Spatial-Temporal Deep Intention Destination Networks for Online Travel Planning

Yu Li, Fei Xiong, Ziyi Wang, Zulong Chen, Chuanfei Xu, Yuyu Yin, and Li Zhou

***Abstract*—Nowadays, artificial neural networks are widely used for users' online travel planning. Personalized travel planning has many real applications and is affected by various factors, such as transportation type, intention destination estimation, budget limit and crowdness prediction. Among those factors, users' intention destination prediction is an essential task in online travel platforms. The reason is that, the user may be interested in the travel plan only when the plan matches his real intention destination. Therefore, in this paper, we focus on predicting users' intention destinations in online travel platforms. In detail, we act as online travel platforms (such as Fliggy and Airbnb) to recommend travel plans for users, and the plan consists of various vacation items including hotel package, scenic packages and so on. Predicting the actual intention destination in travel planning is challenging. Firstly, users' intention destination is highly related to their travel status (e.g., planning for a trip or finishing a trip). Secondly, users' actions (e.g. clicking, searching) over different product types (e.g. train tickets, visa application) have different indications in destination prediction. Thirdly, users may mostly visit the travel platforms just before public holidays, and thus user behaviors in online travel platforms are more sparse, low-frequency and long-period. Therefore, we propose a Deep Multi-Sequences fused neural Networks (DMSN) to predict intention destinations from fused multi-behavior sequences. Real datasets are used to evaluate the performance of our proposed DMSN models. Experimental results indicate that the proposed DMSN models can achieve high intention destination prediction accuracy.**

***Index Terms*—High-order feature interaction, attention mechanism, neural networks, intention prediction, online travel planning.**

I. Introduction

TRAVEL planning has attracted more and more attentions recent years and artificial neural networks are widely used to provide satisfactory plannings. The quality of the recommended travel plan is affected by users' transportation type, intention destination estimation, budget limit and crowdness prediction. Among these factors, user' intention destination estimation is the most important and challenging factor since the user may not be interested in the recommended travel plan unless the plan matches the user's real intention destination.

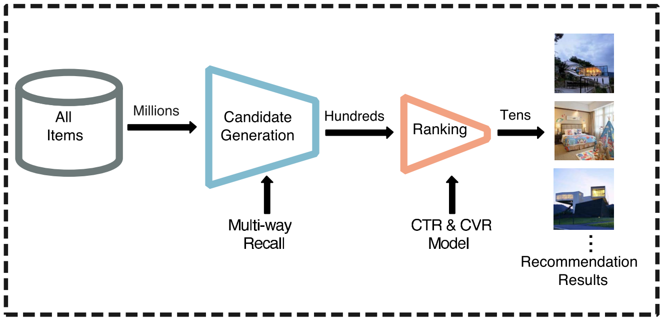


Fig. 1. A general framework of online recommender system.

To enrich users' experiences, intention prediction has been studied in many real e-commerce recommender systems like Taobao, Amazon [1], [2]. Online user intention prediction includes predictions of users' online clicking, searching and purchasing behaviors, etc. Figure 1 illustrates a simple online recommending link in e-commerce recommender systems, and the link consists of two stages: matching and ranking. During the matching stage, the recommender systems will generate hundreds of candidate items from item pool by multi-way recall such as the item-item recall (i2i), destination-item recall (d2i) and so on. During the ranking stage, most recommender systems conduct click-through rate prediction [3]–[7], and some also do a post-click conversion rate prediction [8]–[10].

Online travel platform has a key characteristic that distinguishes it from other e-commerce platforms (e.g. Google, Taobao, YouTube), that is, users' behaviors and intentions are significantly related to *locations*. Users' behaviors are composed by action types and product types, where action types include *click, purchase, collect* and product types include *hotel, train ticket, flight ticket, vacation item, solitary search*. For instance, searching Beijing, booking a hotel in Bangkok, and clicking the visa application in Thailand are users' possible behaviors. Among different product types, *vacation item* is the recommendation target in online travel systems, and vendors in travel platforms provide various categories of

Manuscript received December 5, 2020; revised March 13, 2021 and May 9, 2021; accepted May 28, 2021. Date of publication August 3, 2021; date of current version March 9, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61802098, Grant 61872119, and Grant 61972358; in part by the Natural Science Foundation of Zhejiang Province under Grant LY21F020018; and in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant GK199900299012-025. The Associate Editor for this article was T.-H. Kim. *(Yu Li and Fei Xiong contributed equally to this work.) (Corresponding author: Li Zhou.)*

Yu Li, Yuyu Yin, and Li Zhou are with the Department of Computing, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: zhouli@hdu.edu.cn).

Fei Xiong, Ziyi Wang, and Zulong Chen are with the Alibaba Group, Hangzhou 310000, China.

Chuanfei Xu is with Concordia University, Montreal, QC H3G 1M8, Canada.

Digital Object Identifier 10.1109/TITS.2021.3091534

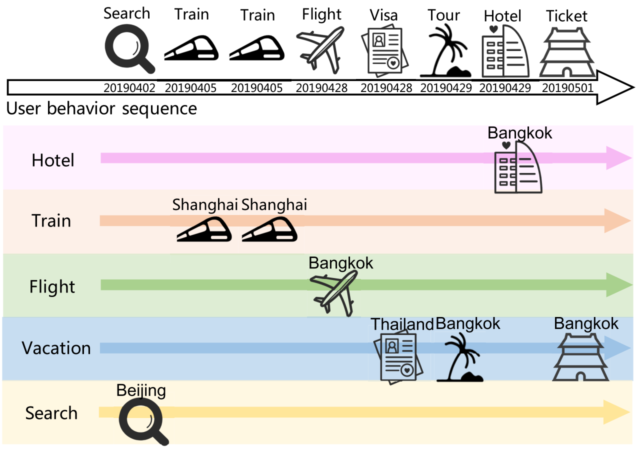


Fig. 2. Example case in travel platforms.

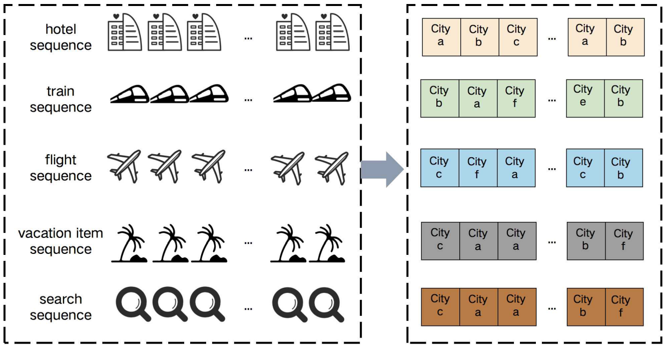


Fig. 3. A mapping from five historical behavior sequences to cityIDs sequences.

vacation items, such as visa applications, scene packages. All products in online travel platforms are associated with locations. When online travel platforms recommend a candidate vacation item to a user, the locations associated to the user's historical behaviors should be taken into consideration. As shown in Figure 2, the user *searched* Beijing on April 2nd 2019, *booked* train tickets for April 5th to Shanghai, after that, he *compared* flight tickets for April 28th to Bangkok, *browsed* hotels and scenic tickets around May 1st in Bangkok. In this example, recommending a hotel package in Bangkok will be much better than a package in Shanghai. The reason is that, according to his historical behaviors, the user only bought tickets for April 5th to Shanghai, then it is possible that he lives in ShangHai and just went back home after a business trip on April 5th. In contrast, the user browsed various products frequently in Bangkok for the period from April 28th to May 1st, and these behaviors indicate that the user may plan a trip to Bangkok during the public holiday around May 1st. Thus, recommending vacation items in Bangkok may match the user better. When recommending travel-related products to a user, we need to prioritize the accuracy of the recommended destination. Inspired by the above observation, we pay more attentions to those geo-features during vacation item recommendations, which has not been considered in most existing online intention prediction models. *In this paper, we focus on predicting users' intention destinations for online travel recommender systems.*

Predicting the actual destination is not easy as users' intention destinations are highly related to their travel status, and user's latest orders may not express his real intention. For instance, the latest purchasing order made by the user in Figure 2 is booking the train tickets to Shanghai, but Shanghai may not be his real intention destination as analyzed above. Moreover, the intersections between users' behaviors may give more precise indications of intention destination. For instance, the user searched/browsed vacation items of different product types in Bangkok in Figure 2 may indicate that he is really interested in Bangkok.

