pages, and users provides useful knowledge to handle data rarity in click-through prediction [68]. Partitioning input space using tree structure represents similarity

between connected nodes with respect to user responses in local areas [2, 99]. In industry, these data hierarchies are created and maintained by domain experts.

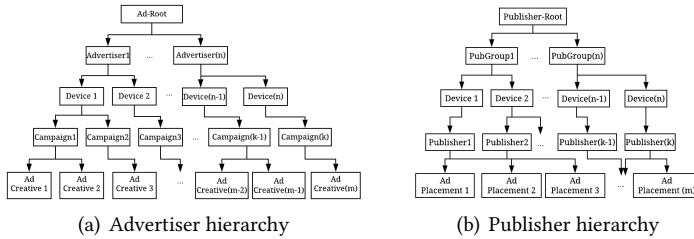(a) Advertiser hierarchy　　　　　　　(b) Publisher hierarchy

Fig. 4. The sample taxonomy representing a hierarchical structure for Advertiser and Publisher data where a) demonstrates grouping ad creatives through multi-level joint points when they are of same campaigns and are designed for the same devices by an individual advertiser b) indicates the publisher hierarchy from ad placement in web-pages, the running devices and grouping of publishers

4.1.2 *Representative Hierarchy Based CTR Prediction Frameworks.* Input features of online advertising systems consist of various sparse categorical features, which contribute to generate rare user responses such as clicks or conversions. To address these issues, many methods propose to create a hierarchical structure from input features to estimate user response from previous similar available samples [2, 68].

Problem Definition. For ads being served to users multiple times, the baseline problem to predict user response is defined as: given a pair of web-page j and ad k, the probability of response, like a mouse click, is calculated through the probability formula $P_{jk} = Pr(Click|Impression; j, k)$. This probability, *a.k.a* Click-through rate value (CTR), can be computed via binomial maximum likelihood estimation (MLE) where V_{jk} indicates the number of times ad k is displayed on the web page j, and C_{jk} is the indicator of click number, respectively [125, 158]. For the case $V_{jk} = 0$ or equals to a small value, the value of MLE estimate for CTR values becomes unreliable. Therefore, in literature, different methods are proposed to exploit hierarchical information for smoothing out the MLE predictions.

For instance, authors in [2] proposed to utilize two hierarchical structures between input features related to web-pages and ads to improve the prediction of user click responses in online advertising. In this study, they tackled a form of sparsity issue in the input data with a few number of available clicks and impressions. In order to reduce the variance made by the sparse clicks and/or impressions, a sampling approach is used to alleviate the rarity issue via negative sampling of majority class, *i.e.* web-pages without a click response. To control the effect of the bias made by sampling, a two-step method is used to predict the click-through rate. In the first step, a maximum entropy model is optimized based on an iterative proportional fitting method to estimate the actual number of impressions at all defined levels in the hierarchical structure. A tree-shaped Markov model is then used to predict the click-through rate value in the whole levels of the hierarchy using correlations between sibling nodes.

Further, a log-linear model (LMMH) [1] is introduced to improve user response prediction by exploiting correlations between sibling nodes at different data hierarchy levels. To increase the scalability of model to large dataset, a spike and slab variable selection method is proposed to control number of parameters in the regression model. This method deals with rare response rates by pooling data along a directed acyclic graph (DAG) obtained through a cross-product of multiple hierarchies.

Another study [62] advances the LMMH method using higher order feature interactions by fitting local LMMH models to relatively homogeneous subsets of the data. Given a relatively homogeneous partitioning of the feature space, several local LMMH models are fitted to data subsets on different nodes of a decision tree. To address over-fitting issue in the model, models are coupled with a temporal smoothing procedure designed based on a fast Kalman filter style algorithm.

Last but not least, a study [68] investigates the data hierarchy for three objects of users, advertisers and publishers to deal with the data sparsity and class imbalance problem for conversion prediction. Taking conversion event as Bernoulli random variable with two possible values of conversion and no conversion, a binomial distribution is used to model the conversion given a triple of user u_i and ad a_k and web-page p_j. To address the data sparsity, they propose to capture the correlation in the conversion output using clustering of similar users with regard to conversion rate values, grouping advertisements from the same campaigns and web-pages with same category types. The conversion estimation is calculated at different levels of the hierarchy made from the cross product of levels in three hierarchical structures of users, publishers and advertisers via the maximum likelihood estimation as follows.

$$P(Y = 1 | u \in C_{u_i}, p \in C_{p_j}, a \in C_{a_k}) = \begin{cases} \dfrac{C_{ijk}}{I_{ijk}} & \text{if } I_{ijk} > 0 \\ unknown & \text{otherwise} \end{cases} \tag{5}$$

where C_{u_i} is the cluster that u_i belongs to. C_{p_j} and C_{a_k} indicate the cluster of web-page p_j and ad a_k, respectively. The final estimation of the conversion rate value is then modelled using logistic regression from the linear combination of MLE estimators at different hierarchical levels.

4.2 Collaborative Filtering Based Approaches

Collaborative filtering is an effective approach to predict online user interests. The general idea is to analyze previous behaviors of users to predict possible future user interests or to generate suggestions that may match to the preferences of the new similar users.

Problem Definition. For collaborative filtering methods, the input is an incomplete sparse matrix $X \in R^{m \times n}$ of user-item preferences, which suffers from the data sparsity, *i.e.* some X_{ij} entries are missing. The goal is to fill in missing entries with predicted scores. The state of the art in collaborative filtering is matrix factorization [52, 95] which is based on an idea that the matrix of users' preferences *w.r.t* items X can be factorized into two low-rank matrices of Users α and Items β. It is modelled as $X \simeq \alpha^T \beta$, where $\alpha \in R^{k \times m}$ and $\beta \in R^{k \times n}$ and k is the dimension of latent features. Conceptually, each α_i represents a user, and each β_j represents an item. The simplest factorization model is to solve the following optimization where latent feature vector of users and items are controlled with user-defined regularization function σ in different studies to prevent the model from the over-fitting issue [61, 95].

$$\min_{\alpha, \beta} \frac{1}{|O|} \sum_{(i,j) \in O} (X_{ij} - \alpha_i^T \beta_j)^2 + \sigma(\alpha, \beta) \tag{6}$$

In general, collaborative filtering is based on the past interactions between users and products. It can be seen as the implicit feedback like click or conversion event on products or explicit feedback like product ratings. Interactions between web-pages and ad banners can be shown as a matrix of web-page-by-ad feedback score (click-through or conversion rate rate or product rate values). The correlation between web-pages and ads is captured to calculate predicted scores for missing entries which can be intuitively related to the user response prediction task. The major studies in this category take advantage of collaborative filtering methods along with side information such as the user and item neighborhood models [61], the data hierarchies [82, 95] and knowledge graph data [146, 149]. Hybrid models are used to tackle scenarios with data sparsity and cold start problems to improve the prediction performance. It is conventional that initial models resorted to apply matrix factorization and inner-product operator on latent factor vectors to establish connection between users and items. Recently, neural architecture [51, 161] and attention mechanism [20] are proposed as the alternative to learn higher order interactions on data. The user responses analyzed to evaluate the performance of models in the studied papers range from explicit product rate scores [61, 161] to implicit user feedback like CVR scores in [95, 146, 149] and recommended ordered list [20, 51].

One of initial works in the collaborative filtering domains, developed based on latent factor models like singular value decomposition, is known as SVD++[61]. For a personalized recommendation system task, the authors improve the accuracy of system by addressing both explicit and implicit user feedback in a hybrid model. To do so, additional terms are added to optimize the loss function (6) which is organized in three levels. In the first level, bias terms, in form of the addition of average of rating value of all items, the bias in average rating made by user α_i, and the corresponding bias for item β_i to control the discrepancy between actual values and predicted values in the loss function, are added. In the second level, a loss function is defined by adding a term to include implicit available feedback. They refer to all items that user had an interaction before. The implicit feedback here are considered from a series of browsing, purchasing and search user history in e-commerce systems. In the last level, a neighborhood model which addresses the effect of bias in average rating value made by neighbor users and items is added. Combining these terms together, the parameters of proposed model are updated using gradient descent optimization. It led to an improvement calculated for product rating prediction and providing top-k personalized recommendation tasks.

In an extended matrix factorization design [95], hierarchical information of web-page and ads are integrated as additional side information into latent features of collaborative filtering to tackle data sparsity and cold start problem. Explicit features from ads and web-pages in side information are linearly augmented to implicit features using a log-linear latent features model and user-interest propagation framework (FIP) to enrich input features. As a hybrid method, it includes hierarchical structural information into their factorization model using three learning ideas such as hierarchical regularization, agglomerate fitting and residual fitting.

In online advertising, interactions are generally made between multiple entities including users, items, and ads. Tensor factorization, as an extended version of matrix factorization, can use the similarity between different types to predict potential interaction between pair of instances. To address the compound similarity of entities with regard to a possible interaction, a Hierarchical Interaction Representation model [82] is proposed to provide a joint representation to model mutual actions between different entities. Three dimensional tensor multiplication is used for modeling characteristics of pair of entities.

Recently, some studies [146, 149] propose to organize user and ad as heterogeneous information graph to improve collaborative filtering. Authors in [146] suggests an end-to-end learning method to incorporate side information from knowledge graph (KG) into an item-based collaborative filtering approach for click-through rate prediction. They propose an extended knowledge graph embedding method which starts building an initial user preference sets in the knowledge graph that are originally set up from previous user click activities. An iterative propagation of user preferences along edges over the knowledge graph is used to create k Ripple sets to model potential user liking versus items. Learning an embedding vector for each ripple set, the embedding vector of user response versus items is calculated from the sum of corresponding embedding vector of ripple sets. The click-through rate score of user u versus item v is modeled using dot product embedding vectors of u and v each of which are trained based on a Bayesian framework and gradient descent learning.

4.3 Supervised Learning Based Approaches

In this section, we review supervised learning based methods which formulate the prediction of user response rates as binary or multi-class classification task in online advertising platforms. These methods can be categorized into two categories including the basic and ensemble predictive methods. Following the structure in Figure 2, input features are generally considered as multiple feature fields gathered from different sources like user, advertiser and publisher. The input layer in classification methods are considered as a numeric vector from concatenation of all fields.

$$x = [x_1||x_2||\cdots||x_n] \tag{7}$$

where n is the number of features and x_i is the representation of field i. For categorical data, feature value is encoded into a numeric vector through directly one-hot-encoding. Fields with continuous values are first discretized to be encoded to binary vectors by one-hot encoding.

Logistic Regression based methods. Logistic regression is one of the first attempts to train models to predict user response from input categorical features. As it is shown in Figure 6(a), this method uses linear combination of coefficient values and input sparse binary feature vector to predict the binary output value. Given the input dataset with m instances of (x_i, y_i) where $x_i \in \{0,1\}^n$ is an n-dimensional feature vector and y_i is the label to represent the user response as (click:1, no-click:0). The predicted probability of x_i belonging to class 1 is modeled by Sigmoid function as:

$$Pr(y = 1 | x_i, w) = \frac{1}{1 + \exp(-w^T x_i)} \tag{8}$$

The model coefficient $w \in \mathbb{R}^d$ is achieved by minimizing the negative log likelihood as follows:

$$\min_w \frac{\lambda}{2} \|w\|^2 + \sum_{j=1}^m \log(1 + \exp(-y_i \phi_{LR}(w, x_i))) \tag{9}$$

where $\phi_{LR}(w, x) = w_0 + w^T x = w_0 + \sum_{j=1}^n w_i x_i$ is the linear combination of coefficients along with bias value w_0 and features that $w_0 \in \mathbb{R}$ and $w \in \mathbb{R}^n$. As it is shown in literature Eq. (9) is convex and differentiable, so gradient based optimization techniques can be applied.

Challenges and extended methods. Some studies [17] indicate that the implementation of logistic regression methods is possible with high scalability through Maximum Entropy approach and a generalized mutual information and feature hashing as the regularization. However, modeling linear interaction between feature values only address the effect of features with class label separately. Therefore, it cannot always generate an acceptable performance in user response prediction task which gets impacted by some issues such as class imbalance originated from low click and conversion rates, the cold start issue for new instances, long cycle of user purchase responses, and non-linear interaction between input features. Authors in [27] use historical information of brand website visit as the proxy to model predictor using logistic regression model. A study [68] suggests to create hierarchy structure from previous user performances that is captured from grouping ad campaigns and publisher pages and users. A logistic regression model is used for linear combination of local MLE estimators.

Employing the side information using transfer learning has also been studied in some work [26, 165]. In [26] a transfer learning method is developed to combine data from a model on small set of conversion data to improve post-view conversion rate for large number of ad campaigns where click event is not necessarily required. In [165] a transfer learning approach was developed to design a natural learning processing method to capture transferable information of related campaigns. It is motivated by the fact that the similar searched content and visited web-pages by users can be indicators of their future purchase interest. In another work, a practical result from applying logistic regression for big data in social media platform demonstrated that the weakness of linear modeling could be reduced by cascading with decision tree models to implement non-linearity of input categorical data [48].

Factorization Based Methods. In order to consider non-linear interaction between features values, factorization machines (FM) combine support vector machine method with factorization models [122]. This allows the method to carry out parameter estimation under the data sparsity using linear complexity. This can be done by modeling the feature value interactions through a product of two latent vectors $v_i, v_j \in \mathbb{R}^k$. The dimensionality of latent vectors is the hyper-parameter that defines the number of latent factors.

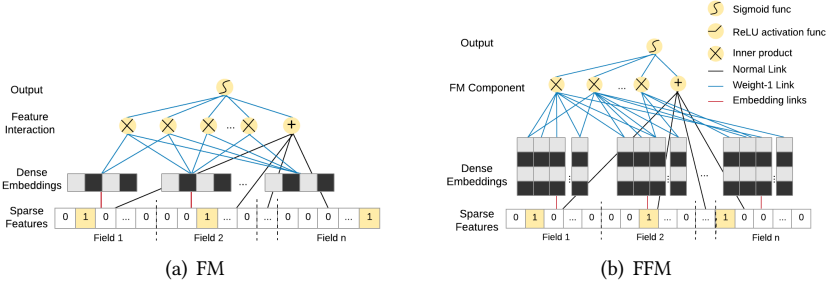$$\phi_{FM} = w_0 + w^T x + \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j \tag{10}$$

Fig. 5. a) FM Model: The architecture of Factorization Machines method as the extension of logistic regression by using dot-product ⊗ operators fed by dense embedding vectors of input sparse features. b) Field-aware factorization machines (FFM) model: The structure is similar to FM model. The difference is that the sparse interaction between each feature value in the current field i with another one in the other field (ex. $i + 1$) is modeled by a separated embedding vector. Each feature field i is represented by an embedding matrix.

Figure 5(a) shows the architecture of factorization machines as the combination of two terms including the feature interaction $\langle v_i, v_j \rangle$ and linear information $w_0 + w^T x$ to model click responses. The idea is that embedding vectors of features can be trained well to preserve feature interaction through dot product operation if there are enough occurrences that the features appear in the dataset.

Extended Factorization Machines models. Since factorization machines have a closed form equation that can be calculated in linear time, it is shown that the parameters of models can be trained using gradient based methods like stochastic gradient descend optimization (SGD) [122]. Some studies showed that FTRL-Proximal algorithm with L_1 regularization and per-coordinate learning rate, which was successfully used for logistic regression based models [92], can also outperform SGD algorithm for extended factorization machine models [139]. However, this method suffers from some limitations.

One of the downsides of factorization machines modeling is that for multi-field categorical data, feature values may come from different field feature that may change the interaction between feature values. But most methods deal with feature values uniformly. Therefore, Field-aware Factorization Machine (FFM) [57] is proposed to discriminate the interaction between various feature values of different fields. To this end, it suggests to add one dimension to model parameters to allocate more than one embedding vector to features since pair of features incorporate different feature types information. This changes the modelling of feature interaction as the following equation:

$$\phi_{FFM} = w_0 + w^T x + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_{i,F(j)}, v_{j,F(i)} \rangle x_i x_j \tag{11}$$

where $F(i)$ is an indicator of field name that feature corresponding to the first entry of feature interaction while $F(j)$ is an indicator of field name that feature is related to the second entry of interaction. Lack of consideration into the importance of features and the limitation of inner-product to model feature interaction are two issues in baseline methods of factorization methods have been studied in many other works as follows.

The baseline factorization machine methods usually consider all combinations of feature values in different fields with the same weight. But interactions between features often vary and do not have equal values. So there is a chance that using less important features in the training set, the noise is actually learnt by the model which can have the adverse effect on the performance. This motivates studies to impose weights on the interactions [53, 106, 162] To this end, authors in [106] proposed a weighted version of field-aware factorization machine (FwFM) that can use the memory efficiently for model parameters. It adds more information through a weight matrix to consider the difference in the strength of interaction between feature values originating from different pair of fields.

In [162] follows a deep learning study in which the importance of feature interaction were studied using an attention network through a layer to learn corresponding weights. A study [99] centered the work on a different aspect to use a cost sensitive approach to address the cold-start issue and using the data hierarchy for the data sparsity. They designed an importance-aware loss function to assign the more importance weights and penalty values for ad samples which are shown more to users but their user response predicted wrongly. Authors in [112] also proposed a robust factorization machines method considering user response prediction as a classification problem under the noise. The uncertainty within input samples is modeled by an optimization through an uncertainty vector with each dimension corresponding to independent noise value.

