



Deep Learning for User Interest and Response Prediction in Online Display Advertising

Zhabiz Gharibshah¹ · Xingquan Zhu¹ · Arthur Hainline² · Michael Conway²

Received: 13 September 2019 / Revised: 1 December 2019 / Accepted: 29 December 2019 / Published online: 17 January 2020
© The Author(s) 2020

Abstract

User interest and behavior modeling is a critical step in online digital advertising. On the one hand, user interests directly impact their response and actions to the displayed advertisement (Ad). On the other hand, user interests can further help determine the probability of an Ad viewer becoming a buying customer. To date, existing methods for Ad click prediction, or click-through rate prediction, mainly consider representing users as a static feature set and train machine learning classifiers to predict clicks. Such approaches do not consider temporal variance and changes in user behaviors, and solely rely on given features for learning. In this paper, we propose two deep learning-based frameworks, LSTM_{cp} and LSTM_{ip}, for user click prediction and user interest modeling. Our goal is to accurately predict (1) the probability of a user clicking on an Ad and (2) the probability of a user clicking a specific type of Ad campaign. To achieve the goal, we collect page information displayed to the users as a temporal sequence and use long short-term memory (LSTM) network to learn features that represents user interests as latent features. Experiments and comparisons on real-world data show that, compared to existing static set-based approaches, considering sequences and temporal variance of user requests results in improvements in user Ad response prediction and campaign specific user Ad click prediction.

Keywords Click prediction · Display advertising · Campaign · LSTM network · Deep learning

1 Introduction

Computational advertising is mainly concerned about using computational approaches to deliver/display/serve advertisements (Ad) to audiences (i.e., users) interested in the Ad, at the right time [1]. The direct goal is to draw users' attention, and once the Ads are served/displayed on the users' device, they might take actions on the Ads and become potential buying customers. Due to the sheer volumes of online users,

the large number of advertisements, and different backgrounds and interests of users (including their changing habits and interests), finding users' interests is often the key to determine whether a user is interested in a certain type of Ad or a specific Ad.

1.1 Online Display Advertising Ecosystem

Figure 1 shows a simplified view of the online display advertising ecosystem. Whenever a user, also called an audience, launches an URL request from a publisher's web page, their request will immediately trigger an Ad call (i.e., an opportunity), if the requested web page contains publisher's Ad banner (the placeholder for displaying the Ad). This Ad call creates an opportunity for the publisher to find an advertiser to place their Ad on the user requested page, so the Ad is eventually delivered to the user.

In a typical real-time bidding setting, publisher will forward the Ad call as a bidding request to an AdExchange, where thousands of advertisers are connected and are looking to buy the Ad opportunity and display Ads to users. After casting the bid auction, AdExchange collects all bid

✉ Zhabiz Gharibshah
zgharibshah2017@fau.edu

Xingquan Zhu
xzhu3@fau.edu

Arthur Hainline
Arthur@bidtellect.com

Michael Conway
mike@bidtellect.com

¹ Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA

² Bidtellect Inc., Delray Beach, FL 33483, USA

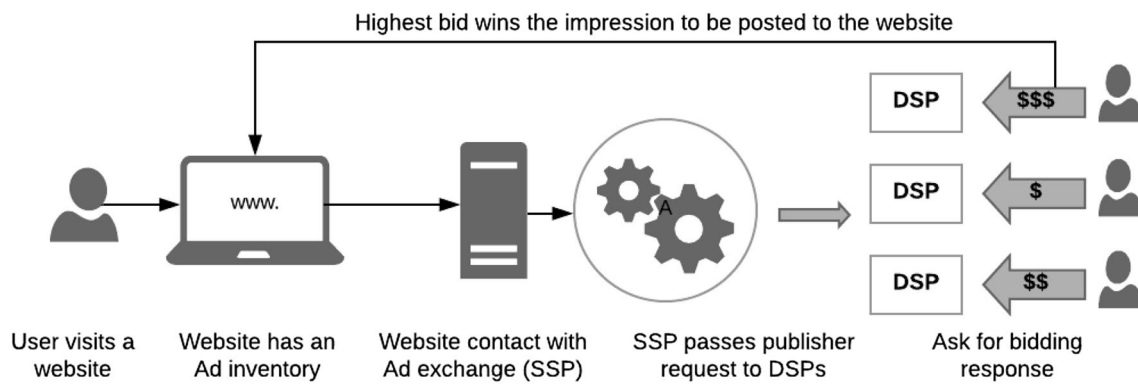


Fig. 1 A conceptual view of the real-time bidding system for display advertising. From left to right, a user/audience visits a publisher's web page containing one or multiple Ad banners. The publisher sends audience information to an AdExchange. The AdExchange sends Ad inventory, as a bid request, to advertisers which use demanding side platform (DSP) to manage their bids. The advertisers place bids as a

bid response. The bidder with the highest bidding price wins the bid and receives winning notice from the AdExchange. The winning bidder sends Ad scripts to the publisher's web page and rendered in the Ad banner, resulting in the advertiser's Ad being displayed/served on the audience's device

responses from advertisers, through their DSP (demanding side platform). The one with the highest bid wins the opportunity, and the advertiser's Ad script will be forwarded to the user requested pages, resulting in an Ad being displayed (or served) on the user's device. A display of the Ad on the user device is then called an Ad impression, which concludes a real-time bidding circle normally happening within tens of milliseconds.

Once the Ad impression is served on the user's end devices, users' action can vary significantly, depending on the viewability [1] of the Ad impression and many other factors. In some cases, although an Ad is served on the user's device, it might not be visible to the user (e.g., the Ad is not in the active browser window of the device, which is referred to as "below the fold"). As a result, no action or response is expected from the users.

If the served Ad impression is visible to the user, it is considered a *view*. In this case, if the user clicks on the Ad impression, it is referred to as an Ad *click*, and the users' click action is considered as their response. After users click on the Ad, they are normally directed to another web page (i.e., landing page). If users finish certain required actions on the landing page, e.g., downloading a software, filling into a form or completing a transaction, this is considered a *conversion*. In this case, the conversion action followed by a click action is also considered as a user response.

In this paper, we focus on the prediction of user click, instead of conversion, but the general principle of using deep learning for user response prediction can also be applied for the conversion prediction task.

1.2 User Response and Interest Prediction

In reality, due to the sheer volumes of online users and limited budget, advertisers cannot afford to place bid on every single Ad auction and have to determine possible interests of the users and then place bid on the audiences whose interests match the advertiser's Ad campaigns [1]. (An Ad campaign is the set of advertisements with a specific advertising theme and objective, pre-defined by the advertiser. For example, a hotel chain may define an Ad campaign to promote its hotel sales during the spring break in South Florida.)

In a display advertising setting, finding users' interest is approved to be a significant challenge, because according to the IAB openRTB specification [2], AdExchange often only passes very little information about the user, such as user device type, user agent, page domain name and URL. As a result, industry commonly relies on some *generative* modeling. Historical data are used to build tree-structured, whose parameters are used to derive the click-through rate (CTR) value of a new impression. Common generative models include CTR hierarchy trees [3] or hierarchical Bayesian frameworks [4]. One inherent advantage of the generative model is that the model provides transparent interpretability for business to understand which factor(s) contribute the most to the CTR values. However, due to the limitations of the models, such methods can normally estimate only a handful of parameters (e.g., using a number of selected factors to split the tree hierarchy) and are unable to consider many rich information from users, publishers and Web sites for accurate CTR estimation.

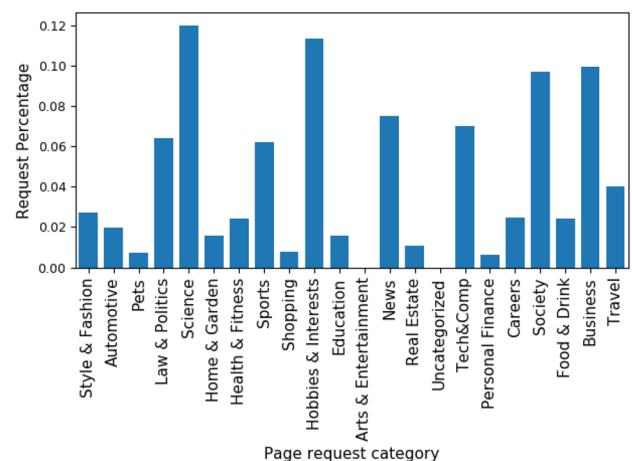
Different from generative models, the increasing popularity of machine learning, particularly deep learning, has driven a set of predictive modeling methods, which treat user clicks as binary events, and uses supervised learning to train a classifier to predict the likelihood of an Ad impression being clicked by users [5], including some deep neural network-based CTR estimation methods [6]. Such methods normally work on tens of thousands of features and are often more powerful than generative models, but have very little transparency in terms of the model interpretability.