Despite *understanding users' travel status* and *figuring out the intersections between users' behaviors*, another challenge in intention destination prediction is that, users' historical behaviors are quite sparse, low-frequency and long period in online travel platforms. Compared with Google, Youtube and Taobao, users may only visit the online travel platforms during public holidays, and leave the platforms for several months after a trip, which make the behavior data low-frequency and long period. Besides, as categorical features are represented with one-hot encoding, huge number of different products in online travel platforms makes historical behavior features sparse and high-dimensional. Moreover, users' behaviors can infer his *long-term* and *short-term* preferences. For instance, a user may have visited Japan twice a year for several years, inferring that his *long-term* preference is Japan; while, he may search visa application services in Thailand and Phra Phrom recent days, inferring that his *short-term* preference is Thailand. Thus, *how to enrich and emphasize users' features according to their historical behaviors for intention destination prediction* is challenging and urgent.

To predict users' intention destinations over limited historical data, we first propose a framework called Deep Multi-Sequences fused neural Networks (denoted as DMSN) to make fully effective use of user's behaviors over different product types. Considering the all user behaviors on the online travel platform are related to locations, we first map a user's behaviors over different product types into multiple cityID sequences, as depicted in Figure 3. In order to make fully effective use of the correlation between the user's multiple behavior sequences, we fuse and align the user's multiple behavior sequences into a global cityID sequence in the order of behavior time to predict users' real intention destinations. When a user has no behavior on some product types, we will not consider the corresponding behavior types in the fusion sequence. But we will use a default behavior sequence of length one to represent the behavior type in which the user has no behavior if we model multiple behavior sequences separately. Moreover,using cityIDs can reduce the sparsity of categorical behavior features as there are tens of thousands of cities in the whole world and popular tourist cities are only hundreds while there are millions of different products in the platforms.

As users' intention destinations are highly related to their travel status, which may change with time, we use behavior *time* as an indicator to calculate attention weights of different historical behaviors. We intersect Recurrent Neural Network and Multi-grained Convolutional Neural Network into DMSN model, and come up with two models: ATtention based Recurrent Neural Network model (DMSN-ATRNN) and ATtention based Multi-grained Convolutional neural network model (DMSN-ATMC). DMSN-ATRNN can effectively capture the change of users' long-term preference for different destinations, but it is limited on the long training process and poor performance in mining the pattern of short-term behavior. By extracting features with multi-grained convolution neural network, DMSN-ATMC can capture user's short-term and long-term preferences simultaneously and thus can achieve high intention destination prediction accuracy.

Moreover, our proposed DMSN models are generalized models, which can be used in different business applications through merging into different existing intention prediction models (e.g. CTR models DIN [5], AutoInt [6]). When merging the DMSN model into existing CTR prediction models [5], [6], we can achieve: i) users' preference over different cities can be learned from the global behavior features and other auxiliary features and ii) the accuracy of the CTR predictions in online travel platforms can be improved with the use of predicted intention destinations.

In summary, we make following contributions:

- A generalized intention destination prediction framework (i.e., DMSN) for online travel recommender systems is proposed.
- Attention-based neural networks are integrated into DMSN models to effectively capture the change of users' long-term and short-term preferences.
- Comprehensive experiments on real log datasets are conducted to evaluate the effectiveness of proposed models.
- Code and a large scale of high quality log dataset of travel scenarios will be released. The dataset will contribute to the research of personalized recommendation in travel scenarios.

The rest of this paper is organized as follows: Section IV describes the proposed models in detail and Section V illustrates the experimental results. Section VI discusses related literatures and Section VII concludes this paper.

II. Problem Statement and System Architecture

In online travel platforms, online intention prediction aims to predict the probability of a user clicking/searching/purchasing a vacation item according to his historical behavior context. As discussed in Section I, to provide more precise intention predictions over vacation items in online travel platforms, it is important to predict users' intention destination accurately. Therefore, in this paper, we focus on predicting users' intention destinations using users' historical data.

Problem 1: Given a user's historical behavior sequences and a candidate intention destination $dest$, our goal is to predict the probability that the user will be interested in vacation items in $dest$.

We propose Deep Multi-Squences fused neural Network (denoted as DMSN) models (Figures 5 and 6) to predict users' real intention destinations in online travel platforms. Our model follows DIN [5] and shares a similar Embedding&MLP paradigm as most of the popular model structures [4], [5], [11]. We utilize users' historical behaviors to predicate user's preference score over a candidate item city. As users' behaviors in travel platforms are quite sparse, in the *Input Layer*, we map and fuse all five categories of users' historical behaviors into a global cityID sequence. As data features in CTR predictions are mostly sparse and high-dimensional, *Embedding Layer* is necessary to represent those high-dimensional sparse features into low-dimensional spaces. Behaviors related to displayed ads greatly contribute to the click action. A main characteristic of travel ads recommendation is that, users' interests on different cityIDs change with their travel periods. Thus, *Attention-based Neural Network Layer* (i.e., Attention layer and RNN layer in Figures 5; Attention layer and CNN layer in 6) is applied to emphasize the user preference for different destinations during different time. Moreover, to capture users' long-term and short-term preferences for different destinations, we use RNN and multi-grained CNN to express users' diverse interests. Our model also follows DIN [5] to adaptively calculate the representation vectors of user interests by taking into consideration the relevances between historical behaviors and the candidate cityID. That is, in the *MLP layer*, the candidate item cityID is included in a fully connected layer to learn the combination of features automatically. The output layer *Rank Loss Layer* is used to finally calculate the user's preference score for the given candidate item cityID.

Compared with the existing intention prediction models [5], [6], our model has three main differences: i) as in travel scenario, users' behaviors on vacation items are quite sparse, thus in the input layer, we map and fuse all five historical behavior sequences to achieve dense representations; ii) users' interest over a city is highly affected by their travel status, i.e., the time period during their traveling, thus, we add time-factors in our attention functions; iii) users' intention destinations can be different in long-term and short-term, thus we utilize RNN and multi-grained CNN models to better capture users' preferences. Details of our models will be discussed later in Sections IV.

III. System Architecture

This paper objects to predict users' real intention destinations in online travel platforms, and Figure 4 illustrates the overall architecture of our proposed models. Our model follows DIN [5] and shares a similar Embedding&MLP paradigm as most of the popular model structures [4], [5], [11]. We utilize users' historical behaviors to predicate user's preference score over a candidate item city. As users' behaviors in travel platforms are quite sparse, in the *Input Layer*, we map and fuse all five categories of users' historical behaviors into a global cityID sequence. As data features in CTR predictions are mostly sparse and high-dimensional, *Embedding Layer* is necessary to represent those high-dimensional sparse features into low-dimensional spaces. Behaviors related to displayed ads greatly contribute to the click action. A main characteristic of travel ads recommendation is that, users' interests on different

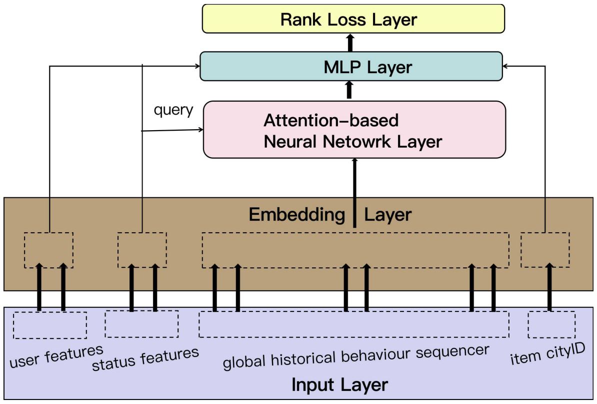


Fig. 4. System architecture.

cityIDs change with their travel periods. Thus, *Attention-based Neural Network Layer* is applied to emphasize the user preference for different destinations during different time. Moreover, to capture users' long-term and short-term preferences for different destinations, we use RNN and multi-grained CNN to express users' diverse interests. Our model also follows DIN [5] to adaptively calculate the representation vectors of user interests by taking into consideration the relevances between historical behaviors and the candidate cityID. That is, in the *MLP layer*, the candidate item cityID is included in a fully connected layer to learn the combination of features automatically. The output layer *Rank Loss Layer* is used to finally calculate the user's preference score for the given candidate item cityID.