Aside from the above mentioned points, the capability of factorization machines to address data sparsity issue using inner-product operation can be limited when confronting high dimensional data. Modeling only 2nd-order feature interactions is not expressive enough for implicit higher order feature conjunctions. This stimulated motivations to propose high order variants of factorization machines method [13, 53, 164]. In [53] authors extended the feature interaction modeling in factorization machines using a bilinear interaction method which combines inner product and Hadamard product together to generate a fine-grained feature interactions. Very recently, [164] proposed a score function to replace inner-product operation between embedding vectors of input features. They discussed that using Lorentz distance, the triangle inequality principle between two points with regard to the origin point is not always consistent. They suggested to use the sign of triangle inequality to learn feature interactions through a proposed Lorentz embedding layer. To this end, a novel triangle pooling layer is proposed to substitute for the typical factorization machines structure.

Hybrid Approaches. Hybrid methods follow a classification technique that involves a number of heterogeneous methods each of which acts complementary to each other. Each method solves a different task and the classification decision is reached by the one method at the end. The distinction between ensemble methods and hybrid methods is that the former models are trained separately to generate the predictions at the inference time. On the contrary, hybrid models follow a joint training which optimizes all parameters simultaneously. This idea makes a motivation for attempts at developing more complex machine learning methods which are able to model a non-linear user interaction in online advertising systems. Regarding the user response prediction, the primitive predictors based on the logistic regression or factorization machines have weakness to capture limited range of feature interaction by addressing linear relations or dot product interactions between input features. Their performance is suffered from the data sparsity, class imbalance problem and cold-start problems. To address these issues,hybrid architecture of classifiers has been proposed in many studies. The categorization of hybrid models are presented as follows:

Logistic regression based methods. One of the first studies to improve logistic regression performance was the addition of decision trees to the structure of model [48]. In order to address the data sparsity in input data consisting of multi-field categorical data, they use a cascade of decision trees structured by boosting ensemble paradigm to provide a non-linear transformation of categorical features. Following a gradient boosting machine, the boosted decision trees generate a feature vector with the user-defined dimension k that is passed to logistic regression classifier for prediction.

The success of deep learning methods in capturing higher order interactions motivates research to include deep neural networks to improve the Logistic Regression in different studies [22, 130].

Although the logistic regression models have shown a good scalability and interpretability to handle the massive data in the online advertising industry [17], the generalization of model for predicting new samples is limited and highly dependant to whether high quality features can be obtained through feature engineering. Using the polynomial regression applied, the logistic regression model can only capture low-order feature interactions. This drives authors in [22] to approach a hybrid

structure of logistic regression and deep neural networks which are trained jointly to consider low and high order feature interactions when there is the data sparsity issue and dealing with massive data. As shown in Figure 6(b), the framework includes two components *i.e.* wide and deep. Wide linear component is modeled by the logistic regression classifier. It analyzes two sets of input features including raw categorical features and transformed features which are designed to memorize sparse feature interactions using a cross-product feature transformation. Following the feature engineering approach on the training data, the transformation function is designed to represent the frequent co-occurrence of features to explore the possible correlation with user responses. The deep neural network component is trained to generalize the prediction for unseen inputs through low-dimensional embeddings. In this model, the final output is calculated from the combination of wide and deep components using the logistic loss function. In initial studies, the embedding vectors are generated from a embedding dictionary by feature hashing method [17] where most frequent categorical values are transformed by projecting into pre-defined fixed-size numerical vectors [22, 25, 130]. The other models covered in the next section fix this issue using trainable embedding vectors.

Some recent studies in this category extend different elements in the hybrid design like embedding vectors [49, 97, 191] and neural network architectures [130, 189, 190] to improve prediction performance. Although it is typically expected that stacking of multi-layer fully connected neural networks can capture arbitrary non-linear relations between input features, dealing with a lot of parameters can cause different issues such as the degradation and over-fitting. A study [130] proposes to use a residual neural network for a deep component, where five hidden layers of residual units combined with original input features are added to the result of two layers of ReLU transformations. The effect of aggregation of embedding vectors on the prediction performance is also studied in [49]. The baseline methods [22, 130] follow a simple concatenation of embedding vectors in Figure 6(b) to be fed in a deep component to capture feature conjunctions. They demonstrate that it may carry less non-linear information in the low-level. Therefore, they suggest a Bi-Interaction pooling encoder to capture more informative feature interactions. Considering an embedding vector for each feature value, the Bi-Interaction pooling operation is designed to generate the aggregated vector as follows:

$$f_{BI}(V_x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i \mathbf{v}_i \odot x_j \mathbf{v}_j \tag{12}$$

where $V_x = \{x_1\mathbf{v_1}, ..., x_n\mathbf{v_n}\}$ is the set of embedding vectors, x_i is binary feature value in sparse input vector, $\mathbf{v_i}$ is embedding vector and $\odot$ operator makes element-wise product of two vectors.

Further, authors in [97] pinpointed that the semantic intrinsic relations between embedding vectors of user and ads can be captured through their proposed structured semantic models. They propose a series of orthogonal convolution and pooling operators rather than trainable convolutional operators which can be applied as embedding vectors to address semantic relations in input features. Experiments reported in the above studies show that applying hybrid methods can improve logistic regression, which highly depends on the quality of features prepared by using feature engineering. This encourages further studies to develop extensions of factorization machines with better generalization ability.

Factorization based hybrid methods. In [44], the authors provided a successful version of hybrid methods as the stack of factorization machines and fully-connected neural networks. The success of this design later led to employ this structure as a base for developing many extensions [152, 176]. The study pinpointed that Wide & Deep method [22] has some challenges in modeling of feature interactions, since the wide component includes the logistic regression model trained using a feature engineering. It can cause a poor generalization. They investigated to use a factorization machine to automatically capture feature interactions from one-hot-encoded features. Following the structure shown in Figure 7(a), the proposed model, DeepFM, combines the power of factorization machines and deep learning for the feature learning in a recommendation application. The new neural network
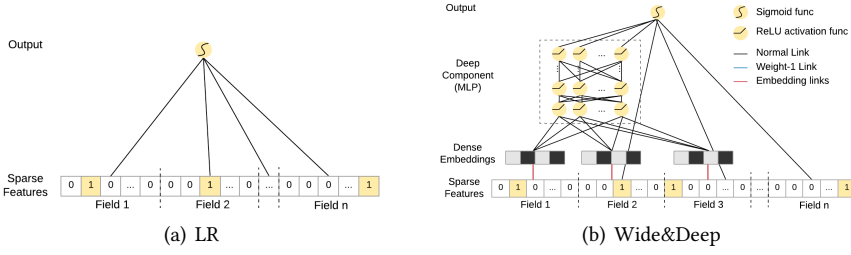
Fig. 6. a) Logistic Regression (LR) model: linear modeling of sparse feature values b) Hybrid model (Wide & Deep model) as the alternative to Factorization Machines to use a deep component to capture higher (≥two) order feature interactions combined with the Logistic Regression addressing low order feature interactions

architecture models linear and 2nd order feature interactions through FM and models high-order feature interactions by fully connected neural network. Replacing the logistic regression with factorization machines and using a shared embedding layer between these two components, they build a model in an end-to-end manner without a feature engineering.

Figure 8 shows the embedding layer structure, which is designed to project discrete feature values to a dense numerical vector space. This projection is modelled by a layer of linear neurons defined on the top of one-hot-encoded input vectors [43]. It includes an embedding matrix of parameters learned for each feature field. Embedding vector representing each categorical field can be shown as follows:

$$e_i = W_i x_i \tag{13}$$

where e_i is the dense embedding vector and x_i is the sparse binary representation. W_i is the embedding matrix for i-field with the dimension of $m_i \times d_i$. m_i denotes the number of discrete values for categorical field i and d_i is the user-defined dimension of dense embeddings. In practice, the functionality of embedding layer is identical to one layer of densely connected neurons without considering bias links and activation functions. It is shown that the embedding matrix W_i can be considered a lookup table for each field. This is because in the case of one-hot-encoded input, the multiplication of input with embedding vectors in Eq. (13) can be replaced by corresponding embedding vectors at referred indices in the embedding matrix. Randomly initialized, the weights w_{ij} in the embedding matrix are trained during the optimization of the target value in different models.
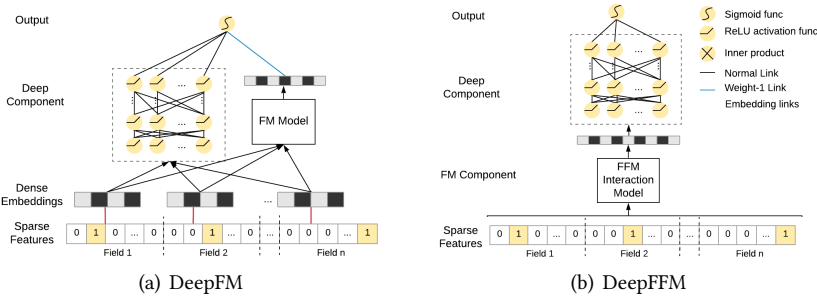
Fig. 7. a)Hybrid model (DeepFM): It is combined by a deep component (fully connected neural network) with a Factorization Machines method(Figure 5(a)) using a shared embedding layer to feed dense embedding vectors as the input to the structure b)Hybrid model (DeepFFM): It cascades the FFM interaction model in Figure 5(b) to a MLP network as the deep component.

In [117] a new hybrid is proposed through combining embedding vectors and a cascade of factorization machines and a MLP network. This method takes advantage of learning ability of neural

Fig. 8. The structure of embedding layer to generate dense embedding vectors. It includes a linear mapping from discrete categorical features represented by one-hot-embedded vectors to dense numerical embedding vectors. It equals to a layer of linear neurons above input layer whose weights are getting trained using gradient descent optimization. The weights are formed in a embedding matrix(lookup table) to accomplish the linear transformation. The rows in the matrix represent embedding vectors for discrete values in the categorical fields

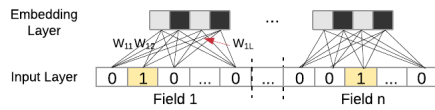networks and discriminative power of latent patterns in a more effective way than MLPs, through adding a product layer between the embedding layer and the first layer of the fully connected neural network. The model is examined with inner and outer product operations in the product layer, to examine different methods to model the feature interactions, combined with Stochastic Gradient Descent training (using L2 regularization) and a dropout mechanism to address the over-fitting issue.

As introduced in Section 4.3, in FFM[57] method as the field-aware factorization machines, each feature value is represented by more than one embedding vectors to model combinatorial features in input space. It addresses different weights for interactions occurring between different feature types.

The large number of features in latent vectors generally cause space complexity problem and memory bottleneck [57]. In addition, DNN based models may run into insensitive gradient issue when dealing with multi-field categorical data which deter the progress of gradient based optimization. To tackle these challenges, a net-in-net architecture is proposed as the generalization of kernel product [118] to model feature interactions. So a micro network including one layer of the fully connected neural network cascaded by dot-product feature latent features is used as the special kernel function to alternate a simple inner-product function in factorization machines.

Following the success of field-aware factorization machines [57] in capturing feature interaction with regard to feature fields information, a study [168] extends this idea to provide a hybrid model of FFM and a fully connected deep neural network to learn feature conjunctions in the input data, as shown in Figure 7(b). In this case, each sparse input feature is represented by multiple embedding vectors to address the effect of feature with regard to the feature field in inner (dot-product) feature interactions. The embedding vector are organized as a 2-dimensional matrix with size of $k \times n$ where k is the dimension of embedding and n is the number of feature fields. Applying $n(n-1)/2$ inner-product calculations between pair of embedding vectors to generate intermediary input vectors, the predicted click-through rate value is generated from the output of Deep component. Field-aware Neural Factorization Machines [177] further extends this method, by introducing a Bi-Interaction operator with a wide concatenation based on Hadmard product operator as the alternative to the inner product layer in Figure 7(b). Using Bi-interaction operator to calculate feature interaction, the dimension of input vector of deep component is changed to $n(n-1)/2 \times k$. The increase of parameters in the embedding vectors of field-aware based factorization machines methods can decrease the prediction performance because of the over-fitting issue. Therefore, it demands to select features before the feature interaction procedure in factorization machines. Compared to Attentional Factorization Machines method [162] which captures important cross features interaction step in FM model, a recent study [176] evaluates the importance of features before applying feature interaction step using Squeeze-Excitation network [53, 176]. The authors [176] introduced an attention-based method to selectively use more informative features in embedding vectors. They propose to apply Compose-Excitation network as the extension of Squeeze-Excitation Networks to select important feature representations.

Generic hybrid methods. Some studies [75, 154] develop a hybrid method to generalize the idea of factorization machines method. The authors in [154] extended the second order feature interaction in factorization machines to higher order levels through a multi-layer network structure where the

maximum level of interaction order is determined by the number of layers. Considering the structure of DeepFM method in Figure 7(a), a cross network is employed for feature crossing operation using equation (14) through weighted dot product of current output vectors at subsequent layers.

$$\mathbf{x}_{l+1} = \mathbf{x}_0 \cdot \mathbf{x}_l^{\top} \cdot w_l + b_l + \mathbf{x}_l \tag{14}$$

where $\mathbf{x}_l \in \mathbb{R}^d$ denotes the output vector calculated at level l and $w_l \in \mathbb{R}^d$ and $b_l \in \mathbb{R}^d$ constitute system parameters in this structure. In the first layer, the dot-product of concatenated embedding vectors $\mathbf{x}_0$ are used to generate the first output. For input data as a multi-field categorical data, the proposed component explains the main difference between this model and factorization machines methods in which dot product interaction here is applied on the concatenation of feature fields rather than between pair of feature fields. The study [75] is the other extended work to create a new cross network in which feature interaction is modeled at a vector-wise level through outer product instead. This leads to generating an embedding matrix in which the operation in each layer has intuitively connection to convolution neural networks by considering the weights as filters.

Deep learning based methods. In literature, various deep learning techniques have been studied for the user response prediction. The majority of previous work following a deep network structure are typically based on two components of embedding and interaction basically designed through deep neural networks to capture non-linear feature interaction in sparse input data. Figure 9 demonstrates the structure of this paradigm. Embedding component is designed to transform the sparse input data into a low-dimensional dense latent space. The embedding vectors are then processed by applying an aggregation mechanism to produce a fixed length vector for the deep component. The high-order interactions between features are addressed through feeding a fixed-length vector into the deep neural network component generally implemented by the multi-layer perceptron [25, 130]. Gradient based training is adopted to learn the non-linear correlation between user features and user responses. In this regard, there are lots of studies conducted to improve the performance of each component. Table 4 demonstrates a summary of representative methods mainly developed based on multi-layer perceptron, recurrent neural networks and convolutional neural networks some of which are combined with attention mechanism design their proposed models.

In online advertising, input features can be gathered from different sources. In Figure 9, sparse binary features representing the input data can be grouped into multiple categories like Table 2 regarding users, advertisers and context. Deep neural network can process input data vectors with a fixed length dimension. However, using a fixed-length vector for users with diverse interests against advertisements can be bottlenecks for prediction, since each user and web-page can have different labels and diversities at the same time. Addressing the variety of user interest generally needs the expansion in the dimension of embedding vector for user features in the aggregation step which increase the risk of over-fitting and the cost of computation.

Dealing with this issue, two sub categories of features such as user profiles and user behaviors [190] are proposed for click-through rate prediction, where an array of user behavior in a period of time attributed by categorical data [100, 190] like visited good, shop and web-page category ids, are used to describe users and their interests. The idea is further followed by [38] to model user behaviors from the continuous image data. In the case of categorical features, since the category of a shop and web-page visited by users may be shown with multiple values, the binary representations are modeled by multi-hot-encoding. They also address the importance of feature interaction in modeling of user behaviors. To this end, they design a local activation unit to provide an adaptive feature representation with regard to different ads, and assign weights to the relevant pair of a visited page and advertisement in the user behavior sequence with regard to the targeted ad. The output vectors are later passed to a weighted sum pooling to generate a fixed length user behavior embedding vector, and then passed into deep component to generate the predicted output value.
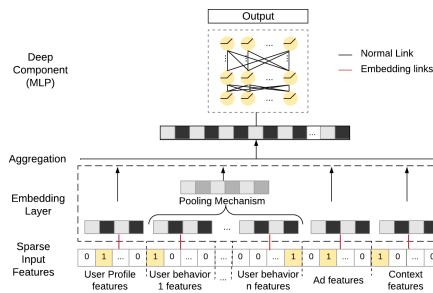
Fig. 9. DNN based architecture (Embedding & MLP) including two main steps embedding layer followed by the aggregation operation to generate a merged vector. The vector is used as the input for second step using MLP to generate predicted user response value(CTR, CVR, ...) developed in different studies. The selected input features are chosen user profile and online user behaviour sequence in addition to those selected for Ads and context

The application of attention units to model user behavior history has also been studied for click-through rate and conversion rate prediction [37, 38, 73, 100, 137, 147, 163]. A multi-head self-attentive networks [137] is proposed where features are mapped to multiple subspaces through multi-head mechanism. This would help the model to consider different orders of feature interaction with adaptive weights. In addition, this method proposes a residual neural network rather than the conventional MLP network to model high order combinatorial feature interactions. The study [100] extends the analysis of user behavior from two temporal and spatial aspects. Because a web-page can be filled with more than one ad, they model user behavior history not only by ads clicked by users but also those not clicked by users. They consider ads shown in the same page above or below the targeted in both spatial and temporal order. Their interactions to targeted ads are modeled by adding an attention based factorization machines layer followed by a fully connected neural network. In [163] the multi-head attention mechanism is adopted to model the user behavior history from a sequence of clicked/purchased advertisement information by adding discriminative features likes dwell time on landing page for a conversion rate prediction task.

Some studies also extend the deep component using different structures of MLP based networks [88, 101, 102, 160]. In order to tackle the data sparsity, researchers [102] propose to consider input features to be fed into a couple of sub-nets built based on MLP network for a feature interaction modeling, using features of users, query and ads entities. The sub-nets are created to model the interactions between user-ad, the correlation between ads. These models are then combined with the third one designed for the prediction in a joint optimization. For conversion events followed after clicking on ads, a deep learning based model is developed [88, 160] to consider not only all clicked ad impressions, but also include all impressions and further user actions like "add to card" (DAction) and "add to wish list" (OAction) taking place before conversion events. Following a multi-task framework, multiple deep components are trained for each event to generate the prediction of conversion post-click rates.