1.3 Challenges and Solutions

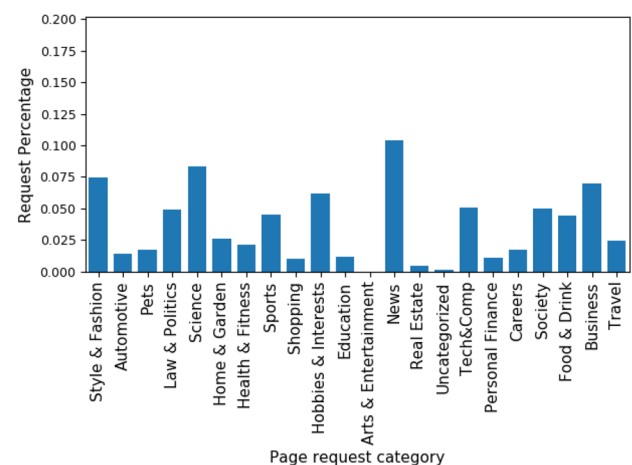
While existing predictive methods have made significant progress, they often consider user information as a static feature set. By doing so, they ignore temporal changes in the user information and therefore cannot accurately predict user interests. In reality, users may have diverse interests, and their behaviors might be consistent only in a short period of time. Therefore, when ignoring temporal order information, user behaviors may appear to be rather random. For example, Fig. 2 reports the category information of the web pages users visited in one day^{1,2} (i.e., the interests of the users), where Fig. 2a shows the interests of the users who clicked on the “Technology and Computing” Ad campaign, and Fig. 2b shows the interests of the users who clicked on the “Shopping” Ad campaign. The results show that if temporal information were ignored, user interests are nearly evenly spread out across several page categories, meaning users have been visiting different types of web pages, making it difficult to estimate connections between users’ page visits and Ad campaigns they are interested in.

In this paper, we propose to consider temporal user information to estimate user clicks and user interests. We generalize these two problems as a binary classification task (for user click prediction) and a multi-class classification task (for user interest prediction). More specifically, we collect users’ page visits as a temporal sequence and train deep LSTM (long short-term memory) networks to make predictions [7]. A unique strength of the proposed model is that it considers users’ temporal information to model their response and interests.

The remainder of the paper is structured as follows. Section 2 reviews related work in CTR prediction, user interest modeling and IAB page categorization. Section 3 reports the proposed LSTM frameworks for user response prediction



(a) “Technology & Computing” campaign



(b) “Shopping” campaign

Fig. 2 Histogram of requested pages based on users’ clicks on two specific types of Ad campaigns: **a** “Technology and Computing” campaign and **b** “Shopping” campaign. The x-axis denotes the category of pages visited by users who clicked on the “Technology and Computing” (a) or “Shopping” campaigns, and the y-axis shows the percentage of each page category. Overall, if the temporal orders were ignored, users’ page visits are nearly evenly spread out across multiple categories, making it difficult to estimate connections between users’ page visits and Ads they are interested in

and user interest modeling, followed by experiments and validations reported in Sect. 4. We conclude the paper in Sect. 5.

2 Preliminary and Related Work

2.1 Click-Through Rate (CTR) Prediction

For online advertising, click-through rate prediction is one of the most essential tasks whose accuracy influences the revenue of businesses in this domain. Most research in CTR

¹ This work is sponsored by an industry partner and the data used in the study and the experiments are collected from the industry partner’s bidding engine.

² The user visited pages are referred to as the web pages on which the industry partners’ Ads have been served/displayed to the users.

estimation is based on well-studied data analytic techniques. For example, key features like landing page URL, keywords, Ad title, Ad text, etc. can be extracted from search advertising and then help train logistic regression model to predict whether a search advertisement will be clicked [8]. The conclusion showed through experiments that with regression model and their feature set, CTR estimation gained a significant improvement in terms of mean squared errors (MSE). For display advertising, machine learning models [9] have also been commonly used to predict CTR by using multivariate linear regression, Poisson regression and support vector regression (SVR), where sophisticated models like SVR has shown to have the best accuracy [9].

One main challenge in CTR and user response (e.g., click) prediction is that data are usually represented in categorical format. The approach to transform them into high-dimensional binary feature representation results in the sparsity issue. In addition, nonlinearity is also a common issue in click-through prediction in which using linear model like logistic regression classifier is more dependent on synthetic features and fails to learn nonlinear relationship [8]. To tackle these challenges, research [10] proposed to take advantage of the data hierarchy in nature by clustering and data continuity in time to use information from data close to the events of interest in contextual advertising. To capture nonlinearity, [11] introduced factorization machines (FMs) to address interaction among features. With great capability of deep neural networks, factorization machines were used to build an embedding layer to capture pattern between inter-field categories followed by fully connected layers in some studies [6, 12]. The authors in [12] introduced a product layer between embedding layer and DNN layer without pre-training of factorization machine. The negative point in this model is that they focused more on high-order feature interactions. To address this issue, DeepFM was proposed [6] to integrate the architecture of factorization machines and deep neural networks in a hybrid design for modeling both low-order and high-order feature interactions.

In addition to traditional machine learning methods, deep neural networks are also used to detect interaction between features gathered from user behaviors in the web. In studies like [13], convolutional neural networks were extended to learn complex interaction between elements in a certain alignment of Ad impressions to predict clicks. The later studies [6] showed that these networks are biased to neighboring features which may come up with local minimum solution with high time complexity. Because recurrent neural networks are able to retain memory between samples and capture relations between instances for long time steps in input data, RNN-based methods have been leveraged to model sequential dependency on click data. The authors in [14] used these networks to consider user browsing behavior for click-through prediction to deal with externalities. In this

case, click on an Ad might be affected by the quality of Ads shown in the long sequence of Ads. RNN networks were also employed [31] using click-through logs of a commercial search engine to model the user queries as a sequence of user context to predict the Ad click behavior and next item recommendation. Addressing the location of users, some work like [16] studies on deep spatial temporal residual networks to find the best trajectories which can be attached with certain Ads to increase the rate of influenced users in targeted locations. Meanwhile, some work [15] also demonstrated that the accuracy of prediction can be improved using techniques like subsampling, feature hashing through multitask formalizing of the problem.

2.2 User Interest Modeling

Understanding user interests is one of the major challenges of online digital advertising because the essential goal of advertising is to find best matching between audience (users) and advertisements. In search advertising, users' search keywords provide well-informed context information to understand their interests at the time of search. For real-time news stream advertising call, [17] proposed a novel rank-aware block-oriented inverted index to match news feed as a query to retrieve k most relevant Ads. For display advertising, user context information is very difficult to collect mainly because publishers and AdExchange often provide very little user information (such as domain names and page URL which are often noisy and inaccurate). In addition, user and data privacy regulations, such as EU GDPR (EU General Data Protection Regulation [18]), also forbid the collection of user identity information for advertising.

Because users' interactions with systems are limited to simple keywords used by users in search engines and the history of categories of Web sites visited by them in e-commerce domain, we need a model to predict user interest and describe their potential preference. Due to the demand for providing more optimal personalized online services for users, some work have been conducted through text mining to analyze search engine logs and user feedback. In [19], the result of analyzing online and search logs showed that user intent for product search can be classified into three categories of target finding, decision making and exploration. Using RNNs to predict user purchasing intent after the stream of clicks in e-commerce Web sites has also been studied in [20]. Using spatial and temporal contextual information in RNNs is another study which tries to predict the next location of users [21]. Recently [22] has been presented to extract latent temporal user interests and capture the dynamics of interest via the combination of attention mechanism and gated recurrent unit-based neural network from user behaviors. In [23], the authors proposed a self-attention network with bias encoding to model session as the intrinsic