Compared with existing CTR prediction models, our model have three main differences: i) as in travel scenario, users' behaviors on vacation items are quite sparse, thus in the input layer, we map and fuse all five historical behavior sequences to achieve dense representations; ii) users' interest over a city is highly affected by their travel status, i.e., the time period during their traveling, thus, we add time-factors in our attention functions; iii) users' intention destinations can be different in long-term and short-term, thus we utilize RNN and multi-grained CNN models to better capture users' preferences. Details of our models will be discussed later in Sections IV.

IV. DMSN MODELS

In this section, we describe our proposed Deep Multi-Squences fused neural Networks (denoted as DMSN) models. Given a user's historical behavior sequences and a candidate intention destination, the goal of DMSN models is to predict the probability that the user will be interested in vacation items in the candidate destination.

A. DMSN Overview

As depicted in Figures 5 and 6, our model follows DIN [5] and shares a similar Embedding&MLP paradigm as most of the popular model structures [4], [5], [11]. We utilize users' historical behaviors to predicate user's preference score over a candidate item city. As users' behaviors in travel platforms are quite sparse, we map and fuse all five categories of users' historical behaviors into a global cityID sequence, and the global cityID sequence is used as the *Input*. As data features in intention destination predictions are mostly sparse and high-dimensional, *Embedding Layer* is necessary to represent those high-dimensional sparse features into low-dimensional spaces. A main characteristic of online travel platforms is that, users' interests on different cityIDs change with their travel periods. Thus, *Attention-based Neural Networks* is applied to emphasize the user preference for different destinations during different time. In detail, to capture users' long and short-term preferences for different destinations, we use RNN and multi-grained CNN to express users' diverse interests (i.e., Attention layer and RNN layer in Figure 5; Attention layer and CNN layer in Figure 6). Our models also adaptively calculate the representation vectors of user interests by taking into consideration the relevances between historical behaviors and the candidate cityID. That is, in the *MLP layer*, the candidate item cityID is included in a fully connected layer to learn the combination of features automatically. The output of DMSN models is the user's preference score for the given candidate item cityID.

Compared with existing intention prediction models [5], [6], DMSN models have three main differences: i) as in travel scenario, users' behaviors on vacation items are quite sparse, thus in the input, we map and fuse all five historical behavior sequences to achieve dense representations; ii) users' interest over a city is highly affected by their travel status, i.e., the time period during their traveling, thus, we add time-factors in our attention functions; iii) users' intention destinations may be different in long-term and short-term, thus RNN and multi-grained CNN are used to better capture users' preferences.

In the following of this section, we discuss the proposed DMSN-ATRNN and DMSN-ATMC models in detail.

B. DMSN-ATRNN Model

In DMSN-ATRNN model, we use recurrent neural network in the Attention-based Neural Network layer to capture the sequential correlation of global behavior sequence and weight historical behaviors. Figure 5 illustrates the architecture of the DMSN-ATRNN model. In the rest of this section, we discuss each layer in DMSN-ATRNN, and we use ATRNN for short.

1) Input Layer: There are four components in the Input Layer.

a) User features: User features contain the basic features of users, including 'user ID', 'user age', 'gender', 'purchase-level' and so on. We denote the set of user features as $\{x_{ui}\}_{i=1}^{n_u}$, where n_u is the number of user features in travel platforms.

b) Status features: User's status information is very important in the travel scenario, for instance, recommending visa application services in the beginning of the trip, scenic packages in the middle of trip and flight tickets in the end of the trip will achieve better click-through rates. We obtain status features from users' itinerary, which contain information like the destination of the trip, the departure time of trip, the completion status of the trip and so on. In addition to

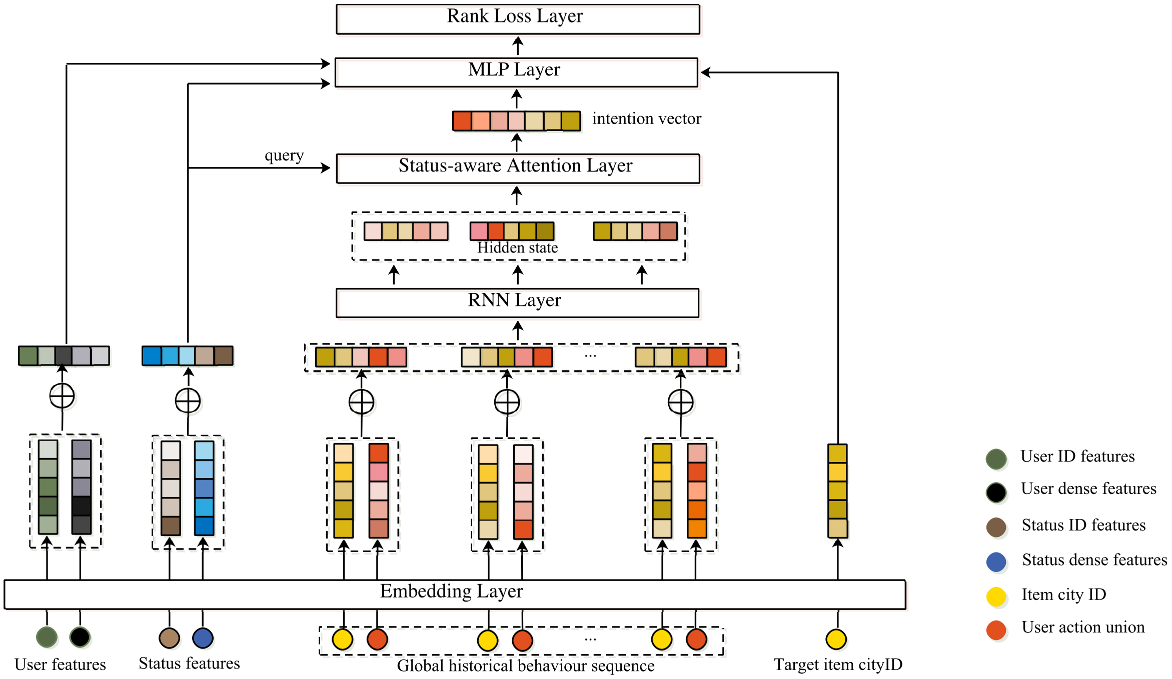


Fig. 5. Architecture overview of DMSN-ATRNN.