For the embedding, some studies propose to handle categorical and continuous input features at the same time [38, 191]. For images, convolutional neural network have been developed in several studies [19, 38, 166]. In the scale of industrial applications, authors in [38] suggest to generate embedding vectors of images using a pre-trained very deep convolutional neural network rather than employing an end-2-end training model. The convolutional networks are also adopted in [78] to extract implicit features from the sparse input data and deal with the overfitting issue in fully connected based networks [25]. In the proposed model, the convolutional neural network structure designed based on shared weights followed by pooling module can considerably reduce the number of parameters. Considering these embedding vectors along with raw features result aggregated vector to be processed by a deep multi layer perceptron component.

Recurrent neural network based methods. Deep neural networks is typically made of multi layer fully connected neurons. Following a stateless structure for neurons, the independent features incorporate the data that flow through multi-layer perceptrons to generate the output without backward links. Considering independently visited or clicked advertisements to model the user behavior history fail to extract efficient useful user interests with regard to user response prediction. Therefore, recurrent neural networks are developed to process the sequence of input data sorted by time to improve user response prediction performance [15, 36, 74, 110, 181, 189].

In online advertising or e-commerce platforms, user intention is often not explicitly expressed through their behavior history. Therefore, it is hard to identify real interested users only based on captured online user behaviors. In addition, living in a dynamic life environment, people's knowledge increase and their latent interest and behavior may change over time [105], indicating that temporal correlation in online user behaviors can show the evolution in the user interest to have more tendency to advanced items compared to previous converted ones. It leads to studies of developing LSTM models based on user behavior sequence from previous bought and clicked items to predict conversion rates [105]. Following Figure 9, to aggregate embedding vectors, two categories of aggregation modules are generally discussed in different recurrent neural network based studies. The approaches like min-max or sum pooling based aggregation are generally applied to model user behavior from independent input features. In the body of neural network structure, GRU/LSTM neural units were commonly used in many studies to model latent user interests [36, 74, 110, 189].

Comparing to deep neural network based methods, recurrent neural networks suffer from computational and storage overheads. They include hidden states in the structure to capture user interests from sequence of user behavior data. Therefore, it makes it difficult to use these network for industrial applications visiting numerous users and ads everyday. It causes limitations to apply these methods to model long term user interests based on long sequential user history records. To tackle these challenges, some studies introduced memory based architectures [15, 101, 110].

Convolutional neural network based methods. Studying to design deep learning network structures are not limited to above discussed ones, since the input space suffer from the data sparsity which makes it hard to learn directly using simple gradient descent methods. Although the deep neural network including multi-layer perceptrons in theory is known as an universal approximator which has a capacity to capture almost all non-linear feature interaction in input space, but the order of magnitude of parameters used a fully connected neural network deters to capture feature interaction in a sparse feature space and leads to over-fitting issue. This encourages to apply convolutional neural networks (CNN) which benefits from parameter sharing and pooling mechanism to work with a feasible number of parameters [14, 32, 37, 78, 83]. Dealing with image data along with multi field categorical data, CNN networks are used to extract non-linear latent features in the form of embedding vectors for raw pixel image data [19, 38, 41, 97]. As one of the primitive studies, authors in [83] conducted experiments to apply convolution filters followed by a flexible max pooling in a CNN network on two datasets including multi-field categorical data and a series of impressions in an e-commerce platform to capture neighbor patterns in input data. The downside of this method was that they applied the convolution for neighbors field feature while the feature interaction between non-neighbor fields is neglected. However, for user response prediction tasks, any order of feature fields are possible. The order of feature fields in the certain alignment in input data does not have meaningful inference like images or texts. Therefore, the other studies developed methods to take advantage of both CNN and deep multi layer perceptron to address high order and low order feature interactions.

For news recommendation, a knowledge-aware model [147] proposes to use knowledge graph to represent news items, with each news article being attributed by word, contexts, and entity embeddings. For user response prediction, CNN network, previously proposed for sentence representation learning,

is used to generate the final embedding vector of user and ad features. Attentive neural network is applied to address user interest diversity to estimate click-through rate values.

Table 4. Summary of selected DNN based user response prediction methods with a hybrid structure.

Main DNN Structure	System Framework			Method	Features	Application Domain	Predict. Task
	Embedding Component	Deep Component	Aggregation				
MLP	**All features**:RE[a]	MLP	FM/Pooling Layer	DSTN [100]	User, Query, Ad, Context, User clicked ad history	E-commerce[b]	CTR
	All features: RE[a]	MLP	Concatenation	MA-DNN [101]	User, Query, Ad, Memorized user interest (memory link)	E-commerce[b]	CTR
	All features: RE[a] cascaded by FM	MLP	Concatenation	PNN [117]	User behavior ,item , context information	Display advertising	CTR
	All features: RE[a]	MLP alongside FM	Concatenation	DeepFM [44]	User behavior ,item , context information	Display advertising	CTR
	All features: RE[a]	Multiple MLP Stacking	Concatenation	ESMM (ESM^2) [88, 160]	User, Item, users' historical preference scores	E-commerce[b]	CVR
	All features: RE[a]	Multiple MLP Stacking	Concatenation	DeepMCP [102]	User, Query, Ad, Context, negative ad sample features	E-commerce[b]	CTR
	All features: RE[a]	MLP alongside Hadamard product layer	Concatenation	NCF [51]	The identity of Users and Items	Movie/Image recommendation	Ranking
	All features: via Auto- feature grouping and high-order feature interaction selection	MLP alongside FM	Concatenation	AutoGroup [79]	User behavior ,item , context information	Display advertising	CTR
CNN	CNN(Pre-trained CNN model using orthogonal base convolutions)	W&D[22]	Concatenation	W&D SSM [97]	User behavior ,item , context information	Display Advertising	CTR
	Image ad features: CNN subnet **Other Ad features**: fully connected subnet	MLP	Batch Normalization	DeepCTR [19]	Image,categorical (Impression)	Display Advertising	CTR
	All features:RE[a] augmented to CNN subnet	MLP	Concatenation	FGCNN [78]	Categorical+continuous features in display advertising	Display Advertising	CTR
	Ad,Query features (under character-level): 1d CNN subnet, Ad,Query features (word-level)	MLP	Cross-convolutional pooling	DCP/DWP [32]	Ad, Query	Sponsored search	CTR
RNN	**Ad, context features**:RE[a] **User behavior features**: Hierarchical GRU based memory network	MLP	Concatenation	HPMN [120]	User behavior sequence,item context info	E-commerce[b]	CTR
	All features: RE[a]	MLP LSTM	Concatenation	NTF [161]	User behavior sequence,item, time	Recommendation	Product rate
	Ad, context features:RE[a] **User behavior features**: Memory induction GRU based unit	MLP	Concatenation	MIMN [110]	User behavior sequence,item context info	Display advertising	CTR
Neural Attention	RE[a]	Multi-head ResNet	Concatenation	AutoInt [137]	User profile, item attributes	Display advertising	CTR
	user profile, Ad features: RE[a] **User behavior features**: Self Attentive Bi-LSTM	MLP	Concatenation	DSIN [36]	User profile, User behavior sequence, item,	Display advertising	CTR
	User behavior (event) sequence: RE[a] **User behavior (timestep) sequence**: Attentive embeddings followed Bi-GRU	MLP	Attention Mechanism	DTAIN [40]	Event, Timestep information	E-commerce[b]	CTR
	user profile, Ad, context features:RE[a] **User behavior features**: self Attentive GRU relative to target ad	MLP	Concatenation	DIEN [189]	User profile, User behavior sequence, item,	E-commerce[b] Display advertising	CTR
	User behavior sequence features: RE[a] controlled by multi-head self-attention structure **Other features**: RE[a]	Jointly training two MLP stacks for CTR and CVR	Concatenation	PFD+MD [163]	User, Item, Post- click info, User Clicked/Purchased sequence, User-item interaction statistical info	Display Advertising	CVR, CTR
	User profile features: RE[a] User behavior sequence features based on Ad image: Pre-trained embeddings	MLP	Concatenation, Sum/Max/ Pooling	DICM [38]	User, Ad (image), user behavior sequence(image)	E-commerce[b]	CTR
	Pre-trained knowledge graph Word embeddings combined with, entity and context embeddings via CNN	MLP	Attentive pooling, Concatenation	DKN [147]	User (clicked news item), News item	News recommendation	CTR
	Query & ad under word-level: 1)Pre-trained word embedding, 2) regular embedding[a] Followed by bi-LSTM and MLP	CNN MLP	Pooling, Query-Ad tensor matching	DSM [41]	Ad(title, URL, description), query words	Sponsored search	CTR

[a] Regular Embedding using trainable look-up table parameters (matrix embedding per feature) following the structure shown in Figure 8

[b] In the e-commerce scenarios, the prediction task is defined as the the probability that user clicks or makes an conversion on the recommended items(ads)

Other methods. In the previous section, we have provided an review on classification methods ranging from linear logistic regression based methods to advanced deep learning based methods. A few other classification based methods which may gain attention are generative adversarial network (GAN) based models [28, 72], transfer learning [138, 178], fuzzy design [56], decision trees [46, 159] and multi task learning [104, 149] approaches.

4.3.1 Ensemble Based Approaches. While early proposals for user response prediction mainly use linear logistic regression classifiers, which provide the simplicity along with the scalability, modern approaches are developed to address non-linear interactions in data using methods likes factorization machines, generalized version of decision trees, and neural networks. Some studies show that using a single machine learning method may lead to non-optimal results, and propose a new aspect of development to design a model structured from an ensemble of machine learning models. These models can bring more improvement in the level of accuracy for the prediction task. Generally, the design of ensemble models are mainly categorized into four sections like Bagging and Boosting, Stacked Generalization and Cascading shown in Figure 10 [4].
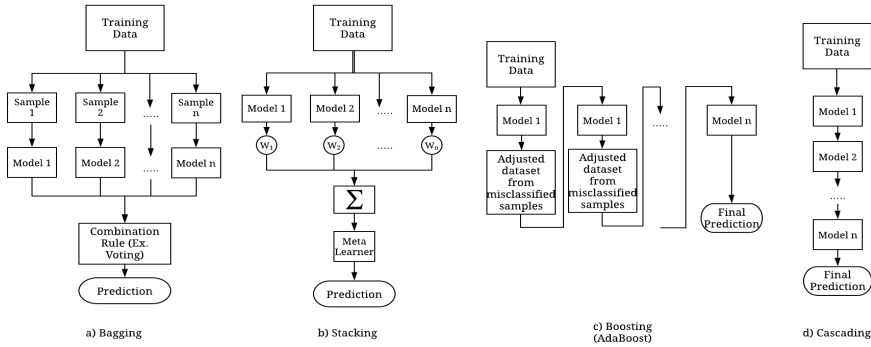

Fig. 10. Ensemble structure types: a) Bagging randomly samples training data, with replacement, to generate N subset data and train N predictor models. The final result is the combination of N classifier outputs; b) stacking: N models are trained in parallel based on the same training data. The final output is combined through a meta-classifier. This classifier is fitted on the output of base classifiers; c) Boosting(AdaBoost): a series of predictor models are trained using a subset of training data sequentially. The subsets of data are created adaptively using misclassified samples in previous model; d) Cascading: is based on concatenation of multiple classifiers

The combination of different classification methods in the form of ensemble structures are utilized in different studies. The study in [179] followed a cascading version of an ensemble model which includes two learners. They investigated the performance of combining factorization machines with a fully connected neural network to predict CTR values for the digital advertising. Because of the data sparsity in the categorical input data, the feature interaction cannot be easily detected directly using deep neural networks which generally lead to the overfitting issue. They propose to cascade factorization machines to a deep neural network in order to address this issue.

In the context of e-commerce websites, [5] suggested multi-modal ensemble learning to consider texts and images of posts as different modalities. They separately built a logistic regression model for historical CTR values and another model for embedding vectors of images and textual information. Following the multi-modal learning approach, the stacking ensemble model is used to combine linearly their results by passing to the final logistic regression classifier.

In another study, authors in [159] propose to develop an ensemble model for conversion rate prediction which is mainly based on GDBT learners. Following the Cascading and stacking techniques, they used multi-level cascade of GBDT models to extract features which are coming for values received from the previous model. To improve the diversity of extracted features, multiple cascade of decision trees are aggregated like Figure 11 through the concatenation to be passed to the conclusive GDBT to generate the final features for the classification. As a part of the contribution of this work, in order to improve the prediction performance, the importance of input features is also considered. They use a separate GBDT model to pre-process the input raw features and generate two class of features that

have weak and strong correlations (WCF and SCF) with regard to the prediction result. These class of features are used relatively as the input to train the model.
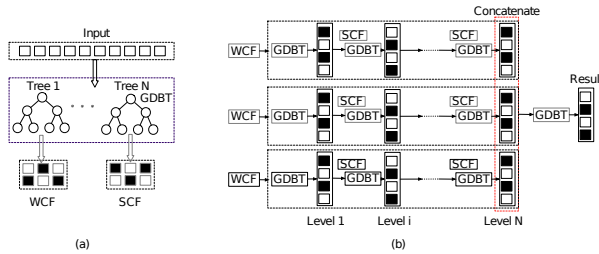

Fig. 11. Ensemble GBDT model: a) Input features randomly selected from the feature pool fed into GDBT model. Their importance for prediction is evaluated based on their correlation to class labels by traversing from the root node into leaf in a tree. They are sorted into two categories according to the explanatory power into WCF(Weak correlation features) and SCF(Strong correlation features) b)The structure of proposed multi-level deep cascade trees including stacking multiple sequence of GDBTs

In literature, some works also comparatively study the effect of ensemble techniques [60, 77]. Following the ensemble techniques introduced in the beginning part and the goal to improve click-through rate values, the study in [77] examined two ensemble techniques Boosting and Cascading with GBDT, LR and a fully connected deep neural network classifiers. They compared the performance of corresponding single learners with the cascaded and boosted version of pair of models in the click-through rate prediction for the sponsored search advertising. For the sponsored search advertising application again, the study in [60] made a comparison between the effect of four different structures of ensemble learning such as the majority voting, bagging, boosting and stacking for pay-per-click classification. The features are selected from different information sources like the attributes describing ad impression, click-through rate value, conversion rate value, and the position of ads in addition to the textual features captured from the title and the body of ads and campaign categories. They are joined together to train ensemble learners such as Naïve Bayes, Logistic Regression, Decision tree and SVM to estimate the pay-per-click value of campaigns.

4.4 Unsupervised and Semi-supervised Approaches
In this section, we review two categories of methods in the literature that do not fully rely on labeled data. In this case, the predictive models are designed based on the implicit and explicit pattern in data. Semi-supervised models refers to approaches like graph neural network based models that involve designing a user feedback estimation model using both labeled and un-labeled sampled data. Two categories of these methods are represented in the following sub-sections.

4.4.1 Clustering Based Approaches. Clustering methods have also been investigated in the literature for online advertising. As an unsupervised approach, clustering involves grouping sample data into related clusters based on similarity among data points.

Some studies develop statistical clustering methods categorical data in different contexts, such as k-modes [131] as an alternative to the popular k-means method which uses hamming distance as the distance metric, COOLCAT made by [7] uses the notion of entropy for the similarity metric, or CLOPE clustering approach in [167] research which develops a scalable method to leverage the trend of a height-width ratio of the cluster histogram as the similarity criterion. But these categorical clustering methods are not well studied in the online advertising with the multi-field categorical data.

On the other hand, some clustering based studies proposed the idea to improve classification based methods. As the initial study in this category, to supplement the logistic regression [17] and gradient boosting decision trees (GBDT) [47] methods for user response prediction task , authors in [119]

propose to use feedback features which are prepared from historical user behaviors. Considering advertiser and publisher come from with different hierarchical granularity, they incorporated the combination of publisher page and advertisement along with user-publisher-creative which are created based on the hierarchical structure of user as the new features. The extra features are quantized using the *k*-means clustering to be added to input features for training.

Some methods [119, 124, 158] organize the user input, such as keywords used in search engine or pages visited by users, by using clustering to reduce the severity of above mentioned issues and improve the correlation with user responses. In [119], the authors suggest that since a different click response probability can be assumed for different query keywords, the topic of ads and query keywords can be used to organize data into clusters where more closely related terms have more similar click-through rate values. They propose to generate groups of terms using hierarchical clustering and keyword-advertiser matrix. The similarity of samples intra and between clusters are evaluated by the textual similarity of terms in ads. Therefore, assuming the fixed CTR value for clusters, the estimated value of click-through rate for new samples is determined by the nearest neighbor clusters.

4.4.2 Network Based Framework. With development on the Internet, information networks are the common element of online businesses. In this subsection, we will go over approaches addressing the network structure in the input data to develop predicting models for the user response prediction.