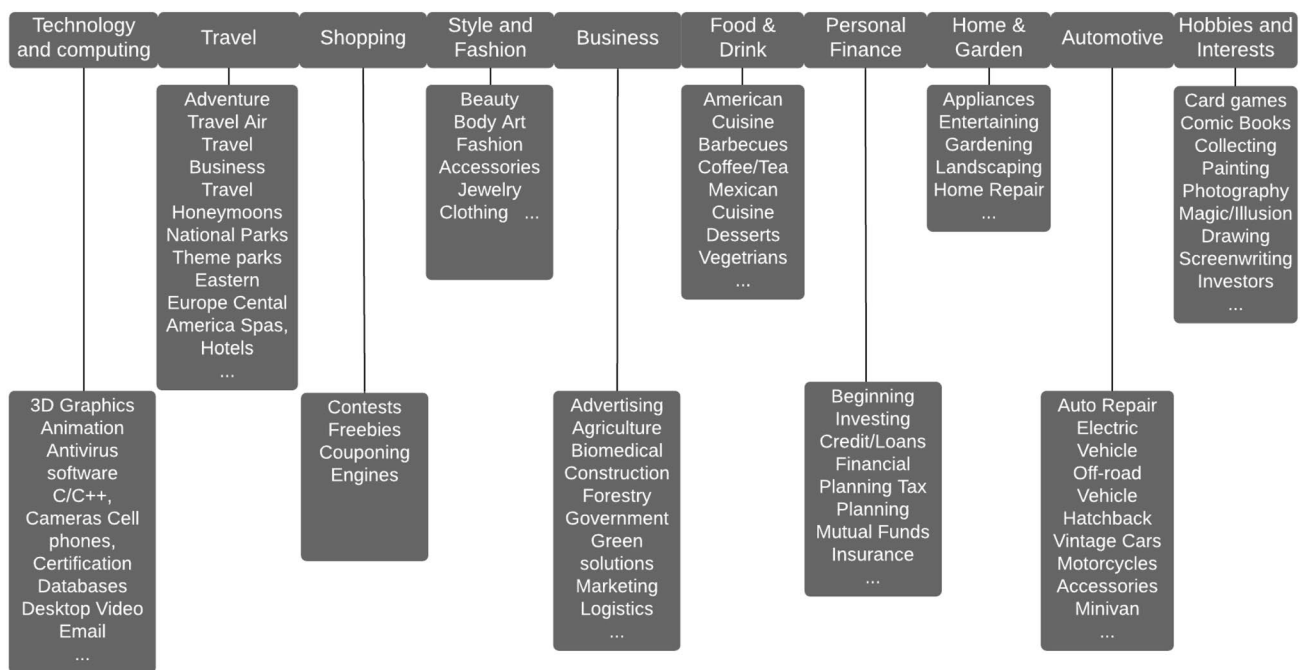


Fig. 3 A portion of the schema of the two-tier IAB (Interactive Advertising Bureau) Ad categorization. All Ad campaigns are categorized into two tiers. The first tier includes 24 categorizes (the figure only includes 10 first tier categorizes), and each first tier category

includes a number of sub-categorizes (shown as the list underneath each top tier category). In our experiments, we use 10 first tier campaign category listed in this figure

component in user behavioral sequences based on their observations that user have homogeneous behaviors within sessions and heterogeneous interactions between sessions.

Although deep learning (including RNNs) has been used to model users interests, existing methods mainly consider users log information as a static feature sect. As we show in Fig. 2, user interests are often limited to a short context period. Therefore, in this paper, we propose to use temporal sequence of various page categories visited by users to train RNN-based models to predict user clicks and user interests.

2.3 IAB Page Categorization

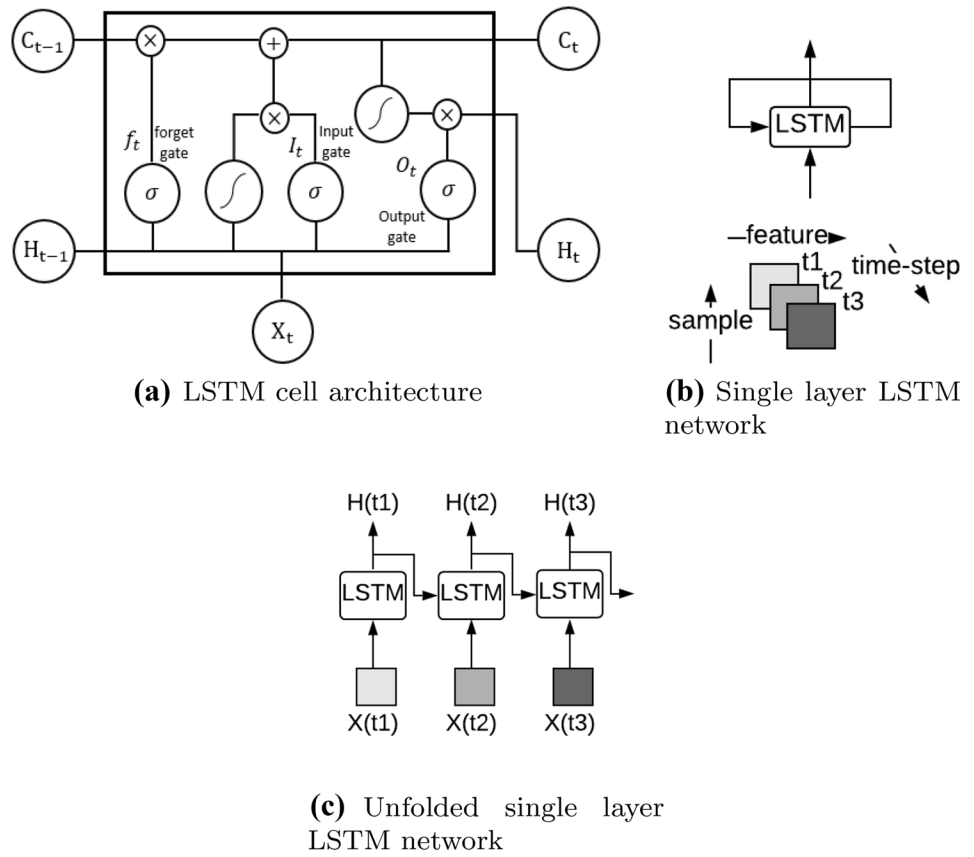
In the digital advertising industry, the majority of systems (particularly online advertising) operate based on real-time bidding (RTB) mechanism to sell, buy and display Ads in real time according to end users' visit as shown in Fig. 1. For the purpose of meeting market's demand for security, transparency and trust in advertising, Interactive Advertising Bureau (IAB) organization provides industry standards and develop legal support for the digital advertising. As part of these standards, in order to differentiate Ads in terms of their content and quality, a categorization in the form of taxonomy is provided. The two-tier taxonomy defines 24 first-level and 360 second-level categories for Ad impressions [24]. A portion of the IAB categorization is shown in Fig. 3.

Based on the IAB categorization, industry also provides online services to allow real-time query of the page category (using page URL as the query), so advertiser can instantaneously obtain the category of the page the user is currently visiting, before placing a bid on the user. In our research, we rely on the industry partner's system to collect page categorizations of the web pages which users have visited during a short period of time, and then use deep learning to learn user interests for prediction.

3 Deep Learning for User Response and Interest Modeling

When it comes to digital advertising for predicting the probability of click on Ads, there is a lack of sufficient information about user intentions and their interest for web surfing. Therefore, it is necessary to seek an efficient approach to estimate user interests based on their historical web behaviors. In this regard, user web surfing and Ads are important elements which should be considered as features for learning. Obtaining reasonable features to mine information can be a critical factor to achieve an efficient system.

Fig. 4 An LSTM cell and its network architecture. **a** Shows the detailed view of an LSTM cell, **b** shows a single-layer LSTM network with respect to the input, represented in a third-order tensor [sample, feature, time step], and **c** shows the unfolded structure of the single-layer LSTM network in **b**. In our research, we stack three LSTM layers to form a deep LSTM network (detailed in the experimental settings)



3.1 Problem Definition

Let U be the a set of users $\{u_1, u_2, u_3, \dots, u_n\}$ and R be a set of events. Each event denoted by $r_{u_i}^{t_i}$ represents the occurrence that an advertisement is displayed to a user u_j in a specific context at time t_i . In this case, the event is encoded as a real-valued vector ($r_{u_i}^{t_i} \in \mathbb{R}^d$). The context in display advertising industry is the page visited by the user which is in turn described by a hierarchy of page category IDs corresponding to various contextual information with different levels of granularity [15, 25].




The set of pre-defined page categories is denoted by \mathbb{C} equals to $\{c_1, c_2, \dots, c_{|\mathbb{C}|}\}$ where $|\mathbb{C}|$ is the number of categories (like tier 2 categories in Fig. 3). For a page visited by user u_j at time step t_i , its page categories can be shown in the form of array like $[c_1, c_2, c_3, \dots]$. For each user $u_j \in U$, we take the history of web pages visited by the user and denote it by $r_{u_j} = \{r_{u_j}^{t_1}, r_{u_j}^{t_2}, \dots, r_{u_j}^{t_m}\}$. Because of the variety in the number of Web sites visited by users, we have $r_{u_j} \in \mathbb{R}^{m \times d}$ where m is the maximum sequence length. Thus, given the historical records of all users as $R = \{r_{u_1}, r_{u_2}, \dots, r_{u_n}\}$ where $R \in \mathbb{R}^{n \times m \times d}$, $d < |\mathbb{C}|$, our **objective** is using historical user activities as the chronological sequence of requests before an arbitrary time step t_i to achieve the following two prediction tasks:


- *User Response Prediction* Predict the probability that a user may interact with an Ad at t_i by generating a click response. In our solution, we formulate this task as a binary classification task.
- *User Interest Prediction* Predict which type of campaign Ad a user might click. In our solution, we formulate this task as a multi-class classification task.