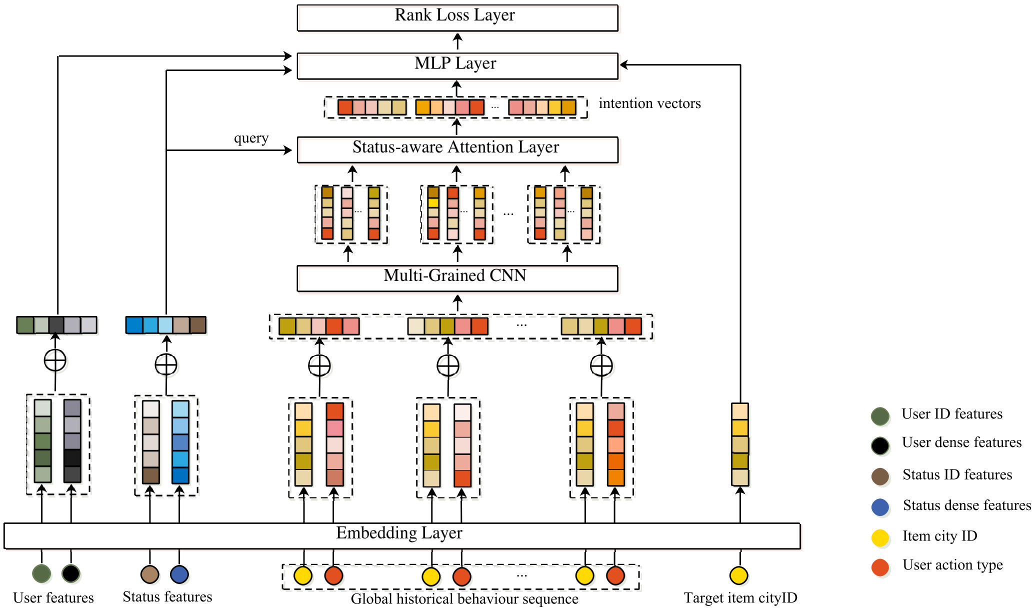


Fig. 6. Architecture overview of ATMC.

user's status information, time-related information is also taken into account, such as the current month, season, hour, etc. We denote the status features obtained from orders as $\{x_{oi}\}_{i=1}^{n_o}$, where n_o is the dimension of status features.

c) Sequential features: As discussed in Figure 3 in Section I, in order to utilize all historical behavior information of a user, we map and fuse different behavior sequences into a global cityID sequence. Each behavior sequence consists of a series of 'action instance', and each 'action instance' contains 'city ID' and 'action unit'. The 'action unit' is a *Cartesian Product* of two sets: product type set P and action type set A, where $P = \{$train ticket, flight ticket, vacation item, hotel, search$\}$ and $A = \{$click, purchase, collect$\}$. Moreover, each 'action unit' will be discretized to ID feature before transferred to the Embedding Layer. The fused global cityID sequence put all 'action instances' in the order of corresponding behavior time. We denote the set of sequential features as $\{x_{si}\}_{i=1}^{n_s}$, where $x_{si} = [x_{cityID}; x_{action\ unit}]_{si}$, and n_s is the length of sequential features. As the sequential features are represented as one-hot encoding and the number of different cities is significantly smaller than that of different products, using cityID instead of specific product can reduce the sparsity of sequential features. Moreover, fusing all behaviors into a single cityID sequence can help relax the problem of low-frequency and long period.

d) Item features: The output of ATRNN model is the predicted preference score of a given candidate city, and thus the candidate 'city ID' is used as the item feature. We denote x_i as the item feature.

2) Embedding Layer: In recommender systems, the input features are sparse and have huge dimension, different from computer vision. An embedding is a mapping of a discrete-categorical-variable to a dense vector of continuous numbers, which has been widely used in Natural Language Processing (NLP) [12] and Recommender System (RS) to alleviate the above phenomenon. In ATRNN, all input features are mapped into the dense vector after embedding, i.e., $\{\mathbf{X}_{ui}\}_{i=1}^{n_u}$, $\{\mathbf{X}_{oi}\}_{i=1}^{n_o}$, $\{\mathbf{X}_{si}\}_{i=1}^{n_s}$ and $\mathbf{X}_i$, which contain richer useful information and yield better generlization.

3) Status-Aware Attention Layer: Attention mechanism [13], [14] is firstly introduced in the encoder-decoder framework for machine translation systems, and it allows the model to emphasize the effect of relevant parts in the input sequence as needed. In this section, we denote t as the current time, t_i as the timestamp i, $\mathbf{X}_o$ as the concatenation vector of all embedding vector of status features, $\bar{\mathbf{h}}_i$ as the hidden state of RNN layer at time i, $\mathbf{W}_a$ and $\mathbf{W}_b$ as the learnable parameters in the attention layer, and set $\tau_i = t_i - t$. In detail, the implementation of attention for sequence-to-one networks is shown in Eq. 1, Eq. 2, Eq. 3 and Eq. 4. Attention weights based on status information are calculated using $\mathbf{X}_o$ as shown in Eq. 1:

$$\alpha_{ti} = \frac{exp(score(\mathbf{X}_o, [\bar{\mathbf{h}}_i, \mathbf{T_i}]))}{\sum_{i'=1}^{T} exp(score(\mathbf{X}_o, [\bar{\mathbf{h}}_i, \mathbf{T_i}]))} \tag{1}$$

where:

$$score(\mathbf{X}_o, [\bar{\mathbf{h}}_i, \mathbf{T_i}]) = \mathbf{X}_o^T \mathbf{W}_a [\bar{\mathbf{h}}_i, \mathbf{T_i}] \tag{2}$$

and

$$\mathbf{T_i} = tanh(\mathbf{W_b} * log(1 + |\tau_i|)) \tag{3}$$

The output of Status-aware Attention Layer is an intention vector $\mathbf{a}_t$ calculated using Eq. 4. $\mathbf{a}_t$ is computed as a weighted sum of the $\{\bar{\mathbf{h}}_i\}_{i=1}^{n_s}$, where the weight assigned to each $\bar{\mathbf{h}}_i$ is computed by a function of the status features $\mathbf{X}_o$.

$$\mathbf{a}_t = \sum_{i=1}^{t} \alpha_{ti} \times \bar{\mathbf{h}}_i \tag{4}$$

4) MLP Layer: The MultiLayer Perceptron (MLP) layer is a feed-forward neural network, which can generalize better to unseen feature combinations through low-dimensional dense embeddings learned for the sparse features [4]. We denote the input of MLP layer as $\mathbf{V}$, the output of MLP layer as $\mathbf{X}_{mlp}$. $\mathbf{V} = [\mathbf{X}_u, \mathbf{X}_o, \mathbf{a}_t, \mathbf{X}_i]$, $\mathbf{X}_u$ is the concatenation vectors of all embedding vectors in $\{\mathbf{X}_{ui}\}_{i=1}^{n_u}$, $\mathbf{X}_o$ is the concatenation vector of all embedding vectors in $\{\mathbf{X}_{oi}\}_{i=1}^{n_o}$ and $\mathbf{a}_t$ is the output of attention layer.

5) Rank Loss Layer: The output of ATRNN is the preferences score of $user_j$ for the item's cityID, in this paper, we model this process as a point-wise ranking problem. We denote $\widehat{y}$ as the output of rank loss layer, meanwhile, $\widehat{y}$ is also the preferences score. $\mathbf{W}_r$ is the learnable parameters in rank loss layer. $\mathbf{X}_{mlp}$ is the output of MLP layer.

$$\widehat{y} = \frac{1}{1 + exp(-\mathbf{W}_r^T \mathbf{X}_{mlp})} \tag{5}$$

y is binary labels with $y = 1$ or $y = 0$ indicating whether click or not. The logistic loss of ATRNN is shown in Eq.(9):

$$L(y, \widehat{y}) = -ylog(\widehat{y}) - (1 - y)log(1 - \widehat{y}) \tag{6}$$

Based on the characteristic of RNN, DMSN-ATRNN can effectively capture the change of users' long-term preference for different destination, and this had been verified in a visualized real example in Figures 8a,b. However, ATRNN still has the following limitations: 1)The users' short-term preference changes can not be well learned; 2) RNN models become difficult to train when the length of sequence is too long.