Graph embedding based methods. Recent years have seen a lot of studies which focused on the application of network representation learning methods for recommender systems and the user response prediction. Motivated by the success of CNN and RNN, there has been an interest in developing neural network based models for the graph structured data. Considering three major challenges in recommender systems, scalability, data sparsity, and cold start, many methods have been proposed in the literature using graph embedding [150] and graph neural network [35, 59, 74].
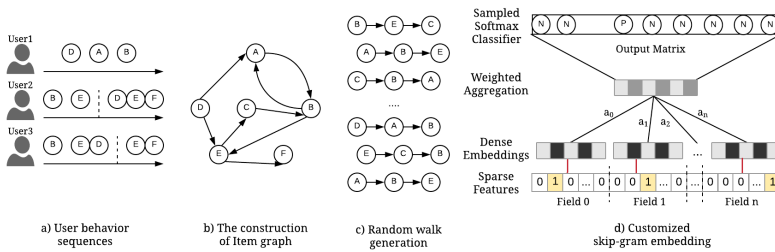

Fig. 12. The billion-scale commodity embedding proposed by [150] for E-commerce recommendation in Alibaba: a) user behavior sequences including items(Ads) visited in one or more session specified by dash lines. b) item(ad) graph generated user behavior sequences that a direct edge representing two subsequent items(ad) in user history. c) generating sequence of nodes using random walk in graph following Deepwalk method. d) Proposed graph embedding algorithm to use side information to reduce data sparsity. Field 0 specifies a node in random walk and field 1 to field *n* are the one-hot-encoded vectors for side information corresponding to the ad node in the graph. Hidden layer is a weighted average aggregation of dense embedding vectors

Authors in [150] designed a recommender model based on the graph embedding which takes advantage of side information to cope with the three challenges. The model includes two sections such as matching and ranking. Focusing on the first section with the network of users interacting with items in an e-commerce website, they applied DeepWalk [109] to generate embedding vectors of items in the directed graph of items formed using user online behavioral history. Because of the data sparsity, there is a lack in the number of interaction in the graph. Therefore, as the part of the contribution, side information such as the price value, shop, category and brand, are included as the one-hot encoded vectors in network representation learning procedure.

Graph Neural Networks (GNN) are known as one the most effective solutions to develop predictive models for network based data. Extended from recurrent neural networks and convolutional neural networks, GNNs have a unique capability to generalize neural networks to cope with directed and undirected input graphs, by using an iterative process to propagate node status over their neighborhoods. After optimization, it can provide an embedding output of nodes in the graph in which the feature information are aggregated using their neighbor nodes. Recently, GNN based methods have received more attention for online advertising and recommendation systems applications [35, 59, 74, 156, 157, 174]. Like deep learning based models in Table 4, Embedding and Interaction components are two major components in the design of these predictive models in the literature. The embedding component is dedicated to map information associated to node and edges in the graph-structured data into numeric vectors. However, the interaction component tries to make the reconstruction of user feedback in the system. This part can be designed by different ways through a MLP component in deep learning based models [157] or inner-product between embedding vectors [156] or element-wise multiplication of embeddings [174] to represent collaboration between users and items to rank and predict user preferences in recommendation and online advertising systems. The embedding component is developed using recent advance in graph neural networks. Authors in [50, 156] propose an propagation layer in their embedding component to refine embedding vectors via aggregation of embedding vectors of neighboring nodes in the graph. The model in [156] is built using the message-passing architecture and defining a model via layer-wise message constructing and aggregating operations. A recent study [74] addresses the limitation of methods like DeepFM [44], by considering a graph structure for feature fields in the input data in which nodes corresponding to feature fields interact with others through weighted edges to reflect the importance of field interactions. A GNN based model is developed to model complex interaction between input field features. In this model, field feature as the nodes in graph are attributed by hidden state vector which is updated using a recurrent approach. An interaction step parameter is defined to consider higher and lower interactions between nodes and their neighbor nodes which are located in one or more hops away. The ending point of the model includes an attention layer to predict CTR values.

One essential challenge of network embedding for the user response prediction is that the embedding learning might not be directly optimized towards the underlying user response prediction. Following an unsupervised learning, nodes are represented using embedding vectors, however, they may not be optimized for downstream tasks like the click-through rate prediction. This issue can be considered as the bottleneck to improve the task. Therefore, the user intention modeling is considered as an alternative [189, 190]. Considering the sequence of the user behavior from the interaction between users and ads in the user intent modeling still have some challenges like the data sparsity and weak generalization. A study [71] showed that sequence of user behavior can be organized as a co-occurrence commodity graph with node representing clicked commodities and weighted edges describing number of co-occurrence times. To address the sparsity problem, a multi-layered neighbor diffusion is performed on the commodity graph. The preceding result is combined with using an attention layer to generate user intention features. These features combine with other ones, such as user profiles, query keywords, and context in fully connected network, for the click-through prediction.

Knowledge graph based methods. Knowledge graphs (KG) are semantic heterogeneous networks including a collection of entities with attributes that are inter-connected together through edges. They are usually described through a triplet with relation connecting head and tail entities like (Head, Relation, Tail). This structure of data has been studied for different applications like link prediction [30] and Web search analysis [143]. The advantages of the knowledge graph for applications suffering from data sparsity and cold start problems, such as recommendation systems, have been observed from different perspectives. First, networks can provide additional semantic relationship information to improve the recommendation performance. Moreover, the diversity of information in

the knowledge graph can extend the information for matching with user interests. In addition, the historical information linked to items in recommender systems can provide implication ability for the system. In summary, we separate knowledge graph based methods for recommender systems and online advertising into three categories: embedding-based, path-based, and hybrid methods.

Network embedding methods This category of methods aim to map the components of the knowledge graph, including entities and relations, into low-dimensional embedding space to preserve network structural information [153]. Some jointly incorporate heterogeneous attributes and content that are assigned to nodes in the graph for modeling [174].

For example, knowledge graph based representation for news recommendation [147] has been studied to address the challenge of topic and time sensitivity for news items selected and visited by users. It means that users generally visit selected news at a short specific of time which may not happen later. In addition, news content usually has brief words and diverse topics. To handle these challenges, a deep neural network is proposed to take advantage of a customized CNN module as the key component to model user interest through multiple channels which consider both word semantics information and corresponding information from a generated knowledge graph data. This leads to generating three categories of embedding vectors for words in the body of news, the associated entity and immediate neighbors in knowledge graph. In the design, an attention mechanism is used to aggregate embedding vectors of user behavior sequences. They avoid using concatenation strategy for the aggregation in this step since entity and word embeddings may have different dimension generated from different contexts. The output are fed into a fully connected neural network to learn the probability of user's click for a selected news piece. Likewise, a study [174] presents a heterogeneous graph neural network model which adopts the aggregation of feature information with regard to a sampled neighboring nodes. A node sampling procedure is suggested to aggregate selected neighbors grouped by their types and their frequency in a designed random walks. Using attention mechanism, the content embedding of neighbor nodes with the same type are first aggregated. It is then followed by another attention round to aggregate embedding vectors of neighbor nodes from different types in the graph. They train embeddings using heterogeneous skip-gram learning. To compare the performance of proposed model, element-wise multiplication and inner-product operations of user and items embedding vectors are used to simulate user response for link prediction and recommendation experiments.

Meta-path based methods This category contains knowledge graph embedding methods which employ meta-path schemes as the guideline to generate random walks and in turn embedding vectors. Although many studies in the category have shown a decent performance for recommender systems [35, 175], current methods heavily rely on manually building random walk corpus for further processes. The selection of meta-path schemes are generally considered as the hyper-parameters set differently by researchers in experiments. So this can be an issue in practice. To tackling this problem, attention mechanism has been employed in recent studies. Authors in [157] design a heterogeneous graph neural network to automatically address the effect of different neighboring nodes and meta-paths using two-level attention layers. In the first level, node-level attention is applied to train the weights for meta-path guided neighbors of each node in the graph. It is then fed to a semantic attention step to calculate weighted combination of different meta-paths for the node embeddings. The predicted interaction between different node types in heterogeneous graph is modelled through training a fully connected neural network at the end.

Other knowledge based methods In this section, we present hybrid knowledge based methods which learn user/item embeddings by exploiting structural information in the knowledge graphs [116, 148]. Recently, a study [148] discusses the extension of GNN method made for a knowledge graph where the edge weights between user and item nodes are not available beforehand. So a personalized scoring function is proposed for training to determine the edge weights via a supervised approach following a relational heterogeneity principle in the knowledge graph. To address the data sparsity issue in

recommendation systems, a leave-one-out loss function is used as a label smoothness regularization to calculate predicted weight values. It leads to calculating node embeddings through aggregating node's feature information over the local neighborhood of the item node with different weights.

A common approach to model user response in knowledge graph based methods is to apply aggregation mechanism to combine embedding vectors of user and items entities via average pooling or attention units over their neighbors. Authors in [116] consider this as early summarization problem. They argue that modeling user response using the inner product of embedding vectors of user and item can have a limitation for user response prediction. Accordingly, a neighborhood interaction model is proposed to integrate a higher order neighbor-neighbor interactions through a bi-attention network in the aggregation step to improve user click through rate prediction.

4.5 Stream Based Framework

Online advertising is essentially a streaming platform, where users, auctions, and ads are continuously and dynamically changing [151]. In this context, data stream refers to continuous feeds of news and information generated by users in an interactive way [69]. Social media platforms are examples of these systems in which millions of users generate data continuously being uploaded. The stream environment provides an opportunity to emerge in-stream advertising with commercials in stream of data. Also known as native advertising, in-stream ads look similar to regular feeds. They are differentiated by an assigned tag indicating a commercial target or the content of feed.

The performance of advertising strategies for stream data has been studied from different aspects according to the condition and policies in online platforms. Click-through rate value is not only a metric to evaluate user experiences. Many studies have developed methods to address pre-click and post-click user experiences [9, 67, 192]. From a different perspective, user response prediction was cast to evaluate ad quality. The high rate of quality value is considered as the positive influence for users to use the platform more and produce even more click responses for the long run. There are some studies experimenting a model to address the impact of ads quality for predicting user response and user engagement, based on in-app advertising such as Yahoo Gemini platform [9, 67]. To this end, a post-click experience instead of click-through rate value is used to evaluate the user experience on the landing page of advertising web-sites. Post-click experience is attributed by metrics like dwell time and the bounce rate. The former measures the spending time in the landing page where the latter indicates the percentage of short and momentary dwell times. The level of user engagement with ads is considered to have a natural connection to the time length users spend in landing websites.

Aside from ad quality metric, in the context of social media, CTR prediction for stream data in Twitter is first studied [70], where positive use responses are defined as retweet, reply, and actual click on promoted tweets. They also use a dismiss feature in Twitter to identify explicit negative instances for the analysis. According to the fact that the number of spots dedicated for promoted tweets are limited, in this study a learning-to-rank method with a calibration mechanism is proposed to combine traditional classification with pair-wise learning to address data sparsity and scalability issues. They formalize two problems of classification and ranking in the framework.

In an alternative work [29], time-sensitivity of streaming data in Twitter and the short memory issue for online learning are studied to exclude obsolete tweets from being considered. Therefore, authors propose to analyze hashtags[3] in social media as the indicators of user interests to provide a personalized ranking of topics. They present an online collaborative filtering method following pairwise ranking approach for matrix factorization (Stream Ranking Matrix Factorization), and propose a pairwise learning to optimize an ordinal loss and a selective negative sampling based on a selective active learning, using three objective losses, including hinge loss, SVM, and RankSVM for training.

---

[3]Hashtags are prefixed expressions using the symbol of # to be used for marking a specific topic in Twitter

Recently, authors in [64] have centered their work on delayed positive feedback at stream media to study the effect of two factors, such as the trend and seasonality, in online advertising. In live streams, the predicting models are dealt with the cold start issue. This is because in online real-time scenarios, fresh data lack enough label information and the few appearance of the positive response of users, which leads to the underestimation of CTR values. They conduct experiments to estimate CTR values for video ads in Twitter platform, and examine predicting models with logistic regression and Wide& Deep [22] models using five loss function designed for delayed positive samples to identify the combination of learners and loss functions for continuous stream data.

4.6 Summary

To summarize different framework covered in above sections, Table 5 outlines main learning strategies used by different methods. In Table 6, we also outline studied methods from a different aspects, including feature engineering, downstream tasks used for the evaluation of models, and domain applications. Recent years have witnessed a significant growth in networking technologies and a larger number of online users across the world. As a result, scalability is a major challenge for recommender and online advertising. In Table 7 we overview different efforts made to provide technical solutions for user response prediction in real world applications. Comparing with academic scale solutions, models deployed for production system need massive resources to store and execute internal processes. To address these requirements, industry attempts to devise paralleled model and data architectures that data can be processed with high throughput and remarkably low latency. Recently some work [90, 96] focus on developing benchmark framework suites to provide adequate flexibility along with good test results to make fair comparisons between academic and industrial models. In Appendix D.4, we also outline some potential directions for future studies.

Table 5. Overview of main ideas of user response prediction methods along with pros and cons

Learning Strategy	Algorithm	Advantages	Disadvantages
Data Hierarchy Analysis	Clustering based	+ Using clusters as an auxiliary information for samples with insufficient observations	- May have a variation in user response rates
Matrix Factorization	Collaborative Filtering	+ Good scalability along with simplicity + It can provide robust performance against sparse data	-Explore all historical data -Weak on anonymous behaviour sequences - No promised performance in case of lack of user info due to privacy issues
Training a classifier	LR	+ Scalability	- Needs feature engineering
	FM based	+ Have a closed form equation that can be calculated in a linear time	- Limited to model 2nd order feature interactions
Feature Learning + Training a classifier	DNN based	+ end-to-end interface with representation learning and non-linear transformation + High flexibility using a modular implementation via open-source frameworks	- Interpretability - Prone to over-fitting due to requirement of large amount of input data - Hyper-parameter tuning issue
	RNN based	+ Can learn from sequential data with variable lengths + Robust performance with regard to data sparsity	- Rely on linear sequential structure; - Hard to take full advantages of GPU/TPU computing architectures; - Long training time
	GNN based	+ Addressing the network structure in the input data to aggregate feature information of neighboring nodes + Joining with attention mechanism to provide good interpretability	- A model trained cannot be directly applied to an input graph with different structure - Computational cost
Stream based framework		+ Adjust prediction in user preferences over time + Joining with external memory network for increment updates + Reservoir technique to use more samples to update the model	- Unpractical to stack up the training data for modeling

5 CONCLUSION

This survey provides a comprehensive overview of computational methods for user response prediction in online advertising. Our goal is to provide a detailed review and categorization of the online advertising ecosystem, stakeholders, data sources, and technical solutions. To achieve the goal, we review and categorize online advertising platforms, type of user responses, data sources and features, and propose a taxonomy to characterize main stream approaches for user response prediction. For each type of user response prediction methods, we also briefly study technical details of representative methods, with a focus on machine learning, especially deep learning, based approaches. In addition

Table 6. Comparing user response prediction methods in terms of feature characteristics, application domains, and downstream tasks.

Feature Types /Organization		Application Domain			Prediction Task (Publications)
		E-commerce	Display Advertising	Recomm. Systems	
Feature Engineering			✓		(1) CVR ([68, 86, 145]); (2) CTR ([17, 27, 48, 65, 121, 180, 193]); (3) Ranking([25]
Feature Learning	Collaborative Filtering Based	✓	✓	✓	(1) CTR ([82, 95]); (2) Ranking([82]); (3) Product Rating([161])
	Multi-field (categorical)			✓	(1) CTR([22, 53]), (2) Ranking([51])
	Textual		✓	✓	(1) CTR ([6, 33], Deep(Char) WordMatch [32], DSM [41]); (2) Ranking ([41])
	Visual	✓			(1) CTR(DICM[38],[19]), (2) Ranking(ACF[20], PinSage[169])
	Sequential	✓	✓	✓	(1) CTR(DSIN[36], DIEN[189], DIN[190], RNN[181], MIMN[110]), (2) CVR(GMP[15], DTAIN[40]), (3) Ranking(FGNN[115], GAG[126], LGSR[170])
	Network based	✓	✓		(1) CTR(FiGNN[74], GIN[71],[150], KNI[116]); (2) Ranking(KGNN-LS[148])
	Hybrid	✓		✓	(1) CTR([5], RippleNet[146], MKR[149], DKN[147]), (2) Ranking(RippleNet[146], MKR[149])

Table 7. Summary of selected practical solution applied in industrial environments

Algorithm	Challenge	Introduced Strategy	Application Domain	Provider
AdPredictor[42]	Scalability	Bayesian probit regression model, Weight pruning, Parallel training	Sponsored search ads	Microsoft Bing
EtsyCTR[5]	Dealing with image data	Transfer learning, Feature Hashing, Ensemble model	E-commerce	Etsy
FBCTR[48]	Massive data	Uniform sub-sampling, Cascade of classifiers, Ensemble model	Display ads	Facebook
DLRM[96]	Memory constraints in embeddings and computational costs of DL components	Using PyTorch and Caffe2 for model and data parallelism	Recomm. Sys	Facebook
DeepFM[45] PIN[118]	Insensitive gradient issue in DNN based models and space complexity of FFM-based models	Shared embedding vectors, an end-to-end prediction model[45], Net-in-Net architecture to combine FM and DNN units[118]	Recomm. Sys	Huawei
DIN[190] DIEN[189]	Large number of DNN parameters, Addressing temporal drifts in user interests representation	Mini-batch aware regularization and local adaptive activation function[190], Attention based user interest extractor layer[189]	Display ads	Alibaba
MIMN[110]	Handling long user behavior sequences	Multi-channel memory network	Display ads	Alibaba
HPMN[120] UBR4CTR[114] SIM[113]	Tackling long sequential user behaviours	Memory network model along with a GRU network[120]. Self-attentive retrieval module to select relevant user behaviors[114]. Cascaded two-stage search model[113].	E-commerce	Alibaba
EGES[150]	Scalability	Graph Embedding Using XTensorflow	Recomm. Sys	Taobao
DICM[38]	Dealing with images to represent user behaviours	A distributed model server to handle image data embedding and reduce the communication latency	Display ads	Taobao
RAM[185]	Balance immediate advertising revenue and long-run user experience	Joint optimization using two-level reinforcement learning	Recomm. Sys	ByteDance
HPS-4[184]	Massive model with large number of parameters	A distributed hierarchical GPU parameter server	Display ads	Baidu
PinSage[169]	Massive input graph with billions of links	Highly scalable graph convolutional network	Recomm. Sys	Pinterest
DCN_V2[155]	Controlling the number of model parameters to learn feature interactions for real-time data	Mixture of low-rank approximation of DCN method[154] organized in stacked and parallel structures	Recomm. Sys	Google

to the algorithms, we also review user response prediction applications, benchmark data, and open source codes. The survey delivers a first-hand guideline for industry and academia to comprehend the state-of-the-art. It also serves as a technical reference for practitioners and developers to design their own computational approaches for user response prediction.