3.2 LSTM for User Modeling

Recurrent neural network (RNN) is an extension of feed-forward networks and has been successfully applied in various sequence data analysis and temporal sequence modeling [26]. While traditional RNN networks are unable to learn all term dependencies in sequences because of gradient vanishing or exploding problem [27], long short-term memory (LSTM) networks were introduced to use special multiple-gate structured cell to replace hidden layer nodes. Using LSTM cells in these networks has been shown as an efficient way to overcome these problems [27]. As shown in Fig. 4a, each LSTM cell (the building block of the LSTM network) includes three gate entries: input gate, forget gate and output gates, to control memorization and updating information learned from sequences. By taking more computational costs through these added elements, the flow of gradient across input sequences is tried to become stable. For each element

Table 1 Schema of the data representation

User	t_1	t_2	t_3	t_4	...	t_n
<i>Temporal order of audience response</i>						
u_1	$r_{u_1}^{t_1} : [c_1, c_2, c_3]$	$r_{u_1}^{t_2} : [c_1, c_3]$	$r_{u_1}^{t_3} : [c_1, c_4, c_5, c_6]$ 	—	—	—
u_2	$r_{u_2}^{t_1} : [c_2, c_3]$	$r_{u_2}^{t_2} : [c_4]$ 	—	—	—	—
u_3	$r_{u_3}^{t_1} : [c_{10}, c_7, c_3, c_{20}]$	$r_{u_3}^{t_2} : [c_1, c_3, c_{15}]$	$r_{u_3}^{t_3} : [c_6, c_{12}, c_{22}, c_{24}, c_1, c_3]$	—	—	—
u_4	$r_{u_4}^{t_1} : [c_8, c_{14}, c_{30}]$	$r_{u_4}^{t_2} : [c_2, c_6]$	$r_{u_4}^{t_3} : [c_{11}, c_{16}, c_{21}]$	$r_{u_4}^{t_4} : [c_4, c_7]$ 	—	—
...

We represent each audience (user) and his/her actions as multi-dimensional temporal sequence. Each row in the table denotes an audience, and t_1, t_2, \dots, t_n denote temporal order of the sequence. (If $i > j$, then t_i happens after t_j .) c_1, c_2, \dots, c_m denote the IAB tier 2 page category of the web page visited by the users. The click icon  denotes an Ad click event from the audience. Not all sequences result in click events

of input sequence, the following internal operations are done in LSTM blocks to follow one forward pass:

$$f(t) = \sigma(W_f x_t + V_f h(t-1) + b_f), \quad (1)$$

$$i(t) = \sigma(W_i x_t + V_i h(t-1) + b_i), \quad (2)$$

$$c(t) = f(t) \odot c(t-1) + i(t) \odot \tan h(W_c h(t-1) + b_c), \quad (3)$$

$$o(t) = \sigma(W_o x_t + V_o h(t-1) + b_o), \quad (4)$$

$$h(t) = o(t) \odot \tan h(c(t-1)). \quad (5)$$

According to Fig. 4, at each time step t forget gate in LSTM uses a sigmoid activation function in Eq. (1) to determine what amount of previous information should be retained from the cell state as the storage of historical information through weights and biases. Then, cell state is computed to store information by taking two following steps. Using input gate, Eq. (2) chooses which input values should be considered. In Eq. (4), $\tan h$ function builds new candidates which is added to value determined by forget cell in the previous time step. At last, Eq. (5) decides which part of current state should be shown in output gate to be used for the next round of training process.

Two significant challenges in online display advertising to model user response and user interest using deep learning approaches like LSTM networks are that the collection of online user behavior data are (1) in multi-variant categorical form because each page may belong to one or multiple categories and (2) user sequences of historical data may have different lengths because users' responses and actions vary over time. They result in multi-length sequences, where data points of each time step may also include variant features. More specifically, in our model, the historical data collected for user modeling contains page category IDs of the pages that a user visited during a short period of time. For a user at a particular time step, we have an array of category IDs

of the page visited by the user discussed in Problem Definition section. Such IDs are represented as $[c_1, c_2, \dots]$ which are in different lengths. Table 1 shows the sample of input sequential data used for user modeling.

3.2.1 One-Hot Encoding with Thresholding

In order to handle multi-length page categories as features to describe each visited page, we use one-hot encoding to represent them as sparse binary features. For each user, we have a sequence of visited pages attributed by a couple of page category IDs that corresponds to their content. Therefore, each time step can be shown as binary vector with length equal to the maximum number of categorical variables where 1 indicates the presence of each possible value from the original data. For example, in Table 1, at time step t_1 the visited page of user u_2 is described by an array of page category IDs as features can be shown as $[0, 1, 1, 0, 0, 0, \dots, 0]$. The dimension of vectors for time step is determined by the number of unique page category IDs in the dataset that in our example, it equals to $|C| = 18$.

Concatenating these vectors generates a matrix with high dimensionality. Therefore, for features like page category IDs with high cardinality, using one-hot encoding usually leads to extra computational costs. In the past, much research has been done to work with such sparse binary features [11, 12]. To address this problem and in order to reduce the dimension of these vectors, we used an alternative to encode more frequent page category IDs based on a threshold-based approach. In this case, page category IDs are sorted based on the number of their occurrences. Those with repetitions more than the user-defined threshold will be kept for the next parts.

3.2.2 Bucketing and Padding

The variable length of sequences, like samples in Table 1, is another technical challenge. To handle sequences of any length and capture short and long dependencies in input

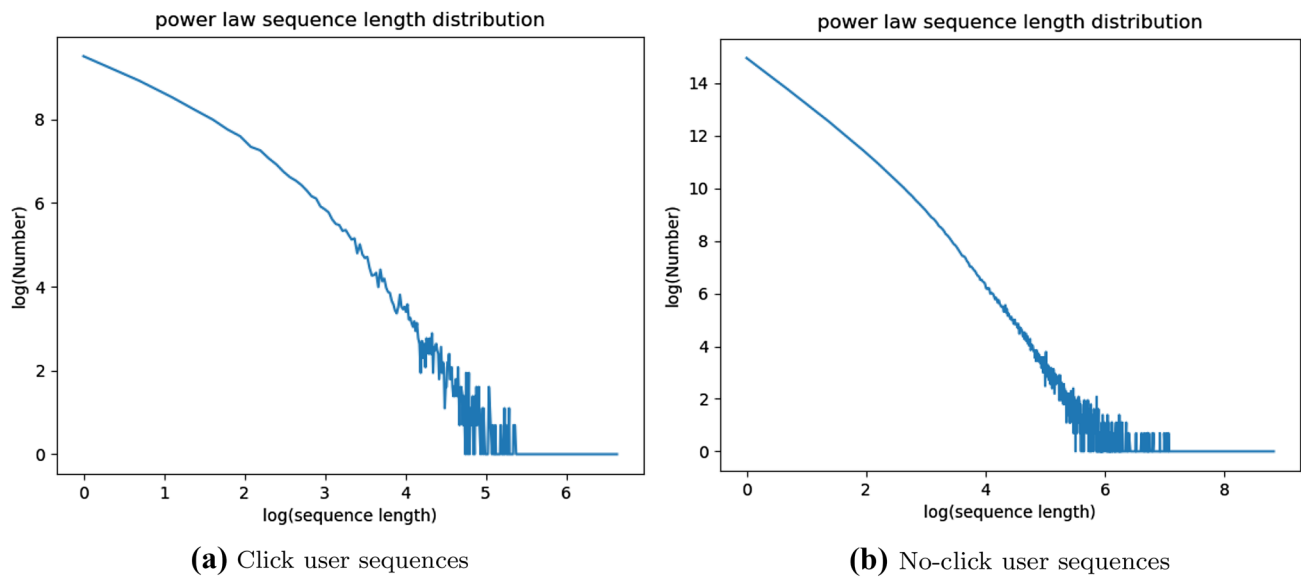


Fig. 5 Distribution of sequence lengths for **a** click users and **b** non-click users (i.e., users with or without click events). Both plots, shown in log–log scale, follow the power-law distribution, meaning major-

ity of samples have sequence length less 100. Samples with larger sequence length are rare

data, padding with constant value (e.g., inserting zeros) is a straightforward strategy to make input dimensions fixed. However, applying this approach to train LSTM with wide range of sequence lengths not only is computationally expensive, also adds extra zero values resulting in bias in outcomes and changes input data distribution. Therefore, we propose to combine padding and bucketing to best utilize temporal information in sequences without inserting too many padding symbols.