C. DMSN-ATMC Model

As discussed above, DMSN-ATRNN can capture long-term preference well, but for short-term preferences, it cannot handle well. For instance, if the user has no behaviors related to New York before, and just start to click/purchase tickets/vacation packages in New York during last few minutes. Then, ATRNN may not be able to capture the importance of features related to New York and the short-term preferences will be missed. To solve this problem, we propose the DMSN-ATMC model, which intersects attenion based multi-grained convolutional neural network into the DMSN model. By extracting features with multi-grained convolution kernels, ATMC can capture user's short-term and long-term preferences simultaneously. Compared with DMSN-ATRNN, the ATMC model uses multi-grained CNN layer instead of RNN layer as shown in Figure 6. Different sizes of convolutional kernels are used to capture long-term and short-term preferences.

The multi-grained CNN layer is inspired by the multi-grained scanning in multi-Grained Cascade Forest (gcForest) [15]. Multi-grained CNN layer can capture the feature representation in different time span. Fig.7 shows three sizes of convolution kernels. We denote the dimension of embedding vector as N, the number of kernels of each granularity as m, the number of differently grained convolution kernel as n_c, the shape of sequential features as $N \times n_s$ and the shape of convolution kernel as $N \times k$. By using multiple size of convolution kernels (padding = same), different feature maps are generated. The shape of feature maps is $n_c \times m \times n_s$. It is observed that differently convolution kernels can cover different range of behaviour sequences. Note that the smaller the k, the more attention on user's short-term behaviour, and the larger the k, the more attention on user's long-term behaviour. Therefore, multi-grained convolution kernels can capture user's short-term and long-term preferences simultaneously. We can get n_c feature maps from the output of

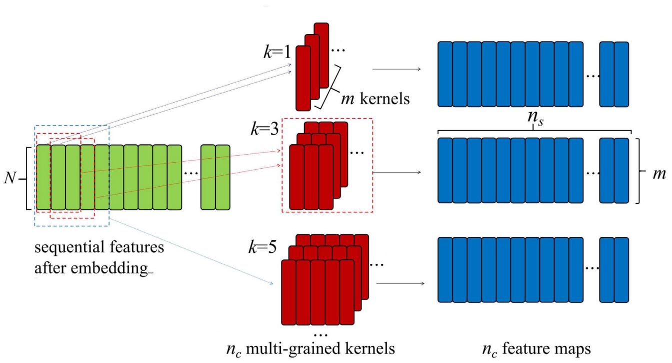

k=1
m kernels
k=3
k=5
N
sequential features after embedding
n_s
m
n_c multi-grained kernels
n_c feature maps

Fig. 7. Architecture overview of multi-grained CNN. Multi-grained convolution kernels are used to extract feature for sequential features and generate multiple feature maps.

multi-grained CNN layer and each feature map is passed into the attention layer independently. The output of attention layer are n_c attention vectors, which are denoted as $\{\mathbf{a}_{ti}\}_{i=1}^{n_c}$. The concatenation of $\{\mathbf{a}_{ti}\}_{i=1}^{n_c}$ as the final attention vector input to the MLP layer.

V. Experimental Evaluation

In this section, experiments are conducted over Fliggy's real data to evaluate the performance of proposed models.

A. Experiment Setup

1) Datasets: In the experiments, we use real data from Fliggy,[1] a well-known online travel recommendation platform. We collect online log data from Fliggy's recommender system in Dec. 2019. Fliggy has four business lines: train tickets booking, flight tickets booking, hotel booking and vacation item purchasing. Relevant vacation items will be recommended to a user after he makes an order. For instance, after a user buys a flight ticket to Bangkok, visa application services in Thailand may be recommended to him.

We collect user behaviors from all business lines, and formulate each action instance as a 2-tuple $[cityID, action\ unit]$ as discussed in Section IV-B.1. For train ticket and flight ticket, we use the arrival city as the *cityID*. For vacation items associated with multiple cities, we select the most hot city as the *cityID*. For hotel and searching, the associated city is used as its *cityID*. These five 'action instance' sequences are fused into a global sequence according to corresponding behavior time. The statistics of five behavior sequences and the global behavior sequence are illustrated in Table I, where length represents the number of action instances in the sequence.

We collect the impression/click data in Fliggy's various recommended scenarios as the label of dataset, the impression and click samples as positive samples, the impression but not click samples as negative samples. Collecting samples exposed by users can effectively reduce noise and ensure users' real interest preferences which is commonly used method of sample collection in real industrial scenes. The statistics of dataset is illustrated in Table II.

TABLE I
Statistics of Five Behaviors Sequences

Sequence	Maximum length	Minimum length	Average length
train	103	0	13.5
flight	156	0	15.9
hotel	257	0	24.9
item	248	0	18.9
search	66	0	6.3
global	319	1	28.6

TABLE II
Statistics of Fliggy's Dataset (M - Million)

	Training	Validation	Testing
# of samples	26.4M	3.99M	4.03M
# of positive samples	1.86M	0.273M	0.274M
# of negative samples	24.5M	3.72M	3.75M
# of users	2.53M	0.478M	0.486M
# of vacation items	0.254M	0.185M	0.173M
# of destinations	0.004M	0.004M	0.004M

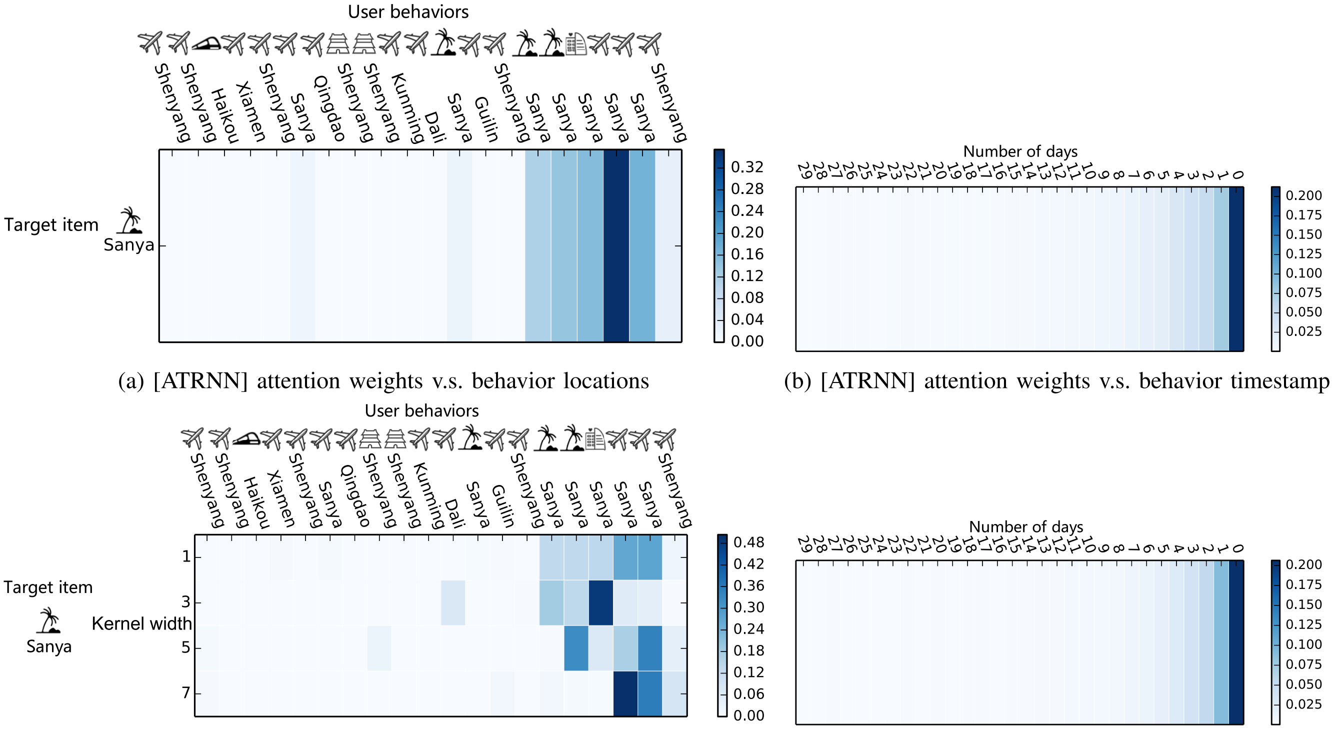
2) Evaluation Metric: In order to evaluate the performance of proposed methods, we adopt the Area Under the ROC Curve (AUC) as the evaluation metric. AUC is not sensitive to class imbalance and thus is widely used in online recommender systems. It reflects the probability that a model ranks a randomly chosen positive sample higher than negative samples and larger AUC represents better performance. Exhaustive experiments show that a small improvement in AUC can lead to a significant increase in the online intention prediction accuracy [4].