6 ACKNOWLEDGMENTS

REFERENCES

[1] Deepak Agarwal, Rahul Agrawal, Rajiv Khanna, and Nagaraj Kota. 2010. Estimating Rates of Rare Events with Multiple Hierarchies through Scalable Log-Linear Models. In *KDD*. 213–222.

[2] Deepak Agarwal, Andrei Zary Broder, Deepayan Chakrabarti, Dejan Diklic, Vanja Josifovski, and Mayssam Sayyadian. 2007. Estimating rates of rare events at multiple resolutions. In *KDD*. 16–25.

[3] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2018. Target Apps Selection: Towards a Unified Search Framework for Mobile Devices. *SIGIR*, 215–224.

[4] Ethem Alpaydin. 2010. *Introduction to Machine Learning* (2nd ed.). The MIT Press.

[5] Kamelia Aryafar, Devin Guillory, and Liangjie Hong. 2017. An Ensemble-based Approach to Click-Through Rate Prediction for Promoted Listings at Etsy. *CoRR* abs/1711.01377 (2017).

[6] Afroze Ibrahim Baqapuri and Ilya Trofimov. 2014. Using Neural Networks for Click Prediction of Sponsored Search. *CoRR* abs/1412.6601 (2014).

[7] D. Barbará, Y. Li, and J. Couto. 2002. COOLCAT: an entropy-based algorithm for categorical clustering. In *CIKM*. 582–589.

[8] Eduardo Barbaro, Eoin Martino Grua, Ivano Malavolta, Mirjana Stercevic, Esther Weusthof, and Jeroen van den Hoven. 2019. Modelling and predicting User Engagement in mobile applications. *J. of Data Science* (2019), 1–17.

[9] Nicola Barbieri, Fabrizio Silvestri, and Mounia Lalmas. 2016. Improving Post-Click User Engagement on Native Ads via Survival Analysis. In *WWW*. 761–770.

[10] Shawn D Baron, Caryn Brouwer, and Amaya Garbayo. 2014. A model for delivering branding value through high-impact digital advertising. *J. of Advertising Research*, 286–291.

[11] Sonja Bidmon and Johanna Röttl. 2018. *Advertising Effects of In-Game-Advertising vs. In-App-Advertising*. Springer, 73–86.

[12] L. Bigon, G. Cassani, C. Greco, L. Lacasa, M. Pavoni, A. Polonioli, and J. Tagliabue. 2019. Prediction is very hard, especially about conversion. Predicting user purchases from clickstream data in fashion e-commerce. *CoRR* abs/1907.00400 (2019).

[13] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata. 2016. Higher-Order Factorization Machines. In *NIPS*. 3359–3367.

[14] Patrick P. K. Chan, Xian Hu, Lili Zhao, Daniel S. Yeung, Dapeng Liu, and Lei Xiao. 2018. Convolutional Neural Networks based Click-Through Rate Prediction with Multiple Feature Sequences. In *IJCAI*. 2007–2013.

[15] Xuchao Zhang Chuxu Zhang Jiashu Zhao Dawei Yin Chao Huang, Xian Wu and Nitesh Chawla. 2019. Online Purchase Prediction via Multi-Scale Modeling of Behavior Dynamics. *KDD* (2019), 2613–2622.

[16] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *KDD*. 1097–1105.

[17] Olivier Chapelle, Eren Manavoglu, and Rómer Rosales. 2014. Simple and Scalable Response Prediction for Display Advertising. *J. of TIST* 5 (2014), 61:1–61:34.

[18] Gong Chen, Jacob H. Cox, A. Selcuk Uluagac, and John A. Copeland. 2016. In-Depth Survey of Digital Advertising Technologies. *J. of IEEE COMST* 18 (2016), 2124–2148.

[19] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep CTR Prediction in Display Advertising. *CoRR* abs/1609.06018 (2016).

[20] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *SIGIR*. 335–344.

[21] Wenqiang Chen, Lizhang Zhan, Yuanlong Ci, Minghua Yang, Chen Lin, and Dugang Liu. 2019. FLEN: Leveraging Field for Scalable CTR Prediction. *CoRR* abs/1911.04690 (2019).

[22] H. Cheng, L. Koc, J. Harmsen, and et al. 2016. Wide & Deep Learning for Recommender Systems. In *DLRS*. 7–10.

[23] Hana Choi, Carl F. Mela, Santiago R. Balseiro, and Adam Leary. 2019. Online Display Advertising Markets: A Literature Review and Future Directions. *J. of Information Systems Research* 31, 556–575.

[24] Shu-Chuan Chu. 2011. Viral advertising in social media: Participation in Facebook groups and responses among college-aged users. *J. of Interactive Advertising* 12 (2011), 30–43.

[25] P. Covington, J. Adams, and E. Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *RecSys*. 191–198.

[26] Brian Dalessandro, Daizhuo Chen, Troy Raeder, Claudia Perlich, Melinda Han Williams, and Foster Provost. 2014. Scalable hands-free transfer learning for online advertising. In *KDD*. 1573–1582.

[27] Brian Dalessandro, Rod Hook, Claudia Perlich, and Foster Provost. 2015. Evaluating and Optimizing Online Advertising: Forget the Click, but There Are Good Proxies. *J. of Big data* 3 (2015), 90–102.

[28] Y. Deng, Y. Shen, and H. Jin. 2017. Disguise Adversarial Networks for Click-through Rate Prediction. In *IJCAI*. 1589–1595.

[29] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. 2012. Real-time top-n recommendation in social streams. In *RecSys*. 59–66.

[30] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. 2011. Temporal Link Prediction Using Matrix and Tensor Factorizations. *J. of ACM TKDD* 5 (2011), 10:1–10:27.

[31] N. Ebadi, B. Lwowski, M. Jaloli, and P. Rad. 2019. Implicit Life Event Discovery From Call Transcripts Using Temporal Input Transformation Network. *IEEE Access* 7 (2019), 172178–172189.

[32] Bora Edizel, Amin Mantrach, and Xiao Bai. 2017. Deep Character-Level Click-Through Rate Prediction for Sponsored Search. *CoRR* abs/1707.02158 (2017).

[33] Muhammad Junaid Effendi and Syed Abbas Ali. 2017. Click Through Rate Prediction for Contextual Advertisment Using Linear Regression. *CoRR* abs/1701.08744 (2017).

[34] Maurizio F., Paolo C., and Dietmar J. 2020. Methodological Issues in Recommender Systems Research. In *IJCAI*. 4706–4710.

[35] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. *KDD* (2019), 2478–2486.

[36] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. In *IJCAI*. 2301–2307.

[37] Hongchang Gao, Deguang Kong, Miao Lu, Xiao Bai, and Jian Yang. 2018. Attention Convolutional Neural Network for Advertiser-level Click-through Rate Forecasting. In *WWW*. 1855–1864.

[38] T. Ge, L. Zhao, G. Zhou, and et al. 2018. Image Matters: Visually Modeling User Behaviors Using Advanced Model Server. *CIKM* (2018), 2087–2095.

[39] Zhabiz Gharibshah, Xingquan Zhu, Arthur Hainline, and M. Conway. 2020. Deep Learning for User Interest and Response Prediction in Online Display Advertising. *J. of Springer DSE* 5 (2020), 12–26.

[40] Djordje Gligorijevic, Jelena Gligorijevic, and A. Flores. 2019. Time-Aware Prospective Modeling of Users for Online Display Advertising. *CoRR* abs/1911.05100 (2019).

[41] Jelena Gligorijevic, Djordje Gligorijevic, Ivan Stojkovic, Xiao Bai, Amit Goyal, and Zoran Obradovic. 2018. Deeply Supervised Semantic Model for Click-Through Rate Prediction in Sponsored Search. *CoRR* abs/1803.10739 (2018).

[42] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-Scale Bayesian Click-through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. In *ICML*. 13–20.

[43] Cheng Guo and Felix Berkhahn. 2016. Entity Embeddings of Categorical Variables. *CoRR* abs/1604.06737 (2016).

[44] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 1725–1731.

[45] H. Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, X. He, and Zhenhua Dong. 2018. DeepFM: An End-to-End Wide & Deep Learning Framework for CTR Prediction. *CoRR* abs/1804.04950 (2018).

[46] Rajan T. Gupta and Saibal K. Pal. 2019. Click-Through Rate Estimation Using CHAID Classification Tree Model. In *Advances in Analytics and Applications*. 45–58.

[47] Dustin H., Stefan S., Eren M., Hema R., and Chirs L. 2010. Improving ad relevance in sponsored search. In *WSDM*. 361–370.

[48] Xinran He, Stuart Bowers, Joaquin Quiñonero Candela, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, and Ralf Herbrich. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *ADKDD*. 1–9.

[49] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *SIGIR*. 355–364.

[50] X. He, K. Deng, X. Wang, Y. Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *SIGIR* (2020), 639–648.

[51] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.

[52] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. *ICDM* (2008), 263–272.

[53] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *RecSys*. 169–177.

[54] Dietmar Jannach, Gabriel de Souza P. Moreira, and Even Oldridge. 2020. Why Are Deep Learning Models Not Consistently Winning Recommender Systems Competitions Yet? A Position Paper. In *RecSys*. 44–49.

[55] Zilong Jiang, Shu Gao, and Wei Dai. 2016. Research on CTR Prediction for Contextual Advertising Based on Deep Architecture Model. *Control Engineering and Applied Informatics* 18 (03 2016), 11–19.

[56] Zilong Jiang, S. X. Gao, and Mingjiang Li. 2018. An improved advertising CTR prediction approach based on the fuzzy deep neural network. In *PloS one*.

[57] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware Factorization Machines in a Real-world Online Advertising System. *CoRR* abs/1701.04099 (2017).

[58] Shubhra Karmaker, Parikshit Sondhi, and ChengXiang Zhai. 2017. On Application of Learning to Rank for E-Commerce Search. *Proc. of the 40th ACM SIGIR Conference* (2017).

[59] K-M Kim, D. Kwak, H. Kwak, Y-J Park, S. Sim, J-H Cho, M. Kim, J. Kwon, Nako Sung, and J-W Ha. 2019. Tripartite Heterogeneous Graph Propagation for Large-scale Social Recommendation. In *RecSys*. 56–60.

[60] Michael A. King, Alan S. Abrahams, and Cliff T. Ragsdale. 2015. Ensemble learning methods for pay-per-click campaign management. *Expert Syst. Appl.* 42 (2015), 4818–4829.

[61] Y. Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *KDD*. 426–434.

[62] N. Kota and D. Agarwal. 2011. Temporal multi-hierarchy smoothing for estimating rates of rare events. In *KDD*. 1361–1369.

[63] S. Krishnan and R. Sitaraman. 2013. Understanding the effectiveness of video ads: a measurement study. In *IMC*. 149–162.

[64] S. Ktena, A. Tejani, L. Theis, P. Myana, D. Dilipkumar, F. Huszar, S. Yoo, and W. Shi. 2019. Addressing delayed feedback for continuous training with neural networks in CTR prediction. *CoRR* abs/1907.06558 (2019).

[65] Ashish Kumar and Jari Salo. 2016. Effects of link placements in email newsletters on their click-through rate. *Journal of Marketing Communications* 24, 5 (Mar 2016), 535–548.

[66] Rohan Kumar, Mohit Kumar, Neil Shah, and Christos Faloutsos. 2018. Did We Get It Right? Predicting Query Performance in e-Commerce Search. *CoRR* abs/1808.00239 (2018).

[67] Mounia Lalmas, Janette Lehmann, Guy Shaked, Fabrizio Silvestri, and Gabriele Tolomei. 2015. Promoting Positive Post-Click Experience for In-Stream Yahoo Gemini Users. In *KDD*. 1929–1938.

[68] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past erformance data. In *KDD*. 768–776.

[69] S. Leong, M. Mahdian, and S. Vassilvitskii. 2014. Advertising in a Stream. In *Proc. of WWW Conf.*

[70] Cheng Li, Yue Lu, Qiaozhu Mei, Dong Wang, and Sandeep Pandey. 2015. Click-through Prediction for Advertising in Twitter Timeline. In *KDD*. 1959–1968.

[71] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph Intention Network for Click-through Rate Prediction in Sponsored Search. *ACM SIGIR* (2019), 961–964.

[72] Xiang Li, Chao Wang, Jiwei Tan, Xiaoyi Zeng, Dan Ou, and Bo Zheng. 2020. Adversarial Multimodal Representation Learning for Click-Through Rate Prediction. *CoRR* abs/2003.07162 (2020).

[73] Zeyu Li, Wei Cheng, Yang Chen, Haifeng Chen, and Wei Wang. 2020. Interpretable Click-Through Rate Prediction through Hierarchical Attention. *WSDM* (2020), 313–321.

[74] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-GNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction. In *CIKM*. 539–548.

[75] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. *KDD* (2018).

[76] X. Lin, H. Chen, C. Pei, F. Sun, X. Xiao, H. Sun, Y. Zhang, W. Ou, and P. Jiang. 2019. A Pareto-Efficient Algorithm for Multiple Objective Optimization in e-Commerce Recommendation. In *RecSys*. 20–28.

[77] X. Ling, W. Deng, C. Gu, and et al. 2017. Model Ensemble for Click Prediction in Bing Search Ads. In *WWW*. 689–698.

[78] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. 2019. Feature Generation by Convolutional Neural Network for Click-Through Rate Prediction. In *WWW*. 1119–1129.

[79] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. 2020. *AutoGroup: Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction.*

[80] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, X. He, Z. Li, and Y. Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction. In *KDD*. 2636–2645.

[81] Hui Liu, Xingquan Zhu, Kristopher Kalish, and Jeremy Kayne. 2017. ULTR-CTR: Fast Page Grouping Using URL Truncation for Real-Time Click Through Rate Estimation. In *IEEE International Conference on Information Reuse and Integration.*

[82] Qiang Liu, Shu Wu, and Liang Wang. 2015. Collaborative Prediction for Multi-entity Interaction With Hierarchical Representation. In *CIKM*. 613–622.

[83] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. 2015. A Convolutional Click Prediction Model. In *CIKM*. 1743–1746.

[84] Xun Liu, Wei Xue, Lei Xiao, and Bo Zhang. 2017. PBODL : Parallel Bayesian Online Deep Learning for Click-Through Rate Prediction in Tencent Advertising System. *CoRR* abs/1707.00802 (2017).

[85] Yozen Liu, Xiaolin Shi, Lucas Pierce, and Xiang Ren. 2019. Characterizing and Forecasting User Engagement with In-App Action Graph: A Case Study of Snapchat. *KDD* (2019), 2023–2031.

[86] Zhe Liu, Xianzhi Wang, Lina Yao, Jake An, Lei Bai, and Ee-Peng Lim. 2020. Face to Purchase: Predicting Consumer Choices with Structured Facial and Behavioral Traits Embedding. *CoRR* abs/2007.06842 (2020).

[87] Amit Livne, Roy Dor, Eyal Mazuz, Tamar Didi, Bracha Shapira, and Lior Rokach. 2020. Iterative Boosting Deep Neural Networks for Predicting Click-Through Rate. *CoRR* abs/2007.13087 (2020).

[88] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *SIGIR*. 1137–1140.

[89] Miriam Marciel, Rubén Cuevas, Albert Banchs, Roberto González, Stefano Traverso, Mohamed Ahmed, and Arturo Azcorra. 2016. Understanding the Detection of View Fraud in Video Content Portals. In *WWW*. 357–368.

[90] P. Mattson, C. Cheng, C. Coleman, G. Diamos, and et al. 2019. MLPerf Training Benchmark. *CoRR* abs/1910.01500.

[91] Stephen McCreery and Dean M. Krugman. 2017. Tablets and TV Advertising: Understanding the Viewing Experience. *Journal of Current Issues & Research in Advertising* 38, 2 (Mar 2017), 197–211.

[92] B. McMahan, H. Holt, D. Sculley, and et al. 2013. Ad Click Prediction: A View from the Trenches. In *KDD*. 1222–1230.

[93] Tao Mei, Xian-Sheng Hua, Linjun Yang, and Shipeng Li. 2007. VideoSense: towards effective online video advertising. In *the 15th Intl Conference on Multimedia*. 1075–1084.

[94] Wei Meng, Xinyu Xing, Anmol Sheth, Udi Weinsberg, and Wenke Lee. 2014. Your Online Interests: Pwned! A Pollution Attack Against Targeted Advertising. In *CCS*. 129–140.

[95] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. 2011. Response prediction using collaborative filtering with hierarchies and side-information. In *KDD*. 141–149.

[96] M. Naumov, D. Mudigere, H. Michael Shi, and et. al. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. *CoRR* abs/1906.00091 (2019).

[97] Chenglei Niu, Guojing Zhong, Y. Liu, Yandong Zhang, Y. Sun, Ailong He, and Zhaoji Chen. 2018. Unstructured Semantic Model supported Deep Neural Network for Click-Through Rate Prediction. *CoRR* abs/1812.01353 (2018).

[98] R. Oentaryo, E. Lim, M. Finegold, and et al. 2014. Detecting click fraud in online advertising: a data mining approach. *J. Mach. Learn. Res.* 15 (2014), 99–140.

[99] R. Oentaryo, E. Lim, J. Low, D. Lo, and M. Finegold. 2014. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In *WSDM*. 123–132.

[100] Wentao Ouyang, Xiuwu Zhang, Li Li, Heng Zou, Xin Xing, Zhaojie Liu, and Yanlong Du. 2019. Deep Spatio-Temporal Neural Networks for Click-Through Rate Prediction. *KDD* (2019), 2078–2086.