In Fig. 5, we report the length of user sequences for both click users (i.e., users who have a click event) and non-click users (i.e., users who do not have a click event). The results show that both click and non-click users' sequence lengths follow power-law distribution, meaning majority user sequences have short length (below 100).

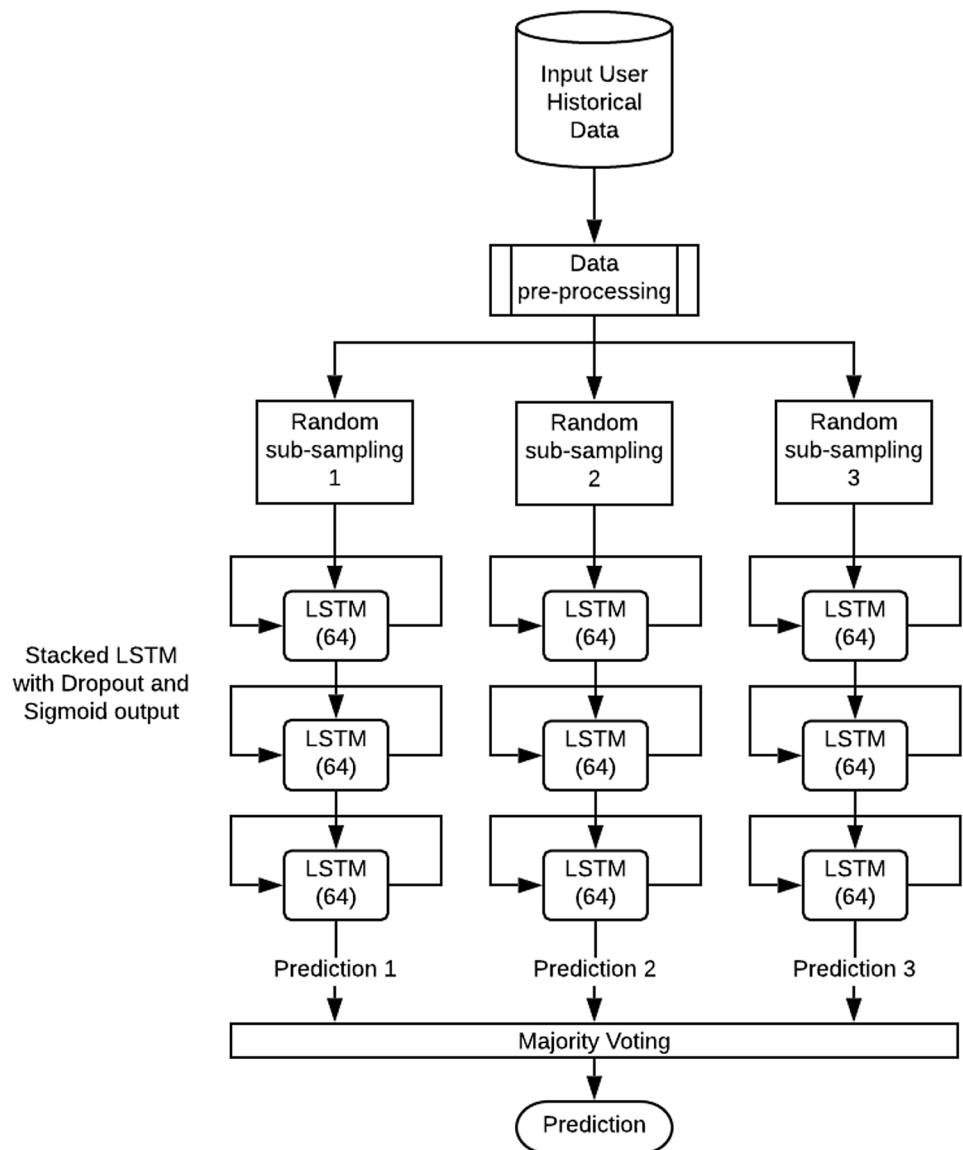
To combine bucketing and padding, we construct several buckets in training samples, where sequences in each bucket have the same lengths corresponding to the range of sequence length in the dataset. Each sample is assigned to one bucket corresponding to its length. In this case, padding of samples mitigates to inside of buckets being used just for assigned sequences as much as necessary to fit into the bucket. As the most important item in the sequence of request page categories is the last item that corresponds to the possible user click response, we use pre-padding approach. It means that each short sample inside buckets with the length lower than bucket size is pre-padded to become a sample with length equal to the maximum length in that bucket.

Following the idea, we designed an ensemble learning method for multi-class classification task. Rather than using the original splitting to generate buckets as the representative subset of samples in input space, we just use the sequence length of buckets to indicate one representation of samples through truncating time steps. It means that for each representation, all samples in input data are trimmed to the selected sequence length by removing some time steps from the beginning of sequences. Then in order to obtain classification, we build one LSTM model for each representation. The final result of classification is generated by applying majority voting as the result merger of all models.

3.3 LSTM_{cp}: User Click Prediction Framework

Figure 6 briefly describes the structure of our proposed method for user click prediction problem as a binary classification. It includes the stacked LSTM model consisting of three LSTM layers followed by one fully connected layer with sigmoid activation to combine the output of hidden neurons in previous layers to predict click instances. In this case, the loss function is defined as the weighted binary cross-entropy which aims to maximize the probability of correct prediction. The weight introduced in this function allows a trade-off between recall and precision in both classes to mitigate the negative effect of the class imbalance problem [28] in our task:

Fig. 6 Proposed user click prediction framework ($LSTM_{cp}$). Given user request and response historical data, our goal is to train stacked LSTM classifiers to predict whether a new user is going to click an Ad or not, i.e., a binary classification task



$$L = 1/N \times \sum_{i=1}^N (y_i \times -\log(p(x_i)) \times w + (1 - y_i) \times -\log(1 - p(x_i))) \quad (6)$$

where N is the number of samples in training set. $y_i \in [0, 1]$ is target label, and $p(x_i) \in [0, 1]$ is the predicted value generated as the output of network. It represents the likelihood that how likely the sample x_i has a click response at the end. w is the coefficient which determines the cost of positive error relative to the misclassification error of negative ones.

3.4 $LSTM_{ip}$: User Interest Prediction Framework

Figure 7 outlines the model for user interest prediction. It is defined as multi-class classification to classify the number

of clicks in 10 different advertising campaigns. The number of buckets are defined uniformly over the range of sequence length in the dataset. Then, for each bucket, one representation of data is generated by trimming all longer samples and pre-padding shorter samples to the selected sequence length. Then prediction is made by following the ensemble learning approach. In this figure, LSTM block follows the structure mentioned in Fig. 6 except the last layer having softmax activation function. In this case, the objective function is similar to Eq. (6) when $w = 1$. It is actually an unweighted categorical cross-entropy loss function in which $p(x_i)$ is the output of the network after softmax layer.

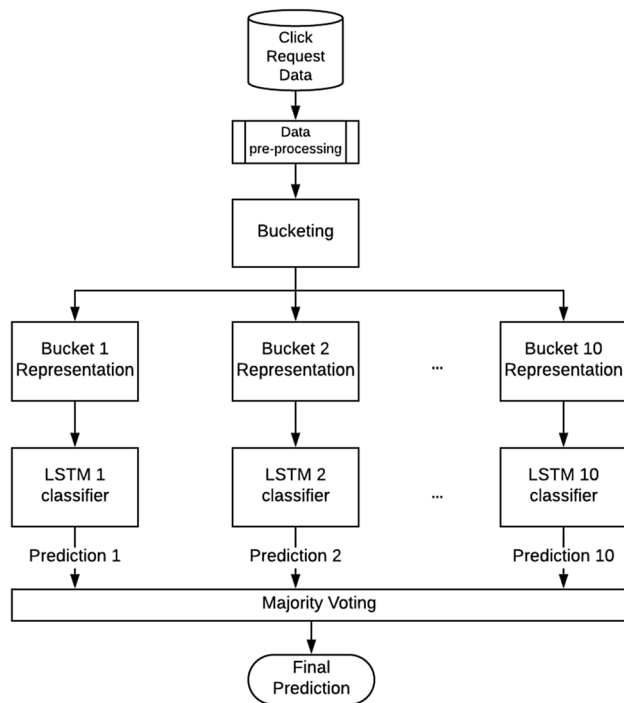


Fig. 7 Proposed user interest prediction framework ($LSTM_{ip}$). Given user request and response historical data, our goal is to train stacked LSTM classifiers to predict which Ad campaign a new user is going to click, i.e., a multi-class classification task

4 Experiments

In this section, we report experiments and comparisons made with several baseline methods on real-world data collected from our industry partner's DSP platform.