3) Competing Models: We focus on intention destination prediction in this paper, and intention destinations are important for online intention predictions in recommender systems. Thus, we evaluate the accuracy of both intention destination prediction and vacation item prediction.

For intention destination prediction, although existing previous intention prediction models focus on recommending vacation items, they can be easily adapted to predict intention destination through changing the targets to cityIDs. Thus, we compare the following models in terms of intention destination prediction:

- *AutoInt* [6] is a state-of-the-art Automatic Feature Interaction neural networks.
- *DIN* [5] is a Deep Interest Network, which is the latest online serving model in Fliggy.
- *DMSN-MLP*: In our proposed Deep Multi-Sequence fused neural Networks (i.e., DMSN), five behavior sequences are fused into a global sequence, which is used with other auxiliary features to predict the user preference score for a destination. DMSN-MLP implemented the DMSN structure that includes the fully-connected layer and excludes the attention layer.
- *DMSN-ATRNN*: As shown in Figure 5, attention based recurrent neural network is implemented in DMSN-ATRNN. The RNN layer is implemented by GRU.
- *DMSN-ATMC*: As shown in Figure 6, attention based multi-grained CNN is implemented in DMSN-ATMC.

[1] https://www.fliggy.com/



(a) [ATRNN] attention weights v.s. behavior locations

(b) [ATRNN] attention weights v.s. behavior timestamp

(c) [ATMC] attention weights v.s. behavior locations

(d) [ATMC] attention weights v.s. behavior timestamp

Fig. 8. Visualized case study for time-aided attention weights of behaviors.

For vacation item prediction, we merge the proposed intention destination prediction models into existing vacation item recommender systems, and the details will be discussed in Section V-D.

4) Implementation Details: We set the maximum length of global sequence to 100. If the sequence length is less than 100, it will be padded with 0. If the sequence length is more than 100, the last 100 ‘action instance’ will be truncated. We implement our method using Tensorflow. All hyper-parameters of all models are adjusted by grid-searching on the validation set. The dimension of embedding for all models is set to 32. The MLP contains four hidden layer, with dimensions 512, 256, 128 and 1. The hidden state size of GRU is set to 32. We use the Adam with a mini-batch size of 512.

B. Visualized Case Study

To obtain the visualized results, we first train the *DMSN-ATRNN* and *DMSN-ATMC* models, and then choose one user who has acted on a scenic package in Sanya to show the effectiveness of time-aided attention functions. As discussed in Section IV, in *DMSN-ATRNN*, a time-aided attention weight vector is calculated for the hidden state according to Equation 1, while one attention weight vector is computed for each of n_c feature maps in *DMSN-ATMC*. The x-axis in Figure 8a are user behaviors following the time order from left to right, and x-axis in Figure 8b illustrate how many days ago the behavior was conducted. The color of each bar represents the attention weight of its corresponding behavior, and the weight is calculated by Equation 1. The darker the color, the higher the weight. As depicted in Figure 8a, with our *DMSN-ATRNN* model, behaviors in Sanya obtains much higher attention weights than behaviors in other cities, which matches the actual intention destination (i.e., Sanya) of the user. The result confirms that our proposed attention model can figure out user’s actual intention destinations. Moreover, the time variable used in Equation 1 enables the ATRNN model to capture the effect of time in city prediction. In detail, as illustrated in Figure 8b, behaviors conducted on the latest day (e.g. searching vacation items and hotels in Sanya, clicking flight tickets from Shenyang) are marked to be more important during user intention prediction than behaviors done a month ago. The result confirms that our proposed attention model can emphasize the time relevance between historical behaviors and users’ intentions.

Similar to the results of *DMSN-ATRNN*, Figures 8c,d depict the visualized results of *DMSN-ATMC*. As discussed in Section IV-C, n_c feature maps will be passed to the attention layer in *DMSN-ATMC*, where n_c represents the number of different kernels in Multi-Grained CNN Layer. And for each feature map, one attention vector is calculated. And as shown in Figure 8c, we use 4 kernels with width $k = 1, 3, 5, 7$, and for each kernel width, one feature map is passed to the attention layer, and thus one attention vector is obtained. Compared with *DMSN-ATRNN*, *DMSN-ATMC* can assign a higher attention weight to more relevant behaviors, for instance the highest weight is 0.48 in Figure 8c while the highest is 0.32 in Figure 8a. Moreover, *DMSN-ATMC* can explore more relations on product types than *DMSN-ATRNN*. For instance, when $k = 1$, behaviors on flight in Sanya are marked as more important, while when $k = 3$, behaviors on hotels in Sanya

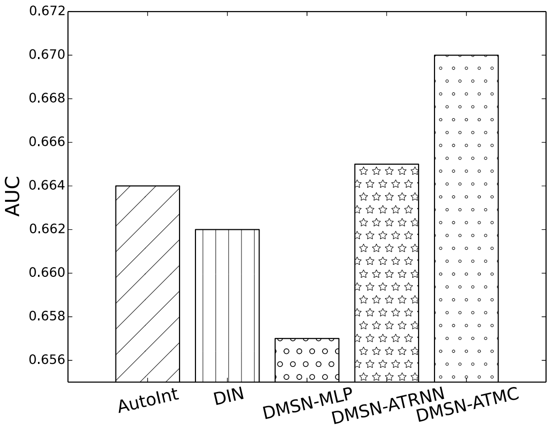


Fig. 9. Comparison of proposed models for user's preferred destinations prediction.

are more important. Because in the approaching time of the hotel behavior, the user also acted on vacation item and flight ticket under the same destination. That is, with different kernel granularities, *DMSN-ATMC* can capture important features in combination of different product types under the same destination and thus provide more comprehensive and rich data to the MLP layer for accurate prediction. Besides, similar to *DMSN-ATRNN*, *DMSN-ATMC* is also sensitive to the conducted time of historical behaviors as shown in Figure 8d.

C. Intention Destination Prediction

We evaluate the prediction performance over intention destinations.

1) Effect of Proposed Models: Figure 9 compares the performance over intention destination prediction. It is observed that our proposed *DMSN-ATRNN* and *DMSN-ATMC* perform better than state-of-the-art prediction models. Besides, *DMSN-ATMC* achieves the highest AUC among all competitors. The reason is that, differently grained convolution kernels can capture the change pattern of users' intention in the long and short term effectively, and can fully capture important features in combination of different product types under the same destination.

2) Effect of Kernel Number in DMSN-ATMC: The number of multi-grained convolution kernels is an important parameter in multi-grained CNN layer, and thus we test the performance over different number of kernels as shown in Figure 10. We can see that, in the beginning, larger number of kernels induces higher prediction AUC, however, overfitting occurs when there are too many kernels and thus the prediction AUC drops. Moreover, *DMSN-MC* represents the model without Attention Layer in *DMSN-ATMC*, and from the result, we can see that the attention layer is effective and necessary.