[101] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Li Li, Zhaojie Liu, and Yanlong Du. 2019. Click-through rate prediction with the user memory network. *CoRR* abs/1907.04667 (2019).

[102] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Chao Qi, Zhaojie Liu, and Yanlong Du. 2019. Representation Learning-Assisted Click-Through Rate Prediction. In *IJCAI*.

[103] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. *SIGIR* (2019), 695–704.

[104] Junwei Pan, Yizhi Mao, Alfonso Lobos Ruiz, Yu Sun, and Aaron Flores. 2019. Predicting Different Types of Conversions with Multi-Task Learning in Online Advertising. *KDD '19* (2019), 2689–2697.

[105] Jing Pan, Weian Sheng, and Santanu Dey. 2019. Order Matters at Fanatics Recommending Sequentially Ordered Products by LSTM Embedded with Word2Vec. *CoRR* abs/1911.09818 (2019).

[106] Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. 2018. Field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising. In *WWW*. 1349–1357.

[107] Zhen Pan, Enhong Chen, Qi Liu, Tong Xu, Haiping Ma, and Hongjie Lin. 2016. Sparse Factorization Machines for Click-through Rate Prediction. 400–409. https://doi.org/10.1109/ICDM.2016.0051

[108] Changhua Pei, Xinru Yang, Qing Cui, Xiao Lin, Fei Sun, Peng Jiang, Wenwu Ou, and Yongfeng Zhang. 2019. Value-Aware Recommendation Based on Reinforcement Profit Maximization. In *WWW*. 3123–3129.

[109] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. *CoRR* abs/1403.6652 (2014).

[110] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on Long Sequential User Behavior Modeling for Click-Through Rate Prediction. *Proc. of the 25th ACM SIGKDD Conf.* (2019).

[111] M. Safaei Pour, A. Mangino, K. Friday, Matthias Rathbun, E. Bou-Harb, F. Iqbal, Kh. Shaban, and A. Erradi. 2019. Data-Driven Curation, Learning and Analysis for Inferring Evolving IoT Botnets in the Wild. In *ARES*. Article 6.

[112] S. Punjabi and P. Bhatt. 2018. Robust Factorization Machines for User Response Prediction. In *WWW*. 669–678.

[113] P. Qi, X. Zhu, G. Zhou, Y. Zhang, Z. Wang, L. Ren, Y. Fan, and K. Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. *CoRR* arXiv:2006.05639 (June 2020).

[114] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User Behavior Retrieval for Click-Through Rate Prediction. *CoRR* arXiv:2005.14171 (May 2020).

[115] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting Cross-Session Information for Session-Based Recommendation with Graph Neural Networks. *J. of ACM TIOS* 38 (2020).

[116] Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. *CoRR* abs/1908.04032 (2019).

[117] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-Based Neural Networks for User Response Prediction. *2016 IEEE 16th International Conference on Data Mining (ICDM)* (2016), 1149–1154.

[118] Yanru Qu, Bohui Fang, Wei-Nan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, yong Yu, and Xiuqiang He. 2018. Product-Based Neural Networks for User Response Prediction over Multi-Field Categorical Data. *J. of ACM TIOS* (2018).

[119] Regelson, Moira, Fain, and Daniel C. 2006. Predicting click-through rate using keyword clusters.

[120] K. Ren, J. Qin, Y. Fang, W. Zhang, L. Zheng, W. Bian, G. Zhou, J. Xu, Y. Yu, X. Zhu, and K. Gai. 2019. Lifelong Sequential Modeling with Personalized Memorization for User Response Prediction. In *SIGIR'19*.

[121] Kan Ren, Weinan Zhang, Yifei Rong, Haifeng Zhang, Yong Yu, and Jun Wang. 2016. User Response Learning for Directly Optimizing Campaign Performance in Display Advertising. In *CIKM*. 679–688.

[122] Steffen Rendle. 2010. Factorization Machines. *2010 IEEE International Conference on Data Mining* (2010), 995–1000.

[123] S. Rendle, L. Zhang, and Y. Koren. 2019. On the Difficulty of Evaluating Baselines: A Study on Recommender Systems. *CoRR* abs/1905.01395 (2019).

[124] Jenna Reps, Uwe Aickelin, Jonathan Garibaldi, and Chris Damski. 2014. Personalising Mobile Advertising Based on Users' Installed Apps. *ICDM Workshop*, 338–345.

[125] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *WWW*. 521–530.

[126] Qiu Ruihong, Yin Hongzhi, Huang Zi, and Tong Chen. 2020. GAG: Global Attributed Graph Neural Network for Streaming Session-based Recommendation. *CoRR* abs/2007.02747 (2020).

[127] Oliver Rutz, Ashwin Aravindakshan, and Olivier Rubel. 2019. Measuring and forecasting mobile game app engagement. *International Journal of Research in Marketing* 36, 2 (Jun 2019), 185–199.

[128] Oliver J. Rutz and Randolph E. Bucklin. 2011. From Generic to Branded: A Model of Spillover in Paid Search Advertising. *J. of Marketing Research* (2011), 87–102.

[129] Rubén Saborido, Foutse Khomh, Giuliano Antoniol, and Yann-Gaël Guéhéneuc. 2017. Comprehension of Ads-Supported and Paid Android Applications: Are They Different? *2017 IEEE/ACM (ICPC)* (2017), 143–153.

[130] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. In *KDD*. 255–262.

[131] Neha Sharma and Nirmal Gaud. 2015. K-modes Clustering Algorithm for Categorical Data. *International Journal of Computer Applications* 127 (2015), 1–6.

[132] Weichen Shen. 2018. Easy-to-use,Modular and Extendible package of deep-learning based CTR models. https://github.com/shenweichen/DeepCTR.

[133] Weichen Shen. 2019. (PyTorch) Easy-to-use,Modular and Extendible package of deep-learning based CTR models. https://github.com/shenweichen/DeepCTR-Torch.

[134] Shu-Ting Shi, Wenhao Zheng, Jun Tang, Qing-Guo Chen, Yao Hu, Jianke Zhu, and Ming Li. 2020. Deep Time-Stream Framework for Click-Through Rate Prediction by Tracking Interest Evolution. *CoRR* abs/2001.03025 (2020).

[135] Enno Shioji and Masayuki Arai. 2017. Neural Feature Embedding for User Response Prediction in Real-Time Bidding (RTB). *CoRR* abs/1702.00855 (2017).

[136] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction. *CoRR* abs/2007.06434 (2020).

[137] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. *CoRR* abs/1810.11921 (2019).

[138] Yuhan Su, Zhongming Jin, Ying Chen, Xinghai Sun, Yaming Yang, Fangzheng Qiao, Fen Xia, and Wei Xu. 2017. Improving click-through rate prediction accuracy in online advertising by transfer learning. In *Intl. conference on WI*. 1018–1025.

[139] Anh-Phuong Ta. 2015. Factorization machines with follow-the-regularized-leader for CTR prediction in display advertising. *IEEE Big Data* (2015), 2889–2891.

[140] G. S. Thejas, Kianoosh G. Boroojeni, Kshitij Chandna, Isha Bhatia, S. S. Iyengar, and N. R. Sunitha. 2019. Deep Learning-based Model to Fight Against Ad Click Fraud. In *ACM SE*. 176–181.

[141] T. Tian, J. Zhu, F. Xia, X. Zhuang, and T. Zhang. 2015. Crowd Fraud Detection in Internet Advertising. In *WWW*. 1100–1110.

[142] Gabriele Tolomei, Mounia Lalmas, Ayman Farahat, and Andrew Haines. 2018. You must have clicked on this ad by mistake! Data-driven identification of accidental clicks on mobile ads with applications to advertiser cost discounting and click-through rate prediction. *Springer - J. of Data Science and Analytics* 7 (2018), 53–66.

[143] Sergio Duarte Torres, Ingmar Weber, and Djoerd Hiemstra. 2014. Analysis of Search and Browsing Behavior of Young Users on the Web. *J. of ACM TWEB* 8 (2014), 7:1–7:54.

[144] Vinh Truong, Mathews Nkhoma, and Wanni Pansuwong. 2019. An Integrated Effectiveness Framework of Mobile In-App Advertising. *Australasian J. of Inf. Systems* 23 (2019).

[145] Flavian Vasile, Damien Lefortier, and Olivier Chapelle. 2017. Cost-sensitive Learning for Utility Optimization in Online Advertising Auctions. 1–6. https://doi.org/10.1145/3124749.3124751

[146] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. *CIKM '18* (2018), 417–426.

[147] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. *CoRR* abs/1801.08284 (2018).

[148] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. *KDD* (2019), 968–977.

[149] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *WWW*. 2000–2010.

[150] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba. *KDD Intl. Conf.* (2018), 839–848.

[151] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *Found. Trends Inf. Retr.* 11 (2016), 297–435.

[152] Qianqian Wang, Fang'ai Liu, Shuning Xing, and Xiaohui Zhao. 2018. A New Approach for Advertising CTR Prediction Based on Deep Neural Network via Attention Mechanism. In *Comp. Math. Methods in Medicine*.

[153] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29 (2017), 2724–2743.

[154] Ruoxi Wang, Bin Fu, Gang Fu, et al. 2017. Deep & Cross Network for Ad Click Predictions. *CoRR* abs/1708.05123 (2017).

[155] Ruoxi Wang, Rakesh Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and Ed Huai hsin Chi. 2020. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. *CoRR* (2020).

[156] X. Wang, X. He, M. Wang, F. Feng, and T. Chua. 2019. Neural Graph Collaborative Filtering. *CoRR* abs/1905.08108 (2019).

[157] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, P. Yu, and Yanfang Ye. 2019. Heterogeneous Graph Attention Network. *CoRR* abs/1903.07293 (2019).

[158] X. Wang, W. Li, Y. Cui, R. Zhang, and J. Mao. 2011. Click-Through Rate Estimation for Rare Events in Online Advertising.

[159] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. 2018. Multi-Level Deep Cascade Trees for Conversion Rate Prediction. *CoRR* abs/1805.09484 (2018).

[160] Hong Wen, Jing Zhang, Yuan Wang, Wentian Bao, Quan Lin, and Keping Yang. 2019. Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction. *CoRR* abs/1910.07099 (2019).

[161] Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, and Nitesh V. Chawla. 2019. Neural Tensor Factorization for Temporal Interaction Learning. In *WSDM*. 537–545.

[162] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. *CoRR* abs/1708.04617 (2017).

[163] Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. 2020. Privileged Features Distillation at Taobao Recommendations. *CoRR: Information Retrieval* (2020).

[164] C. Xu and M. Wu. 2020. Learning Feature Interactions with Lorentzian Factorization Machine. In *AAAI*. 6470–6477.

[165] Hongxia Yang, Quan Lu, Angus Xianen Qiu, and Chun Han. 2016. Large Scale CVR Prediction through Dynamic Transfer Learning of Global and Local Features, Vol. 53. PMLR, 103–119.

[166] Xiao Yang, Tao Deng, Weihan Tan, Xutian Tao, Junwei Zhang, Shouke Qin, and Zongyao Ding. 2019. Learning Compositional, Visual and Relational Representations for CTR Prediction in Sponsored Search. In *CIKM*. 2851–2859.

[167] Y. Yang, X. Guan, and J. You. 2002. CLOPE: a fast and effective clustering algorithm for transactional data. In *KDD*.

[168] Yi Yang, Baile Xu, Furao Shen, and Jian Zhao. 2019. Operation-aware Neural Networks for User Response Prediction. *J. of Elsevier Neural networks* 121 (2019), 161–168.

[169] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *CoRR* abs/1806.01973 (2018).

[170] Xu Yong, Chen Jiahui, Huang Chao, Zhang Bo, Xing Hao, Dai Peng, and Bo Liefeng. 2020. Joint Modeling of Local and Global Behavior Dynamics for Session-Based Recommendation*. *J. of FAIA* 325, ECAI 2020 (2020), 545–552.

[171] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation. In *SIGIR*. 1469–1478.

[172] Yuan Yuan, Xiaojing Dong, Chen Dong, Yiwen Sun, Zhenyu Yan, and Abhishek Pani. 2018. Dynamic Hierarchical Empirical Bayes: A Predictive Model Applied to Online Advertising. *CoRR* abs/1809.02213 (2018).

[173] Y. Yuan, F. Wang, J. Li, and R. Qin. 2014. A survey on real time bidding advertising. In *IEEE SOLI*. 418–423.

[174] C. Zhang, D. Song, C. Huang, A. Swami, and N. Chawla. 2019. Heterogeneous Graph Neural Network. In *KDD*. 793–803.

[175] Chuxu Zhang, A. Swami, and Nitesh V. Chawla. 2019. SHNE: Representation Learning for Semantic-Associated Heterogeneous Networks. *WSDM* (2019).

[176] J. Zhang, T. Huang, and Z. Zhang. 2019. FAT-DeepFFM: Field Attentive Deep Field-aware Factorization Machine. In *ICDM*.

[177] Li Zhang, Weichen Shen, Shijian Li, and Gang Pan. 2019. Field-Aware Neural Factorization Machine for Click-Through Rate Prediction. *IEEE Access* 7 (2019), 75032–75040.

[178] Weinan Zhang, Lingxi Chen, and Jun Wang. 2016. Implicit Look-Alike Modelling in Display Ads - Transfer Collaborative Filtering to CTR Estimation. *CoRR* abs/1601.02377 (2016).

[179] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep Learning over Multi-field Categorical Data - - A Case Study on User Response Prediction. In *ECIR*.

[180] Weinan Zhang, Tianxiong Zhou, Jun Wang, and Jian Xu. 2016. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. In *KDD*. 665–674.

[181] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *AAAI*. 1369–1375.

[182] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System? A Sequential Meta-Learning Method. In *SIGIR*. 1479–1488.

[183] Y. Zhang, P. Zhao, Y. Guan, L. Chen, K. Bian, L. Song, B. Cui, and X. Li. 2020. Preference-Aware Mask for Session-Based Recommendation with Bidirectional Transformer. In *ICASSP*. 3412–3416.

[184] Weijie Zhao, Deping Xie, Ronglai Jia, Yulei Qian, Ruiquan Ding, Mingming Sun, and Ping Li. 2020. Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems. *CoRR* abs/2003.05622 (2020).

[185] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. 2020. Jointly Learning to Recommend and Advertise. *CoRR* abs/2003.00097 (2020).

[186] Yifei Zhao, Yu-Hang Zhou, Mingdong Ou, Huan Xu, and Nan Li. 2020. Maximizing Cumulative User Engagement in Sequential Recommendation: An Online Optimization Perspective. In *KDD*. 2784–2792.

[187] Hua Zheng, Dong Wang, Qi Zhang, Hang Li, and Tinghao Yang. 2010. Do Clicks Measure Recommendation Relevancy? An Empirical User Study. In *RecSys*. 249–252.

[188] Guorui Zhou, Ying Fan, Runpeng Cui, Weijie Bian, Xiaoqiang Zhu, and Kun Gai. 2018. Rocket Launching: A Universal and Efficient Framework for Training Well-performing Light Net. *CoRR* abs/1708.04106 (2018).

[189] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. *CoRR* abs/1809.03672 (2019).

[190] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. *KDD* (2018), 1059–1068.

[191] Guorui Zhou, Kailun Wu, Weijie Bian, Zhao Yang, Xiaoqiang Zhu, and Kun Gai. 2019. Res-embedding for deep learning based click-through rate prediction modeling. *CoRR* abs/1906.10304 (2019).

[192] K. Zhou, M. Redi, A. Haines, and et al. 2016. Predicting Pre-click Quality for Native Advertisements. In *WWW*. 299–310.

[193] Wen-Yuan Zhu, Chun-Hao Wang, Wen-Yueh Shih, W. Peng, and J. Huang. 2017. SEM: A Softmax-based Ensemble Model for CTR estimation in Real-Time Bidding advertising. *IEEE J. of BigComp.* (2017), 5–12.

[194] Xingquan Zhu and Ian Davidson. 2007. *Knowledge Discovery and Data Mining: Challenges and Realities*. IGI Global.

[195] X. Zhu, H. Tao, Z. Wu, J. Cao, K. Kalish, and J. Kayne. 2017. *Fraud Prevention in Online Digital Advertising*. Springer.

[196] B. Zoph and Q. Le. 2016. Neural Architecture Search with Reinforcement Learning. *CoRR* abs/1611.01578 (2016).

A APPENDIX

A.1 Term Definitions

In Table A.1, we introduce some important terms used in the context of online advertising, and also summarize common terminologies used in the paper.