4.1 Benchmark Data

We pulled out data from our industry partner's DSP and prepared two datasets to validate user click and user interest predictions.

- **Post-View Click Dataset** This dataset is mainly used for validating binary user click prediction. We pulled 5.6 million users' request records from 1-day log events. These anonymous records include chronological sequences of various request categories which represent user browsing interactions. In this case, there are two types of positive and negative responses from users where success occurs if a post-view click takes place at end of a chain of visited impressions. Because of the rarity of positive responses (click) in digital advertising, this dataset suffers from severe class imbalance problem. Therefore, to deal with this issue, we use random down-sampling to obtain a training set with class distribution ratio of 10:90% corresponding to positive:negative samples.

- **Multi-Campaign Click Dataset** This dataset includes historical records with positive response in post-view click dataset. The positive response in this case is defined as occurred when users click on an Ad whose campaigns have categories mentioned in the first tier of Fig. 3. This dataset is mainly used for validating multi-class user interest prediction.

One issue we encountered in these datasets is that the sequence lengths are severely skewed where a large proportion of sequences are very short in length even less than 3 time steps, as shown in Fig. 5. Our bucketing and padding combined approach, introduced in Sect. 3.2, is specially designed to handle this challenge.

4.2 Baseline Methods

In order to assess the proposed frameworks for user click and user interest models, we implement two types of baselines approaches: (1) generic machine learning-based approaches and (2) deep learning-based approaches. All baselines use the ensemble framework shown in Figs. 6 and 7. All approaches are compared based on the same training/test data and are using same number of base classifiers.

4.2.1 Generic Machine Learning-Based Approaches

In order to validate the performance gain of deep learning-based methods, compared to generic machine learning algorithms, we implement ensemble frameworks similar to Figs. 6 and 7, using different types of generic machine learning method, including naive Bayes, random forest, logistic regression, linear support vector machine (SVM) and SVM.

For fair comparisons, all ensemble frameworks use the same number of base classifiers, so methods are taking advantage of additional base classifier for a better performance gain. In the following results, we directly use each classifier, such as naive Bayes or random forest, to refer to the ensemble framework implemented using respective machine learning algorithms.

4.2.2 Deep Learning-Based Approaches

- **CNN** Convolutional neural networks (CNN) and LSTM are two commonly used deep learning algorithms. To compare their performance for user click and interest prediction, we use CNN as base classifiers to train ensemble frameworks, and refer to the results as CNN in the tables/figures below.
- **DeepFM** DeepFM is of the hybrid architecture including factorization machines and deep neural networks extended on wide and deep method [29] to learn high-

order and low-order interactions of input data without any feature engineering. In our experiment, we use DeepFM as a base classifier to train ensemble frameworks for user click and interest predictions.

- **LSTM_{cp}**: LSTM_{cp} is the proposed method which uses the framework in Fig. 6 for user click prediction.
- **LSTM_{ip}**: LSTM_{ip} is the proposed method which uses the framework in Fig. 7 for user interest prediction.
- **LSTM_{bp}**: In order to validate whether the bucketing and padding (Sect. 3.2.2) provide additional benefits for user interest prediction, we implement the framework in Fig. 7 using LSTM as the base classifier, but removing the bucketing and padding modules. In other words, LSTM_{bp} and LSTM_{ip} are similar, except that the former does not have the bucketing and padding. If LSTM_{ip} outperforms LSTM_{bp}, it would imply that bucketing and padding provides additional benefits for user interest modeling.

4.3 Experimental Settings and Performance Metrics

4.3.1 Experimental Settings

We implemented seven methods for comparisons (including our proposed method). All neural network-based models were implemented through TensorFlow and CUDA to take advantage of using GPU and trained by Adam optimization as a variant of gradient descent. The remaining models are built using scikit-learn library in Python. For training models, the dataset is split into training, test and validation sets using 70:20:10 ratio. In the data preprocessing step, we convert input sequential data to binary vector by one-hot encoding with multiple categorical campaign IDs and disposing of less frequent ones. So we select top categorical campaign IDs based on frequency using a threshold to keep those categories with more 1000 occurrence in our dataset. To control over-fitting problem in neural networks early stopping mechanism is used to stop after 10 subsequent epochs if there is no progress on the validation set. Dropout rate was set at 0.4 for neural networks. For the rest of methods, L₂ norm regularization is used in the training process. All experiments are evaluated based on fivefold cross-validation.

4.3.2 Data Preparation and Model Training

Because LSTM requires input to be arranged in tensor format, for our proposed method, we represent data as a third-order tensor ($\mathbb{R}^{n \times m \times d}$), where n , m and d correspond to the number of users, the frequent sequence length and the number of most frequent page category IDs, respectively (in our experiments, we set $m = 70$ and $d = 153$ which are based on the statistical characteristics of the data. For remaining methods, their input data are collected by projecting the

third-order tensor data ($\mathbb{R}^{n \times m \times d}$) to a second-order tensor $\mathbb{R}^{n \times d}$, by adding up values in sequence length dimension). So all other baselines (except the proposed method) do not consider temporal information of the users' requests, but aggregate users' requests as a table for learning.

Each model is trained by minimizing weighted binary cross-entropy shown in Eq. (6). By default, we use cost ratio as 5 for positive samples because of the effectiveness seen in our experiments. All models are trained based on the same training sets (the training sets are converted to a third-order tensor or a second-order tensor when needed) and are evaluated on the same test sets.

4.3.3 Performance Metrics

We use area under the receiver operating characteristics curve (AUC) as the major evaluation metric because it shows the model accuracy of ranking positive cases versus negative ones. We also employ accuracy, F_1 -measure, precision and recall as additional performance metrics.

4.4 Performance Comparison

4.4.1 User Click Prediction Results

As a binary classification task, the performance of proposed method is compared with SVM, random forest and logistic regression in addition to a variant of convolutional neural network (CNN) [30] and DeepFM method [6]. The first neural-based opponent (CNN) includes the convolutions layers with three filter windows ($h = 3, 4, 5$) followed by dropout regularization ($p = 0.2$). The deep component in the DeepFM model consists of a three-layer feed-forward neural network with 400-400-400 hidden neurons and dropout rate at 0.5. Since input data have an extremely imbalanced class distribution with around 5,646,569 non-click user sequences (negative samples) versus 31,144 click user sequences (positive samples), we use random under-sampling and ensemble learning to build the model in Fig. 7.

Table 2 User click prediction results (binary classification task)

Method	Precision	Recall	F_1 -measure	AUC	Accuracy
Naive Bayes	0.2969	0.2730	0.2844	0.6029	0.8708
Random forest	0.2991	0.2770	0.2876	0.6048	0.8709
Logistic regression	0.3355	0.3027	0.3183	0.6202	0.8780
Linear SVM	0.3699	0.2474	0.2963	0.6018	0.8895
SVM	0.3938	0.2176	0.2803	0.5914	0.8949
CNN	0.3423	0.4427	0.3862	0.6769	0.8672
DeepFM	0.3077	0.4148	0.3533	0.6589	0.8571
LSTM _{cp}	0.3140	0.5183	0.3910	0.7003	0.8481

The better performance in experiments are shown by bold-face values

Table 2 reports the result of prediction using different methods. The first obvious result is that deep learning methods outperform the remaining methods which verifies the power of neural networks to capture nonlinear correlation between input features and classes. Comparing three deep learning methods based on AUC–ROC score and the average score of F_1 -measure and recall, our proposed method is the best among all. As our proposed method pays more attention to history of requested pages before click, having higher performance in our proposed method shows the importance of this feature in click prediction.

4.4.2 User Interest Prediction Results

For multi-class user interest prediction task, we compare proposed approach with naive Bayes, random forest (with 100 tree estimators), logistic regression and two versions of

SVM with linear and RBF kernels and DeepFM methods. The input data are click samples used in the previous task for click response prediction.