3) Effect of Hyper-Parameters in DMSN-ATMC: We study the impact of hyper-parameters of *DMSN-ATMC* including the activation functions, the number of neurons in hidden layer and the number of hidden layers. We compare the results of different activation functions in Figure 11a and relu is most suitable for *DMSN-ATMC*. Moreover, Figure 11b demonstrates the impact of number of neurons in hidden layer. When we increase the number of neurons in a fixed setting 4 for the number of hidden layers, 128 is the best setting for *DMSN-ATMC*. We can also observe that increasing the number of hidden layers from 2 to 4 can improve the performance. But model performance degrades when the number of hidden layers is set greater than 4 caused by overfitting.

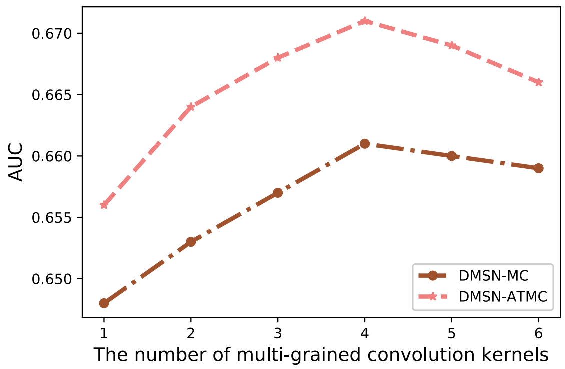


Fig. 10. Effect of the number of multi-grained convolution kernels.

TABLE III
EFFECT OF FEATURE FUSING

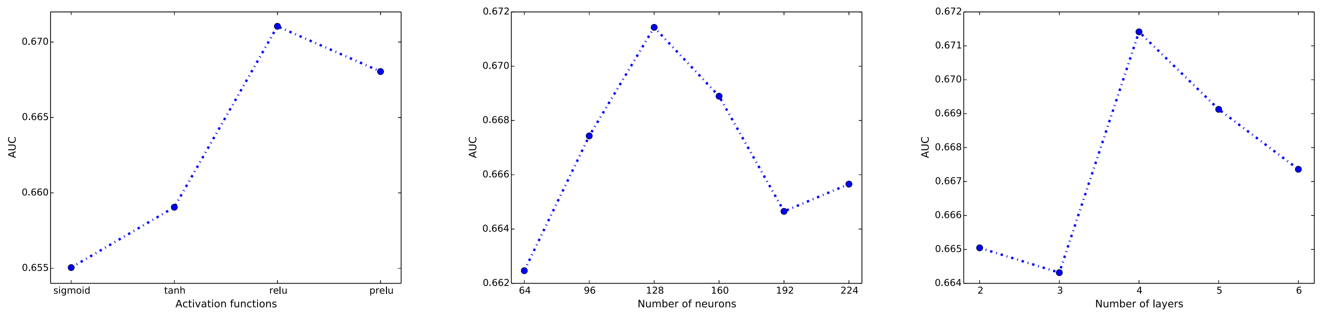
Fusing Strategy	AUC
Hotel Single	0.657
Train Single	0.66
Flight Single	0.658
Item Single	0.662
Search Single	0.654
Fusing Strategy I	**0.671**
Fusing Strategy II	0.665

4) Effect of Fusing Strategies: In order to verify the effects of different behavioral sequence fusion strategies, we test whether such fusing is helpful in *DMSN-ATMC* by using only one type of sequence, and compare the mentioned two fusing strategies, i.e., **strategy I**: fusing in the very beginning as a global sequence v.s. **strategy II**: fusing after attention layer as five single sequences. In order to align the input format, we use a default behavior sequence of length one to represent the behavior type in which the user has no behavior if we model multiple behavior sequences separately as in **strategy II**. As illustrated in Table III, using fused global behavior sequence achieves better performance than using independent feature sequences. And fusing in the very beginning provides the multi-grained CNN layer and attention layer more dense information, thus strategy I outperforms strategy II.

5) Effect of Sample Size: Figure 12 depicts the effect of different size of training sets. We compare all competitors in this test and *DMSN-ATMC* outperforms other models under all sampling rates. With multi-grained CNN layer, *DMSN-ATMC* can predict intention destinations mode accurately even with small size of samples.

D. Vacation Item Prediction

Intention destination predictions are essential for vacation item prediction in online travel platforms. Thus, we also



(a) Activation functions (b) Number of neurons (c) Number of layers

Fig. 11. Impact of network hyper-parameters on AUC performance.

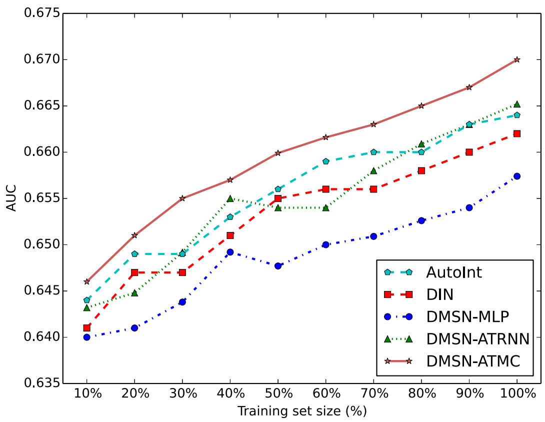


Fig. 12. Effect of sampling rates on Fliggy's dataset.

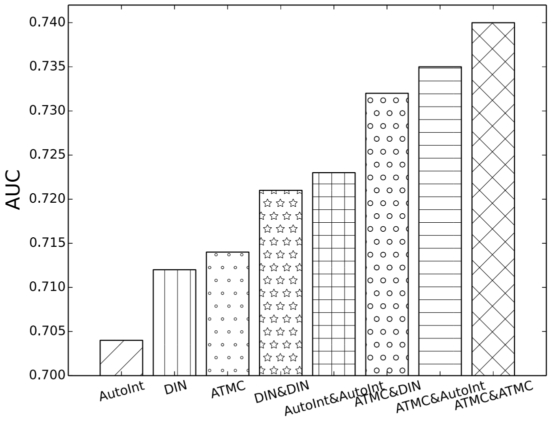


Fig. 13. Comparison of different models on Fliggy's Dataset in terms of vacation item prediction.

evaluate the effectiveness of our proposed models in terms of vacation item predictions. To do this, we use a two-stage method to merge predicted intention destinations into vacation item predictions. Specifically, we first calculate the preference scores of different destinations, and then estimate the preference scores of vacation items. In detail, the final preference score of $user_j$ to vacation item $item_i$ is calculated as:

$$\begin{aligned} & p(item_i|context^j_{global}) \\ & \quad = p(dest_k|context^{global}_j) \times p(item_i|context^{item}_j) \\ & \quad = Score_{Model1} \times Score_{Model2} \end{aligned}$$

where $context^{global}_j$ is the global context information of $user_j$ in Fliggy, $p(dest_k|context^{global}_j)$ is the probability that $user_j$ will goto destination $dest_k$ according to his historical behaviors in $context^{global}_j$, and $p(item_i|context^{item}_j)$ is the probability that $user_j$ will click $item_i$.

Moreover, both $p(item_i|context^{item}_j)$ and $p(dest_k|context^{global}_j)$ can be obtained by any of the tested five competitor models. As *DMSN-ATMC* always performs the best in proposed *DMSN* models, we exclude *DMSN-ATRNN* and *DMSN-MLP* in the following discussions.

Figure 13 illustrates the prediction results over vacation items. The first three bars using *AutoInt*, *DIN* and *DMSN-ATMC* to predict vacation items directly without predicting destinations first. The remaining bars using 2-stage procedures, and *Model1&Model2* means we use *Model1* to predict $p(dest_k|context^{global}_j)$ and *Model2* to predict $p(item_i|context^{item}_j)$. We can see that, 2-stage methods result in higher vacation item prediction accuracy, and this verifies the importance of intention destination predictions. Moreover, ATMC&ATMC performs the best, indicating that *DMSN* models can effectively improve the ranking results.