Table A.1. Summary of key terms and notations in online advertising

Notation	Descriptions
Display Ads	The form of ads containing rich media content that are displayed in reserved spaces in websites
Search Ads	The form of ads that are displayed in the search result page triggered based on user search query. They are sorted and displayed to users to attract their attention to commercial products.
Native Ads	The form of ads which are similar to regular posts in stream media platforms including video and images commercial contents.
Advertiser	The stakeholders which promote products and services in online advertising, by serving ads to online users
Publisher	The stakeholders which run websites with potential placements to show ads to online ads
Audience	The online users who are exposed to ads in a respective online context
Ad Exchange	The marketplace where advertisers and publishers are connected, through DSP and SSP, to negotiate selling and buying price using real-time bidding (auctions)
Auction	A virtual real-time sale being held upon SSP ad request to gather bids of advertisers for an ad impression.
Demand Side Platform (DSP)	A software platform in online advertising eco-system working on the behalf of advertisers to manage campaigns and submit a relevant bid price to bid requests
Supplier Side Platform (SSP)	A software platform in online advertising working on the behalf of publishers to send a bid request to Ad exchange network. It loads the ad creative by calling ad server which has a winning ad in an auction
Ad Campaign	A set of advertisements with a common theme (objective) in targeting similar group of users
Ad Creative	The message or artwork (advertising object), designed by advertisers, to be served to the audience' devices
Ad Placement	It refers to the location where an ad is displayed on the web-page. The common places that are available for advertiser to run an ads are the footer, the header and sidebars of website or anywhere in the beginning or middle of article and videos.
Banner Ad	A rectangular graphic window, such as an iFrame, dedicated to show ad creative (image or text content) in the publisher's web-page
Impression	The rendering/presence of an ad on the user's device
Landing page	A web-page used to show to viewers, after they click on the ad
Click	A user mouse click event or user tap event on the ad when they visiting an ad on desktop or mobile devices
Conversion	The user actions, such as purchasing a product or subscribing to a service, after clicking an ad creative and being directed to advertiser's landing web-pages
Gross Merchandise Volume	A e-commerce metric considered by businesses to indicate the amount of sales made by users over a specific period before deducting any expenses like those associated with online marketing
Data Management Platform (DMP)	A software platform in online advertising designed to collect and analyze data for both advertisers and publishers. DMPs provide services to DSPs or SSPs to improve ad campaign efficiency
First party Data	First party is referred to as the stakeholder itself, so first party data is referred to as data collected from the activities of business users of respective stakeholders. The data ranges from user profile information like demography user historical behaviors such as visited pages and purchase history, user subscription data or their activities in social media
Second party Data	Second party is referred to as the other party of each stakeholder. The first party data of a company is referred to as the second party data of other companies
Third party Data	Data gathered from outside sources to be packaged and sold to others. The data is organized to clusters and segments in terms of page information, user characteristics, and audience interest to be chosen by buyers. Third party data are typically gathered from DMP by analyzing user cookie information

B MEDIA TYPES, DEVICES, AND PLATFORMS

Driven by the communication and networking technologies, online advertising has been continuously evolving in the past two decades. Starting from static banners on websites, the industry is now dynamically serving ads based on types of media platforms, user devices, and media types [151]. In this section, we briefly summarize types of media platforms, user devices, and ad types in the online advertising eco-system.

B.1 Media Platforms

There are various advertising platforms that have been proposed to serve advertisements to users, depending on the context of users accessing to the network.

B.1.1 Sponsored Search Marketing. Sponsored search is a search engine based advertising platform that uses user query as the context and returns a list of ads related to user queries for advertising. The

result of ads are usually displayed as a sorted list of commercial hyperlinks among the search result page on the basis of similarity to the search result. To differentiate ads from regular list of search results, an "Ad" tag is usually marked to promoted records. Following the pay-per-click mechanism, the advertisers pay the search engine platform when user click on ads. Then the potential revenue for publishers is calculated using click-through rate values.

B.1.2 Display Advertising. Display advertising is one of latest forms of online advertising working on the basis of real-time bidding to select and show personalized ads in the dedicated areas of web pages. Different from search advertising where user queries provide clear context of user preference, the context of display advertising is often limited to the pages visited/requested by users. Given a user requesting to visit a URL in the web browser, ads are delivered from real-time auctions set up by ad exchange network. The main approach in this advertising is targeting individuals using information available from them in different internet sources (like cookies for browsing history). The main goal for advertisers is to find strategies to reach to right customers to engage them to take desirable action. The performance of marketing is evaluated based on user responses like click or conversion which can determine display-related advertising revenues and cost for publishers and advertisers.

B.1.3 Social Media Advertising. Social media networks are becoming common elements of daily life, and providing an opportunity for people to interact and share information. The popularity of these platforms motivate businesses and companies to target their potential customers among online users. To this end, advertisers aim to provide personalized posts or tweets in social media platforms, like Twitter, Facebook, and Instagram, to attract user responses, which are usually measured through click-through rate or conversion rate. To address the quality of ads, some metrics like post-click-experience are also introduced to analyze the dwell time users may take on a landing page following the click event. Some studies [9, 67] show that post-click-experience can be used to gain additional knowledge about user preferences and modeling user responses.

B.2 Ad Types

In order to promote products and services in online advertising, different types of advertisements have been designed as the means of advertising.

B.2.1 Banner Ads. Banner ads, which have existed since the very beginning of online advertising in 1990s, incorporate the main standard media type in the Internet for advertising. Organized by compounds of text, image, or animated contents placed in the specific area of web pages, banner ads present the advertisers' message to users to attract their attention about the promoted content. Clicking on the areas generally indicate some sort of matching between ad content and user preferences. As a result, a click leads to a transition of users from the publisher web pages to advertisers' websites for marketing purposes.

B.2.2 Textual Ads. Text ads are the most well-known type of ads in many advertising platforms like sponsored search advertising, short message (SMS) marketing, email advertising, or display advertising [55]. A text ad includes a textual creative ad shown alongside search results [32] or part of the emails or text message sent to subscribed users to show promotional messages. The elements of ads are organized to lead to a click response directing users to promoted pages [65].

B.2.3 Video Ads. With the increase of broadband Internet, the usage of video ads to deliver promotional content has gained increasing popularity and becoming an effective way of interact with audience. Today the striking amount of video ads are employed to transfer commercial messages to online users. In the form of live videos or downloadable video content, this type of advertisements can be presented to users like banners in websites, tweets, or feeds in social media platforms.

Roll-out Ads. Roll-out ad is a specific form of video ads that are joined to the other video content in streaming websites like YouTube or Vudu. These videos automatically appear before playing the original content can be seen in either skippable or non-skippable formats. According to the place, the video ads being attached to the beginning or the end of original one are called pre-roll or post-roll advertisements. The mid-roll version of video ads is defined to augment to some points in the middle of original videos which is contextually relevant and less intrusive to users [93].

B.2.4 In-App Ads. This type of ads represents the version of advertisements which appear within applications like video games in desktop systems or smartphones. Like digital advertising, mobile apps are designed to reserve some space for ads. There are generally two ways used to display and place ads in apps. The first is considered as static brands involved in the background and integrated as the part of app. This type follows the guaranteed contract setting and similar to the primitive version of ads in the past which cannot directly receive user responses. Therefore, they can be considered superficial or even not recognizable by users [11]. The second type, which is considered more interactive ones, places ads in transitions. They usually appear in two forms of full screen or regular rich media ads. Following ad real-time bidding model, relevant ads are selected from negotiation between DSP and SSP to be displayed across the screen [144].

B.3 Source of Features

The source of features for user response prediction task is related to information transferred in online advertising ecosystem. In the ad network workflow in Figure 1, actors, such as advertiser and publisher, play a role together with intermediary nodes to provide users with relevant commercial content. In order to acquire positive user feedback, various data features are used to represent users and describe advertisers and online content providers. The data sources for feature extraction are information from demand side or supplier side, or even external third party sources. The representative list of features regarding supplier-side platform, demand-side platform and third-party groups are shown in Table B.1.

B.3.1 Suppler Side Features. As showing in Figure 1, SSP or supplier side platforms are intermediary nodes in the ad network which work on behalf of publisher to manage the inventory of available ad placements in web pages. As soon as users submitting a keyword in search engine or visiting a website in display advertising, an auction is triggered to find ad to be served to the users. In this case, information regarding visiting user and ad slots are transferred to relevant DSP nodes through ad exchange network. As shown in Table B.1, this information is characterized via features describing ad creatives and their appearance in the publisher web-pages, such as features about the content of web-pages, placement id, size, width and height, visibility status and format, as well as user information such as device types, user agent, browser information *etc.*

Source	Features
Supplier Side Platform	Page URL, Device type, Devide Id, HTTP cookie, OS Version, Browser type/Version, User agent, Geo location, Ad slot (ID, width, height, visibility, format), placement ID, Publisher ID, User ID
Demand-Side Platform	Bidding price, Paying price, Campaign Category, creative ID, Advertiser ID
Third-party	User segmentation, User demography, Site information, Page information, Page categorization

Table B.1. A summary of major sources of categorical features in online advertising

B.3.2 Demand Side Features. Another important party in the online advertising ecosystem is DSP (Demand side platform) which intervenes in connections between advertisers and ad exchange network. An ad exchange casts auction for bid request triggered by SSPs to DSPs to select the display of ad on the publisher's website.The Ad exchange network collects bid prices offered by connected advertisers through their DSPs, and selects ads corresponding to higher bid to present in publisher websites. The bid price calculated by advertisers depending on three set of main factors such as information available

about users, the constraints of publisher pages, and ad campaigns. The information is characterized using features such as ID and the floor price of ad creatives, the URL of landing page, the segmented user profiles, and ad campaign categories to describe the campaign and content of ads. They are augmented with the online user browsing history captured by DMP(Data Management Platform) nodes to facilitate decision making to choose matching ads for users.

B.3.3 Third-Party Side Features. Features used to select the best matching ads to users' interests are not only limited to those from by DSPs and SSPs. Instead, external sources can also provide compliment and valuable information needed to describe objects involved in advertising scenarios. In this context, they are known as third party data which are captured in various websites and social media platforms, using data aggregation, cookie, or machine learning approaches. These data can include different information exploited ranging from user web cookies, to meta-data of devices and geographic information. For example, White Ops, an online ad verification and fraud detection company, provides third-party side services allowing users to query whether the traffic (*i.e.* the page visit) is initialized from a genuine human user or a bot in real-time, using data collected from "trillions of transactions". Such services allow an advertiser to determine whether an auction is potentially fraudulent and stop bidding on fraud auctions [195]. Online advertising systems can leverage third party data to organize new features used to target users for their campaigns.

B.4 Device Platforms

B.4.1 Desktop Advertising. Since the very beginning of banner ads on the web, desktop was and still is the dominate device platform for advertising. Available through desktop systems, desktop advertising entails expanded version of ads including text-based advertisement, roll-out video ads, and in-app advertisements appearing in search engine results, streaming web services or software. The capabilities of smart phones make these devices as the predominant opponent of desktop systems since it can be used for same purposes. However, it does not discount the value of desktop branding as long as the desktop and laptop systems stays on.

B.4.2 Mobile Advertising. As smartphones and mobile devices are becoming essential tools for communication, online advertising also quickly adapts to mobile devices for marketing. In early days of mobile phones, the common advertising form was SMS advertising in which advertisers send the textual ads to customers. The rise in popularity of multi-purpose mobile phones and wearable devices made an opportunity for online companies to use a new way for advertising and targeting audience. Nowadays, mobile advertising makes up a significant portion of online advertising [99, 142] which can be roughly classified into the following two types:

Mobile Web Advertising. Like desktop workstations, one of the major advantages of smartphones is web browsing, which relies on search engines to get relevant information for user needs. There is no wonder that users take advantage of their phones to look for services and products located near them. In era of virtual assistants, voice search has seen significant growth using smartphones. Now, mobile web advertising is being developed to leverage new ways, like natural language based optimization, to provide sponsored search for verbal personalized ads for users. Using cross-platform compatibility followed in mobile devices make it possible for advertisers to focus on the content of their ads for potential customers with low cost about the ways that are published in different devices [112]. But advertising is not limited to voice data. The transcript of verbal or written conversations between customer-service agents and people consist of valuable information about users implicit and explicit behavior which can be analyzed to predict user life events and campaign relevant products [31].

Mobile App Advertising. With the development of smartphone devices, there is an continually increase in the production of mobile applications. The bulk of research confirmed the fact that users today spend more time in online applications than web browsing on cellphones [3]. Mobile app

advertising, known as in-app advertising, is the specific type of advertising where ads are served within smartphone app eco-system. In this way, there is a trend that smartphone app developers prefer more to produce free applications by supporting an ad model to gain revenue from presenting ads on their apps [129]. This type of advertising deals with different advertisement units like banners ads which are shown at top or bottom of apps and transitional apps that are places for transitions inside applications.

B.4.3   Tablet Advertising. Tablet advertising is designed to present relevant ads for tablet users. Although both tablet and smartphones are essentially mobile devices, they are quite different in terms of quality of display and goals of usages. Some studies highlighted the difference in user behavior regarding advertisement when using tablet devices [91]. The recent years have seen a decline in using tablets since brands continue to create bigger screen smartphones. But there are still many people use this device for entertainment activities[4] like playing online video games and second screen viewing[5] to watch the main media stream or live event and use tablets at the same time. Business users also use it to complement office activities. It can lead to generate plenty of opportunities to design distinctive tablet-based business-themed approaches for ad campaigns. In these devices, advertisements take advantage of high resolutions banners and videos to provide a better user experience and grab more user engagement.

C   EVALUATION PROCEDURE OF USER RESPONSE PREDICTION MODELS

The evaluation of user response prediction can be categorized into offline (simulation based) evaluation and online test, which are from academia and industry perspectives, respectively.

C.1   Offline Evaluation

Majority papers published in this domain are developed by researchers from academia. Common approaches used by them to assess model performance are to use a simulation of the real-world environment. In this condition, sample data from different stakeholders in recommender and online advertising systems are gathered to follow offline trials. As shown in Figure 2, the studies conducted about recommendation systems and online advertising typically aim to provide two types of outputs corresponding to different stages in the system. These studies are categorized into two type of point-wise and list-wise methods. In the former, the models predict the single numeric estimated value of user feedback score such as product rates, click-through rate, or conversion rate values. The latter prepares the ranked list of products ordered by predicted user interaction scores. In point-wise models, models are evaluated using different metrics such as root mean square error for regression problems (product rating), Area over ROC curve (AUC) and Accuracy metrics for binary classification methods i.e. CTR prediction, CVR (purchase) prediction. The output in list-wise scenarios are optimized via ranking metrics like Precision (Precision@K) and Recall (Recall@K) over top k recommended list. The performance of these systems are also evaluated from other aspects such as Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR). A common approach to evaluate the quality of a recommended item list with k elements is evaluating the predicted score of user interaction for test samples located among random selected set of unvisited items[51]. Table D.2 compares the performance of different methods for a binary classification task to predict click events. Although the success of model designed based on deep learning make it cornerstone to develop models in academia and industry for user response prediction, it cannot be neglected that in some scenarios like recommendation system contests (RecSys conference) [54] [34] which evaluate the methodologies for offline experiments and indicate that winning solutions could be selected from the old techniques like SVMs, KNN, Logisitic Regression and Decision using ensemble based models. Preparing proper

---

[4]http://googlemobileads.blogspot.com/2011/11/consumers-on-tablet-devices-having-fun.html
[5]https://www.statista.com/topics/2531/second-screen-usage/

environments to assess baselines is also important [123], and the research has shown that early methods could outperform recent proposed algorithms as long as they are well set up for experiments. These results provide some indications about the evaluation procedures. It first can show the significance of feature engineering and knowledge about of the application domain to use appropriate features in modeling. It also raises the reliability issue in experiments. Although many reported results are based on cross-validation, statistical significance tests and availability of their code for reproducibility (Some of which are gathered in Table D.3), several arbitrariness in experiment designs should be addressed by research community. In the following sub-section, we discuss evaluation methodologies followed in the reviewed papers covering common approaches to set up experiments.

C.1.1 Experimental Setup. The evaluation step in prediction and classification tasks aim to assess how well model can cope with new unseen samples. Typical data splitting is performed by randomly selecting samples without replacement from datasets to create three partitions: training, validation, and test. However, in recommendation and online advertising systems, the dataset include data logs of user interactions with online systems in which each sample come with a timestamp. So considering the sequence of samples in term of time and applying a chronological order constraint in the data splitting in datasets is a rational expectation. The data-splitting in this case is followed by choosing a arbitrary cut-off point in the dataset to prepare training and test subsets. A typical approach is leave-one-out method which assigns the latest data to the test-set while the reminder data are dedicated for training and validation sets [51]. To address short-term and long-term sequential data, there are two form of event-based and session based datasets. In the event-based dataset, the common idea is selecting randomly a couple of time intervals where samples before the split point are assigned for training set whereas the one after splitting time point are for test set (ex. in a dataset including 7 subsequent days, the first subsequent days are assigned to training while the last day data is for test set [44, 117]). In the session-based datasets, the same procedure is applied with a difference that the samples before or after the split point include the session of events. There are different approaches to define the session notion to represent short-term user interaction logs. In [36], the cut-off to split data logs into session is determined as the gap of 30 minutes between subsequent interactions. Section D.1 summarizes a number of benchmark datasets presented in different existing academic and industrial experiments.

C.2 Online A/B Test

A/B test is an evaluation mechanism which provides controlled environment to compare models by splitting user traffic into two different portions, A *vs.* B, to compare model performance. So models are assessed using real-world system users to drive desired user feedback signals. To avoid misleading evaluation results using the knowledge about the field, the statistical significance tools are applied to create variation of data, and make statistical test to evaluate confidence intervals. The rate of improvement compared to the baselines and rate of error are central in results based on A/B test. A common metric used to present online comparison is to calculate the relative improvement like Normalized Log Loss (NLL) metric as $\frac{LL(\overline{p})-LL(p)}{LL(\overline{p})}$ where $LL(\overline{p})$ is the log loss value of the best predictor on the test dataset while $LL(p)$ is for any baseline predictor p. Typically baseline models are built based on logistic Regression (LR) models that are highly engineered with rich features in production simulations [22, 45, 57, 118]. In recent developed models, the baseline models are chosen from successful models like DeepFM [44] or DIN[190] for online experiments [79, 80, 189]. However, previously, authors in [57] compare the performance of the proposed model, *i.e.* FFM with the baseline LR models via calculating Return on Investment (ROI) relative improvement in display advertising. Some other works like [45] calculate the relative improvement in recommendation metric values like Coverage, Popularity and Personalization to compare DeepFM model with a baseline LR model in Huawei app market environment. Because of complexities in online businesses, there is no

standardized plan to follow online experiments. So the number of research papers conducting this evaluation procedure are relatively limited in the literature.