In Fig. 8, we report the Receiver Operating Characteristic (ROC) curves and AUC values of different methods for user interest prediction. In addition, Table 3 also summarizes the performance of different methods for user interest prediction using other performance metrics. Considering the AUC values, the overall results in Table 3 illustrate that the proposed method in this classification task outperforms the others. It shows the effectiveness of $LSTM_{ip}$ network in detecting the correlation of sequential data and click response. Having the higher performance for neural network approaches compared to linear classifiers like LR and linear SVM emphasizes the importance of nonlinear latent patterns in input space. According to results, SVM predictors with nonlinear RBF kernel are not successful either in this task or the previous

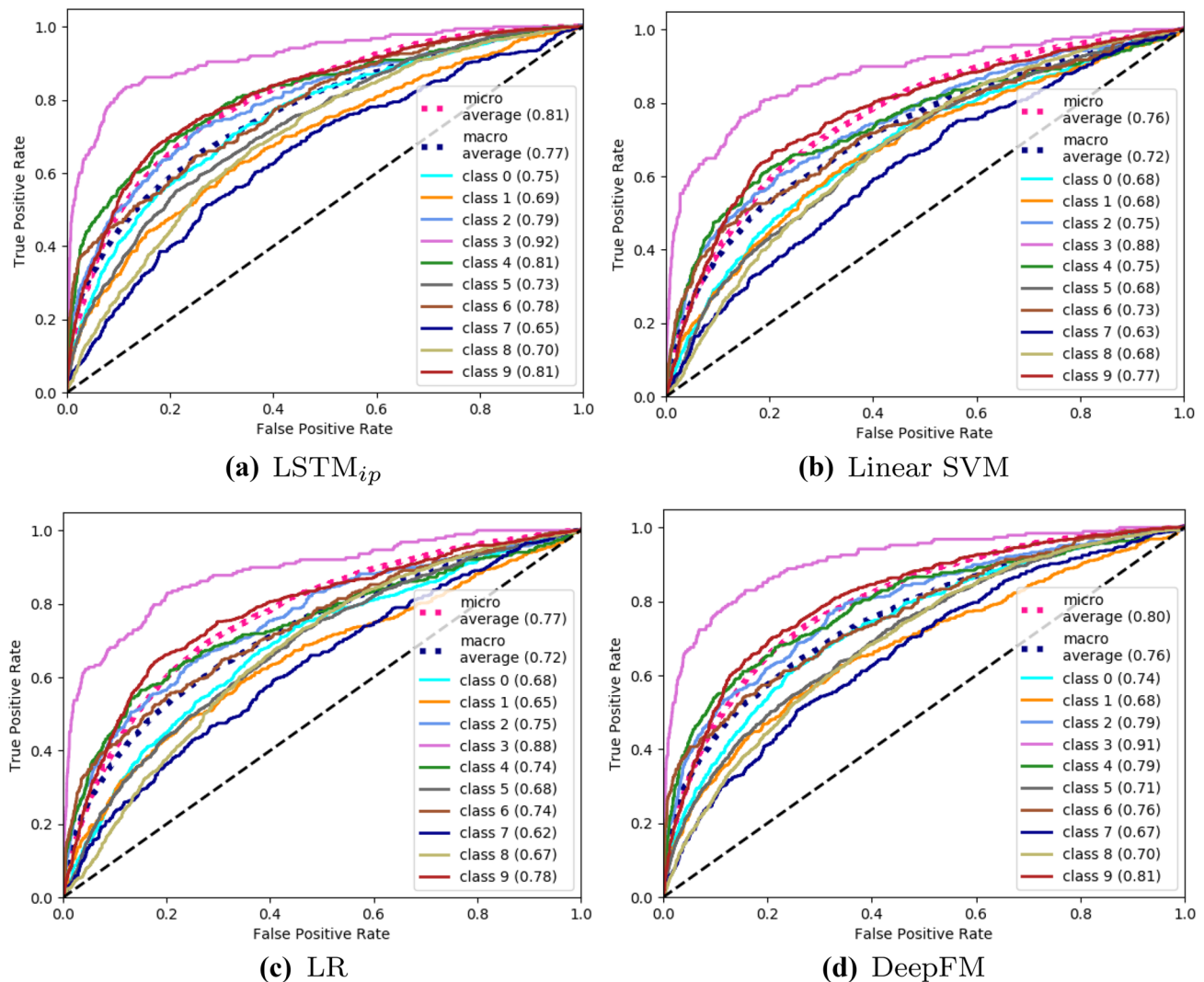


Fig. 8 Receiver operating characteristic (ROC) curve and the AUC values for user interest prediction. Each colored curve denote the ROC curve prediction for one campaign category (there are 10 campaign categories in total). Each plot denotes one type of classifier used for learning

Table 3 User interest prediction results (multi-class classification task)

Method	Precision	Recall	F_1 -measure	AUC	Accuracy
Naive Bayes	0.2486	0.2940	0.2713	0.6708	0.2713
Random forest	0.3640	0.3401	0.3697	0.7404	0.3697
Logistic regression	0.3268	0.3894	0.3268	0.7172	0.3268
Linear SVM	0.4004	0.2316	0.3207	0.7177	0.3207
SVM	0.1037	0.0654	0.0691	0.4593	0.0691
DeepFM	0.4007	0.3355	0.3777	0.7529	0.3777
LSTM _{bp}	0.3907	0.3368	0.3766	0.7511	0.3766
LSTM _{ip}	0.4015	0.3447	0.3845	0.7624	0.3845

one. It may be because of the reason that they cannot learn hyperplanes in nonlinear kernel spaces under too sparse data. Comparing among deep learning models, it can be seen that under the same LSTM classifier, applying bucketing and padding has achieved the higher performance compared to others including the base model and DeepFM methods.

Among all methods, although our proposed method obtains the precision score to be comparable to others, it achieves at least 0.01 gain in AUC–ROC score over baseline models. Considering the accuracy values, although the performance of methods are below 0.5, it still is higher than 0.1 using random classification for this multi-class classification. Hence, it can show that using bucketing and padding technique can provide the great effectiveness in classification performance and be a great help for training deep networks to learn user interests.

4.4.3 Parameter Sensitivity Study

In this subsection, we conduct experiments to study the sensitivity of the proposed frameworks, shown in Figs. 6 and 7, with respect to different number of classifiers (n) for user click and use interest prediction.

Parameter Sensitivity for User Click Prediction In this section, we report the performance of our method using different number of classifiers (n) for user click prediction. The results are reported in terms of AUC–ROC and accuracy scores in Table 4. We test this parameter by changing the

value and considering other parameters take their default values.

The reported values include the worst mean AUC–ROC scores, the best mean AUC–ROC scores and the mean AUC–ROC scores of proposed framework in addition to the average majority voting AUC–ROC scores, i.e., the score to classify samples to the class which obtains the largest number of votes. Our objective is to study how ensemble size affects the performance of various methods in the user click prediction by setting the value of classifier number (n) as 3, 5 and 7. According to the results in terms of AUC scores, when increasing the number of classifiers for the majority voting, the values of AUC and accuracy scores rise as the more information is added from multiple classifiers. Compared to the single classifier, the ensemble-based version of method provides an improvement in the performance of user click prediction. In terms of the AUC score, the best performance is obtained at the ensemble size of 3 which is mentioned in the prior experiments and shown in Fig. 6. When (n) exceeds this value, the performance becomes more stable. Using the higher number of classifiers for majority voting, our method still outperforms the baseline methods. This can represent that there is a robustness in the functionality of the method. Taking the accuracy scores into account for the proposed method, no ensemble method using a single classifier has the worst performance among all. We can see that with the increase in n , the accuracy of the proposed method is generally higher than the mean accuracy value and is also higher than the accuracy of method with no ensemble (ensemble size 1).

It is worth noting that in Table 4, our method (LSTM_{cp}) does not show the best accuracy compared to other methods (such as SVM) for the same ensemble size represented in Table 2. This is mainly because our goal is to maximize the user click and user interest prediction in terms of the AUC scores, which combines the accuracy on both positive (clicks) and negative (non-clicks) instances. Because user clicks are only a small portion of the dataset, accuracy does not reflect the genuine performance of the classifier for prediction.

Parameter Sensitivity for User Interest Prediction In the second part, we investigate the variation of ensemble size on the performance of proposed method for user interest prediction. The experimental results are shown

Table 4 Sensitivity study of the proposed framework (LSTM_{cp}) with respect to the ensemble size (n) for user click prediction

Ensemble size	AUC				Accuracy			
	Min	Max	Mean	Majority voting	Min	Max	Mean	Majority voting
1	0.6964	0.6964	0.6964	0.6964	0.8466	0.8466	0.8466	0.8466
3	0.6957	0.7015	0.6984	0.7003	0.8405	0.8507	0.8456	0.8481
5	0.6932	0.7011	0.6970	0.6997	0.8392	0.8539	0.8469	0.8503
7	0.6902	0.7028	0.6961	0.6991	0.8352	0.8572	0.8477	0.8512

Table 5 Sensitivity study of the proposed framework (LSTM_{ip}) with respect to the ensemble size (n) for user interest prediction

Ensemble size	AUC				Accuracy			
	Min	Max	Mean	Majority voting	Min	Max	Mean	Majority voting
1	0.7511	0.7511	0.7511	0.7511	0.3766	0.3766	0.3766	0.3766
5	0.7488	0.7545	0.7511	0.7591	0.3748	0.3810	0.3773	0.3833
10	0.7479	0.7602	0.7522	0.7624	0.3737	0.3882	0.3794	0.3845
15	0.7474	0.7621	0.7521	0.7628	0.3723	0.3873	0.3784	0.3851

in Table 5. We study how the ensemble size impacts on the performance by setting the number of classifiers as 1, 5, 10 and 15 for the comparison. Following the same approach in the previous part, the presented values in the table are the worst and the best mean values of both accuracy and AUC–ROC scores in conjunction with the corresponding majority voting scores. Similar to the previous experiments, these values are calculated by using the specified number of classifiers over fivefold cross-validation. According to the results, the performance with ensemble size (n) 5, 10 and 15 is better than that of $n = 1$ for single classifier. In the proposed method, we choose ensemble size as 10 to be a standard number of iterations, although beyond the scope of this study, larger ensemble sizes may generate interesting outputs. In addition, using ensemble-based approach for the proposed method provided the majority voted AUC–ROC and accuracy scores greater than the mean AUC–ROC and accuracy scores. Both evaluation metrics increase in general by increasing the ensemble size. Because of using ensemble approach, separate classifiers are assigned to different buckets. It turns out recent time points in the sequence data are considered including less number of padding values as the noise via bucketing and padding approach. The performance becomes relatively stable for the larger ensemble size.

5 Conclusion

CTR estimation is one of the most important steps in real-time bidding for computational advertising. In this paper, we focus on the task to build a new framework for user click response and user interest prediction using LSTM-based deep neural networks. Using padding and bucketing to learn binary user click prediction and multi-class user interest prediction, our method allows sequences to have variable lengths and different number of dimensions and can maximally leverage temporal information in user sequences for learning. Experiments and comparisons on real-world data collected from our industry partner show that our method is able to encode useful latent temporal information in request sequences to predict users' response and interest in online digital advertising.

Acknowledgements This research is sponsored by Bidtellect Inc. and by US National Science Foundation through Grant No. CNS-1828181.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Zhu X, Tao H, Wu Z, Cao J, Kalish K, Kayne J (2017) Fraud prevention in online digital advertising. Springer briefs in computer science. ISBN 978-3-319-56792-1, pp 1–51
2. IAB (2016) POpenRTB API specification version 2.5. <https://www.iab.com/guidelines/real-time-bidding-rtb-project>. Accessed 15 Mar 2019
3. Liu H, Zhu X, Kalish K, Kayne J (2017) ULTR-CTR: fast page grouping using URL truncation for real-time click through rate estimation. In: Proceedings of the of the IEEE international conference on information reuse and integration (IRI), pp 444–451
4. Ormandi R, Yang H, Lu Q (2015) Scalable multidimensional hierarchical Bayesian modeling on spark. In: Wei F, Albert B, Qiang Y, Philip SY (eds) Proceedings of the 4th international conference on big data, streams and heterogeneous source mining: algorithms, systems, programming models and applications (BIGMINE'15), vol 41. JMLR.org, pp 33–48
5. Li C, Lu Y, Mei Q, Wang D, Pandey S (2015) Click-through prediction for advertising in twitter timeline. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM, New York, NY, USA, pp 1959–1968
6. Guo H, Tang R, Ye Y, Li Z, He X (2017) DeepFM: a factorization-machine based neural network for CTR prediction. In: Proceedings of the 26th international joint conference on artificial intelligence, Melbourne, Australia, August 19–25
7. Gharibshah Z, Zhu X, Hainline A, Conway M (2019) Deep learning for online display advertising user clicks and interests prediction. In: Proceedings of the Asia Pacific Web (APWeb) and web-age information management (WAIM) joint conference on web and big data (APWeb-WAIM), pp 196–204
8. Richardson M, Dominowska E, Ragno R (2007) Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on world wide web, ACM, New York, NY, USA, pp 521–530

9. Avila CP, Vijaya MS (2016) Click through rate prediction for display advertisement. *Int J Comput Appl* 136(1):521–530
10. Wang X, Li W, Cui Y, Zhang R, Mao J (2010) Click-through rate estimation for rare events in online advertising. In: Hua XS (ed) *Online multimedia advertising: techniques and technologies*. IGI Global, Hershey, pp 1–12
11. Rendle S (2010) Factorization machines. In: *Proceedings of the 2010 IEEE international conference on data mining (ICDM)*, December 13–17, pp 995–1000
12. Qu Y, Cai H, Ren K, Zhang W, Yu Y, Wen Y, Wang J (2016) Product based neural networks for user response prediction. In: *IEEE 16th international conference on data mining (ICDM)*, pp 1149–1154
13. Liu Q, Yu F, Wu S, Wang L (2015) A convolutional click prediction model. In: *Proceedings of the 24th ACM international conference on information and knowledge management (CIKM)*. ACM, New York, NY, USA, pp 1743–1746
14. Deng W, Ling X, Qi Y, Tan T, Manavoglu E, Zhang Q (2018) Ad click prediction in sequence with long short-term memory networks: an externality-aware model. In: *The 41st international ACM SIGIR conference on research and development in information retrieval (SIGIR '18)*. ACM, New York, NY, USA, pp 1065–1068
15. Chapelle O, Manavoglu E, Rosales R (2014) Simple and scalable response prediction for display advertising. *ACM Trans Intell Syst Technol* 5(4), Article 61
16. Zhang D, Guo L, Nie L, Shao J, Wu S, Shen HT (2017) Targeted advertising in public transportation systems with quantitative evaluation. *ACM Trans Inf Syst* 35(3), Article 20
17. Zhang D, Li Y, Fan J, Gao L, Shen F, Shen HT (2017) Processing long queries against short text: top-k advertisement matching in news stream applications. *ACM TOIS* 35(3):28:1–28:27
18. European Commission (2018) 2018 reform EU data protection rules. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en. Accessed 15 Mar 2019
19. Su N, He J, Liu Y, Zhang M, Ma S (2018) User intent, behaviour, and perceived satisfaction in product search. In: *Proceedings of the eleventh ACM international conference on web search and data mining (WSDM)*. ACM, New York, NY, USA pp 547–555
20. Sheil H, Rana O, Reilly R (2018) Predicting purchasing intent: automatic feature learning using recurrent neural networks. [arXiv:1807.08207](https://arxiv.org/abs/1807.08207)
21. Liu Q, Wu S, Wang L, Tan T (2016) Predicting the next location: a recurrent model with spatial and temporal contexts. In: *Proceedings of the thirtieth AAAI conference on artificial intelligence (AAAI)*. AAAI Press, pp 194–200
22. Zhou G, Mou N, Fan Y, Pi Q, Bian W, Zhou C, Zhu X, Gai K (2019) Deep interest evolution network for click-through rate prediction. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 33(01)
23. Feng Y, Lv F, Shen W, Wang M, Sun F, Zhu Y, Yang K (2019) Deep session interest network for click-through rate prediction. In: Sarit Kraus (ed) *Proceedings of the 28th international joint conference on artificial intelligence (IJCAI'19)*. AAAI Press, pp 2301–2307
24. IAB (2017) Iab tech lab content taxonomy. <https://www.iab.com/guidelines/iab-quality-assurance-guidelines-qag-taxonomy/>. Accessed 15 Mar 2019
25. Agarwal D, Agrawal R, Khanna R, Kota N (2010) Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*. ACM, New York, NY, USA, pp 213–222
26. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. [arXiv:1506.00019](https://arxiv.org/abs/1506.00019)
27. Zhang L, Wang S, Liu B (2018) Deep learning for sentiment analysis : a survey. *Wiley Interdiscip Rev Data Min Knowl Discov* 8:e1253
28. Guo T, Zhu X, Wang Y, Chen F (2019) Discriminative sample generation for deep imbalanced learning. In: *Proceedings of the international joint conference on artificial intelligence (IJCAI)*, pp 2406–2412
29. Cheng H-T, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, Anderson G, Corrado G, Chai W, Ispir M, Anil R, Haque Z, Hong L, Jain V, Liu X, Shah H (2016) Wide and deep learning for recommender systems. In: *Proceedings of the 1st workshop on deep learning for recommender systems (DLRS 2016)*. ACM, New York, NY, USA, pp 7–10
30. Kim Y (2014) Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha, Qatar, pp 1746–1751
31. Zhang Y, Dai H, Xu C, Feng J, Wang T, Bian J, Wang B, Liu T-Y (2014) Sequential click prediction for sponsored search with recurrent neural networks. In: *Proceedings of the twenty-eighth AAAI conference on artificial intelligence (AAAI)*. AAAI Press, pp 1369–1375