D   APPLICATIONS, RESOURCES & FUTURE DIRECTIONS

For the online advertising, the click-through rate and conversion rate are common user response metrics to evaluate the marketing performance and determine the revenue of advertisers and publishers in online advertising and recommender systems. In this section, we review some applications and tasks related to user interactions in online content providers. We will introduce the benchmark user-log datasets and the publicly available source-codes provided by researchers. Such information is useful in a case we need to deal with baseline methods in the domain of user response prediction.

D.1   Benchmark datasets for the user response prediction

Benchmark datasets are helpful type of data sources to carry out a fair comparison between different methods. A list of publicly available datasets that were examined in previous studies are summarized in Table D.1. Several benchmarking datasets have been prepared and made publicly available to conduct studies for the user response prediction. Table D.2 demonstrates the performance of different user response prediction methods on selected datasets. We describe two representative datasets in the following.

Criteo Dataset. This is one of the important benchmark datasets gathered from the seven days real data logs by Criteo company. It was initially prepared for a competition held by Kaggle in 2014 to encourage the development of approaches for click-through rate prediction task. Criteo dataset includes 39 high cardinal features which are consisting of 26 categorical features and 13 continuous features for each pair of ad and user describing the event that ad visited by the user. Each row corresponds to one impression. Each instance of user and ad has a label to indicate whether the impressed ad receives ad click response or not. It have a full and condensed version containing around 500 and 45 million samples respectively.

Avazu Dataset. The second dataset prepared by Kaggle for a competition in 2014 including user click behavior gathered from Avazu mobile advertising platform. Each row in this dataset describes an impression (an ad displayed to users). Each impression event is attributed by a set of features such as categorical features for user, device and advertisement like hour of day, banner position, site id and device model. Avazu dataset contains around ten days click-through data of mobile ads following the chronological order. For experiments, two subsets of dataset including 24 fields from the first 9 days of logs for training and remaining for test and evaluations are also available.

D.2   Open-source Implementations

In this subsection, we collect a set of methods introduced in different published papers in one place. Table D.3 denotes a list of presented methods covered in this paper. It includes the presented method along with an employed methodology and a link to the corresponding GitHub page of their implementation. We intend to facilitate it for research communities who need to follow the same program setting to compare and evaluate different methods. The majority of the implementation are developed by using Python programming language. In addition to official implementations, there are some toolkits like DeepCTR [132] and DeepCTR-Torch [133] implemented based on Tensorflow and PyTorch platforms. They are not only provided users with third party implementations of contemporary methods in literature, but also they prepared a platform including several software components to build customized models. In this case researchers can examine different methods under the same input and output interface and apply the similar setting to all approaches.

Table D.1. Summary of benchmark datasets for online advertising user response prediction.

Category	Dataset	|Feature|	|Impression|	|User|	|Item|	|Click|	|Conversion|	Citation
Display Advertising	Criteo[a]	39	45,840,617	-	-	-	-	[16, 17, 44, 45, 53, 57, 64, 73–75, 78, 79, 106, 112, 117, 118, 136, 137, 145, 154, 155, 168, 176]
	Avazu(In-app)[b]	33	40,428,967	-	-	-	-	[14, 21, 28, 46, 53, 73, 74, 78–80, 82–84, 87, 97, 112, 118, 136, 137, 164, 176, 177]
	iPinYou[c]	16	19.5M	-	-	14.79K	-	[79, 80, 107, 117, 118, 121, 179, 180, 193]
	Taobao[d]	-	-	987K	4.1M	-	-	[110, 114, 120, 150]
Conversion post-click logs	Yoochoose[e]	9	-	-	-	-	-	[40, 83, 115, 170]
	Taobao[f]	-	84M	0.4M	4.3M	3.4M	18K	[88]
	Tencent[g]	12	-	-	-	50M	-	[103, 168]
E-commerce Platform	Amazon (Review logs)[h]	-	-	-	-	-	-	[50, 72, 110, 113, 116, 120, 134, 150, 156, 189–191]
	Avito[i] (Classified Ad portal)	27	170,588,667	-	-	-	-	[100–102]
	Frappe[j] (mobile app)	10	288609	957	4082	-	-	[73, 152, 162]

[a] http://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/
[b] https://www.kaggle.com/c/avazu-ctr-prediction
[c] https://contest.ipinyou.com/
[d] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649&userId=1
[e] https://recsys.yoochoose.net/challenge.html
[f] https://tianchi.aliyun.com/dataset/dataDetail?dataId=408&userId=1
[g] https://algo.qq.com/
[h] https://nijianmo.github.io/amazon/index.html
[i] https://www.kaggle.com/c/avito-context-ad-clicks/data
[j] https://github.com/hexiangnan/neural_factorization_machine/tree/master/data/frappe

D.3 Applications

In this subsection, we briefly review applications typically developed in relation to the user click-through rate prediction. The evaluation of the probability that users make interactions more than a click on promoted item or an ad have been studied from different aspect in many research.

Revenue per click prediction. This is an application to show earning of advertisers made from ad campaigns. In this application a metric called RPC(revenue per click) is used to compute the advertising interest based on user feedback in the form of clicks or conversions. Like the click-through rate prediction, the data sparsity is the common issue from advertiser perspectives to estimate the revenue based on user click responses. It means that not only the proportion of click events over impression are so small in user behavior histories, but also the number of bid units which receive the click response and lead to revenue is also very few. Although this metric is essentially important to analyze the performance of the online advertising, there are a few studies published in literature regarding this metric because of revenue confidentiality policy adopted by many companies. In the following part, we go over an important work in this domain:

Authors in [172] proposed a model to dynamically determine the data-driven hierarchy defined for the ad group and campaign and advertisers account. Meanwhile, they presented an empirical bayes method to get inferences through the hierarchical structure. In the context of the sponsored search advertising, bid values typically assigned for keywords to calculate the potential advertiser revenue. However, bid units are formed as the atomic units for the combination of a keyword in addition to match type and an ad group. The performance data were collected on the advertisers' side for experiments contain daily impressions, clicks, conversions and attributed revenue at the bid-unit level. Therefore, the prediction problem was defined as the prediction of the next day's RPC for given

Table D.2. Reported user response prediction (click-through rate prediction) results in different experimets for eight commonly used datasets. The performances were evaluated using AUC-ROC score metric.

Algorithm	Dataset(AUC-score)							
	Criteo	Avazu	Avito	Amazon	Taobao Alibaba	MovieLens	iPinYou	Bing News
MA-W&D[101]			0.7976[a]					
W&D-SSM[97]						0.7754[d]		
DeepMCP[102]			0.7927					
DTSN-I[100]			0.8395					
DIEN[189]				0.7792[b], 0.8453[c]				
DIN[190]				0.8818[b]	0.6031	0.7337[d]		
DIN-ATT[191]				0.9106[b], **0.9404**[c]		0.7429[d]		
DSIN[36]					0.6375			
DeepFM[44]	0.8016							
xDeepFM[75]	0.8052							0.8400
FNFM[177]	0.7470							
FiBiNET[53]	0.8021	0.7803						
PNN[117]	0.7700						0.7661	
FNN[179]							0.7071	
AutoInt[137]	0.8061	0.7752				0.8456[e]		
FAT-DeepFFM[176]	**0.8104**	0.7863						
FGCNN[78]	0.8022	0.7883						
RippleNet[146]						0.9210[e]		0.6780
DKN[147]								0.6570
Fi-GNN[74]	0.8082	**0.8120**						
KNI[116]				0.9238[c]		**0.9704**[d], 0.9449[e]		
AutoGroup[79]	0.8028	0.7915					0.7859	
AutoFIS[80]	0.8009	0.7852						

[a] The experiment was performed with no sampling approach on the dataset
[b] Electronics section [c] Book section [d] Section includes 20M rating instances [e] Section includes 1M rating instances

Table D.3. Summary of classification based methods with open-source implementations

Category	Model	Framework	Link
Factorization machines	PNN [117], KFM [118]	Tensorflow,libFFM	https://github.com/Atomu2014/product-nets-distributed
	NFM [49]	libFM,Tensorflow	https://github.com/hexiangnan/neural_factorization_machine
	FiBiNET [53]	DeepCTR, DeepCTR-Torch	
	AFM [162]	Tensorflow	https://github.com/hexiangnan/attentional_factorization_machine
	Robust-FM [112]	Spark	https://www.dropbox.com/sh/ny6puvtopl98339/AACExLZ0waDL_ibWhfNItJfGa?dl=0
	FLEN [21]	Tensorflow	https://github.com/aimetrics/jarvis
	AutoFIS [80]	Tensorflow	https://github.com/zhuchenxv/AutoFIS
	Others[44, 57, 122, 177]	PyTorchFM	https://github.com/rixwew/pytorch-fm
Deep Learning	AutoInt [137]	Tensorflow	https://github.com/DeepGraphLearning/RecommenderSystems
	FGCNN [78]	DeepCTR [132], Tensorflow	https://github.com/shenweichen/DeepCTR
	CCPM [83]	DeepCTR-Torch [133] PyTorch	https://github.com/shenweichen/DeepCTR-Torch
	xDeepFM [75]	Tensorflow	https://github.com/Leavingseason/xDeepFM
	DIN [190]	Tensorflow-gpu	https://github.com/zhougr1993/DeepInterestNetwork
	DSIN [36]	DeepCTR [132], Tensorflow-gpu	https://github.com/shenweichen/DSIN
	DIEN [189]	Tensorflow	https://github.com/mouna99/dien
	Wide&Deep [22]	Tensorflow	https://github.com/Tensorflow/models/tree/master/official/r1/wide_deep
	FNN [179]	Theano	https://github.com/wnzhang/deep-ctr
	DSTN [100]	Tensorflow	https://github.com/oywtece/dstn
	W&D SSM [97]	Tensorflow	https://github.com/niuchenglei/ssm-dnn
	Meta-embedding [103]	Tensorflow	https://github.com/Feiyang/MetaEmbedding
	DeepMCP [102]	Tensorflow	https://github.com/oywtece/deepmcp
	FAT-DeepFFM [176]	PyTorch	(unofficial) https://github.com/p768lwy3/torecsys
	MA-DNN [101]	Tensorflow	https://github.com/rener1199/deep_memory
	RocketLaunching [188]	PyTorch	https://github.com/zhougr1993/Rocket-Launching
	GMP [15]	Tensorflow	https://github.com/graphmp/graphmp
	MIMN [110]	Tensorflow	https://github.com/UIC-Paper/MIMN
	UBR4CTR [114]	Tensorflow	https://github.com/qinjr/UBR4CTR
Network based	KNI [116]	Tensorflow	https://github.com/Atomu2014/KNI
	Fi-GNN [74]	Tensorflow	https://github.com/CRIPAC-DIG/Fi_GNNs
	RippleNet [146]	Tensorflow	https://github.com/hwwang55/RippleNet
	DKN [147]	Tensorflow	https://github.com/hwwang55/DKN
	KGNN-LS[148]	Tensorflow	https://github.com/hwwang55/KGNN-LS

bid unit, given the historical clicks and revenue data. The features are the hierarchical structure information of the bid units. It encompasses corresponding campaigns, ad groups, and keywords, as well as geo-targeting information at the campaign level, which are shared by the bid units under

each campaign. Following an empirical approach to get inferences through the hierarchical structure, they proposed an extended empirical bayes method that was capable of dynamically constructing the hierarchy and used the loss concept in decision tree models for optimization.

Mistaken click prediction. In performance-based business models like the pay-per-click, the quality of click event plays an important role for the revenue made for advertisers. Relying on click events, there is a probability that advertisers are charged by Ad Exchange Network using valueless clicks. These type of clicks is generally considered as accidental clicks quite often happen in mobile devices, when users are confronting interstitial ads. They are interrupted by the ads covering the whole screen. They may click on ads and be directed to a website and bounce back without spending a considerable time. Ignoring these type of click may lead to overestimation of click-through rate values for the mobile advertising.

To this end, authors in [142] proposed a data-driven method to detect mistaken clicks on ads. Although from advertisers perspective, valuable clicks are those followed by conversions, but it is not always true for all click events. In the context of Yahoo mobile apps, they categorized the clicks using extra information about the time users spending on the advertiser landing page into three sections accidental, short and long. Decomposing a dwell time distribution into three above classes, they proposed a technique to apply a smooth discounting factor to charge less advertisers with regard to accidental clicks.

Fraud in Online Advertising. Today, the main principle to target users for online advertising is tailoring ads to user interest profiles. It consequently leads to a billing model to charge advertisers and pay publishers based on how many times targeted users interact with ads. In this business model, publishers register themselves in the ad network to host ad placements and advertisers organize ad campaigns for target users. From publisher's perspective, the revenue from advertising is directly dependent to number of users interacting with ads in web-pages and the cost paid by advertisers for displaying targeted ads. The rate of investment in online advertising is ascending annually. It tempts some people to commit fraudulent activities. According to common performance based business models like the cost-per-click and cost-per-impression for the sponsored search and display advertising, fraudulent form of clicks and views for textual rich media and video ads has gained a lot of attentions [89].

In study conducted in [94], it is demonstrated that organizing user interest profile relying on user visits have a vulnerability can be used for web-based fraud activities like the cross-site request forgery scripting and click-jacking to embed hidden requests not initiated by users. The increase in adopting IoT based solutions let users connected to Internet from different devices. Compromised devices which do not follow basic security measures can be easy target for such exploitations [111]. The fraud activities could orchestrate an attack against DoubleClick Ad Exchange Network to a manipulate user interest profile which can modify the publisher revenue. The main attribute of this attack is preparing a mechanism to modify user interest profiles without explicit interactions with ad exchange network and further knowledge about external involving factors. It was planned to generate polluted user profiles worked based on the behavioral targeting and re-targeting which led to present biased targeted ads that in turn need the higher bid price made by advertisers and revenue for the publisher.

To deal with these smart threatening attacks, different studies have been conducted to take the advantage of machine learning based methods to address challenges from different aspects [98, 140, 141]. In [140] authors investigated to use an auto-encoder neural network and GAN to regenerate click events. They designed a neural network model to predict a fake click events by adding some extent of noise to input data. In the context of sponsored search advertising, the threat of fraud crowdsourcing activities was discussed in [141]. In this case, the fraudulent behaviors including a series of search and click ads is distributed among vast number of web-publishers where fraudulent traffic is buried in the majority of normal traffic. They constructed a graph to represent a click history of users. They then

applied a clustering algorithm using a dispersity filter to find the coalitions that attacks are concerted against a common set of advertisers.

Others. The application of predicting user response rates is not limited to above applications. They are some studies consider the user response from different aspects to predict short-term final desirable purchase activities [15, 40] or long-term consistent influence like a branding [10] in common mobile and display advertising.

D.4 Future Directions

User response prediction is considered as an important task to evaluate the effectiveness of online advertising and recommender systems. It has led to various research studies to address this problem. In this section, we present a number of possible future directions identified in the recent studies. They can be found useful by researchers in the community to develop the next solutions.

Joint optimization models based on business value metrics. Implicit feedback like CTR value prediction is commonly used as an objective to select candidate ads for end-users [48, 147]. An accurate prediction of CTR values can make a direct impact in the biding value of ad in the display advertising [17]. Although CTR values can potentially show the user tendency with regard to items and commercials in online provider systems, it is not a definite indicator to show the success of business to gain new customers. Accordingly, business oriented conversion rate values, which convey additional information about user intent with regard to products and services in online systems, are proposed. For example, by comparing different user conversion feedback values initiated by click incident on items such as add-to-cards, add-to-wishlist and order productsa, order rate is found to be the most robust objective rather compared to add-to-cart to provide relevant personalized recommended list made by e-commerce search [58]. Pareto efficient learning-to-rank algorithm [76] is also proposed to calculate solution by aggregating two loss functions for CTR and Gross Merchandise Value (GMV) prediction using proper constraints. Studies such as [108, 185] encompass a couple of user interactions as the conversion user feedback. These actions are mapped to profit value which are considered as the reward for the system. In the model training following reinforcement learning, a policy to find better user actions is learnt to maximize accumulated profits and accommodate user preferences. Learning multiple factors with regard to optimize user preferences and commercial profits have a potential to shape the new trend in developing recommendation system algorithms.

Neural Architecture Search for User Response Prediction. Deep neural networks are becoming increasingly popular in recommendation and online advertising, but training process for them are potentially expensive. For many methods, the structure of neural network are manually set via extensive empirical studies. In our review, we have seen few attempts like [117, 179] to present fully connected neural network for user response prediction. To model user interactions with other components in the context of display advertising, diamond shape of multi-layer perception network to have larger hidden layer is suggested by comparing different structures. But broadly speaking, this process is not well-studied in the literature and heavily relies on intuitive understanding of developers and knowledge from the application domain like [79, 80]. Neural architecture search is to automate the architecture selection process to find the best neural network architecture via an optimization procedure [196]. Multi-objective evolutionary optimization [136] has recently been proposed to select network architecture, by organizing the search space as a direct acyclic graph, and utilizing learning-to-rank to search and filter out a selection of architectures in each iteration. Future research may consider domain knowledge and business metrics to design efficient and effective neural architecture search strategies for user response prediction.

Online Learning User Response Prediction and Recommender Systems. Online environment is inherently dynamic that the stream of data are changing over time. It may lead a gradual shift in user

preferences which can affect the performance of predictive models. So recommendation systems generally need to apply online learning (retraining mechanisms) to update and tackle new user interactions. This is partially related to cross-domain recommendation approaches that a pre-trained recommendation model is applied to different downstream tasks [171]. The full training of model is the straight-forward solution which is only helpful when we have a limited amount of data [87]. For online scenarios, selection based retraining and fine-tuning are other types of retraining solutions. The former applies a selection method for sampling of older user interaction and new data to create an updated training data while the latter proposes a transferring strategy to train a model using the new user interaction information. Very recently authors in [182] propose to learn transfer component in a cyclic fashion using meta-learning approach for sequential input data. Nonetheless, this topic is one of the important subjects for practical recommendation systems that needs to be well-addressed in future.