TABLE IV
AVERAGE RELATIVE IMPROVEMENT RESULTS OF CTR

Business line	CTR improvement
Train	+3.4%
Flight	+2.5%
Vacation items	+1.4%
Hotel	+1.4%

E. Online Vacation Item Recommendation

We deploy our model on Fliggy's online recommender platform, and conduct experiments on standard A/B testing environment. We use the real-time click-through rate (CTR) as the metric to evaluate the performance of our proposed ATMC&ATMC model. We ran the experiment for one week on each business line in Fliggy. As shown in Table IV, our proposed ATMC&ATMC achieves significant CTR improvement on four business lines, and achieves the highest CTR improvement on train ticket business line.

VI. Related Work

This paper studies how to provide satisfactory travel plannings for users, especially we focus on predicting users' real intention destinations in traveling scenario. Intention prediction has attracted attentions from both industry and academic communities. As many internet companies utilize click-through rates to predict users' preferences, various prediction systems have been developed for online intention predictions [4], [5], [16]–[19]. Wide&Deep learning system [4] proposed by Google combines the advantages of both the linear shallow model and deep models for recommender systems. Wide&Deep model achieves remarkable performance in APP recommendation. DIN [5] model proposed by Alibaba group is designed for online advertising. Different from other click-through rate models which compress user features into a fixed-length representation vector, DIN designs a local activation unit to adaptively learn the representation of user interests from historical behaviors with respect to a certain ad. Besides industry communities, intention prediction is also well studied in academic communities [20], [21]. A context aware click-through rate prediction method [20] is proposed with factorized three-way $\langle user, ad, context\rangle$ tensor. Hierarchical importance-aware factorization machine [21] is developed to model dynamic impacts of ads prediction. The structure of intention prediction model has evolved from shallow to deep. And the number of samples and the dimension of features used in intention prediction model have become larger and larger. Thus, more and more model structures are designed to improve the feature extraction performance [4], [5], [11], [22]–[26]. The widely used base click-through rate prediction model is a combination of embedding layer (for learning the dense representation of sparse features) and MLP (for learning the combination relations of features automatically). The embedding method is pioneered discussed in NNLM [22], which learns distributed representation for each word to avoid curse of dimension in language modeling. Dense representations of features after embedding layer are interacted using specially designed transformation functions for target fitting LS-PLM [23] and FM [24] models are a class of networks with one hidden layer, which captures the first- and second-order feature interactions in recommender systems Variants of factorization machines include Field-aware Factorization Machines (FFM) [27],GBFM [28] and AFM [29]. FFM models fine-grained interactions between features from different fields, GBFM and AFM considered the importance of different second-order feature interactions. All these base models only focus on low-order feature interactions. In terms of high-order feature interactions, NFM [25] stacked deep neural networks on top of the output of the second-order feature interactions to model higher-order features. Moreover, Deep Crossing [11], Wide&Deep Learning [4] and YouTube Recommendation click-through rate model [26] extend LS-PLM and FM by replacing the transformation function with complex MLP network, which enhances the model capability greatly. PNN [30] tries to capture high-order feature interactions by involving a product layer after embedding layer. DeepFM [3] imposes a factorization machines as "wide" module in Wide&Deep with no need of feature engineering. Deep&Cross [31] and xDeepFM [32] took outer product of features at the bit- and vector-wise level respectively to learn feature interactions. All above approaches are not trivial to explain which combinations are useful, thus the work [6] explicitly models feature interactions with attention mechanism in an end-to-end manner, and probe the learned feature combinations via visualization. For applications with rich user behaviors, features are often contained with variable-length list of ids, e.g., searched terms or watched videos in YouTube recommender systems. Above works often transform corresponding list of embedding vectors into a fixed-length vector, which may cause loss of information. DIN [5] tackles it by adaptively learning the representation vector w.r.t. given ad, improving the expressive ability of model. The proposed DMSN models in this paper use the latest technique in the literature of deep learning which is attention mechanism [33]. Aided with attention-based neural networks, the DMSN models can get an expected annotation and focuses only on information relevant to the generation of next target word. Attention is first proposed in the context of neural machine translation [33] and has been proved effective in a variety of tasks such as neural machine translation and recommender systems. Neural Machine Translation (NMT) [33] takes a weighted sum of all the annotations to get an expected annotation and focuses only on information relevant to the generation of next target word. DeepIntent [34] applies attention in the context of search advertising and it uses RNN to model text to help paying attention on the key words in each query. DIN [5] designs a local activation unit to soft-search for relevant user behaviors and takes a weighted sum pooling to obtain the adaptive representation of user interests with respect to a given ad. Compared with existing works, this paper exploits the characteristics of online travel platforms and propose to integrate attention-based neural networks into multi-sequence fused frameworks for users' intention destination prediction. Moreover, there exist related literatures studying destination predictions [35]–[38]. [35] utilizes a hidden Markov model for predicting driver destinations and routes. [36] proposes a hybrid model to predict future gathering events through trajectory destination prediction. [37] studies a novel data embedding method and ensemble learning method to provide accurate and timely destination predictions of taxis. [38] utilizes trust-enhanced collaborative filtering to predict users' intention POIs according to their historical trajectory behaviors. However, none of existing methods consider to use users' multi-category behaviors for online travelling destination prediction.

VII. Conclusion

In this paper, we focus on predicting users' intention destinations during travel plannings in online travel platforms. Accurate predicted destinations can significantly improve the vacation item recommendation performance. In detail, we propose two models to predict users' real intention destinations. Both models follow the Deep Multi-Squences fused neural Networks (DMSN) architecture, which fuses different categories of user behavior sequences into a global cityID sequence and use the global sequence to predict destination preferences. To emphasize users' preference for different destinations during different periods, we intersect the DMSN model with attention-based recurrent neural networks (ATRNN) and attention-based multi-grained convolutional neural networks(ATMC). Experimental results on real data from Fliggy's offline log and online A/B testing illustrate the effectiveness of our DMSN models.

References

[1] H. Zhu *et al.*, "Learning tree-based deep model for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1079–1088, doi: 10.1145/3219819.3219826.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[3] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1725–1731, doi: 10.24963/ijcai.2017/239.

[4] H. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10, doi: 10.1145/2988450.2988454.

[5] G. Zhou *et al.*, "Deep interest network for click-through rate prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1059–1068.

[6] W. Song *et al.*, "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage., (CIKM)*, Beijing, China, Nov. 2019, pp. 1161–1170, doi: 10.1145/3357384.3357925.

[7] W. Ouyang *et al.*, "Deep spatio-temporal neural networks for click-through rate prediction," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2078–2086.

[8] X. Ma *et al.*, "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 1137–1140, doi: 10.1145/3209978.3210104.

[9] H. Wen, J. Zhang, Q. Lin, K. Yang, and P. Huang, "Multi-level deep cascade trees for conversion rate prediction in recommendation system," in *Proc. 33rd AAAI Conf. Artif. Intell.*, vol. 33, pp. 338–345, Jul. 2019, doi: 10.1609/aaai.v33i01.3301338.

[10] L. Shan, L. Lin, and C. Sun, "Combined regression and tripletwise learning for conversion rate prediction in real-time bidding advertising," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 115–123, doi: 10.1145/3209978.3210062.

[11] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. C. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds. San Francisco, CA, USA, Aug. 2016, pp. 255–262, doi: 10.1145/2939672.2939704.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst., 27th Annu. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119. [Online]. Available: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

[13] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–21. [Online]. Available: https://openreview.net/forum?id=HkE0Nvqlg

[14] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008. [Online]. Available: https://papers.nips.cc/paper/7181-attention-is-all-you-need

[15] Z. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3553–3559, doi: 10.24963/ijcai.2017/497.

[16] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, J. Fürnkranz and T. Joachims, Eds. Haifa, Israel: Omnipress, Jun. 2010, pp. 13–20. [Online]. Available: https://icml.cc/Conferences/2010/papers/901.pdf

[17] X. He *et al.*, "Practical lessons from predicting clicks on ads at facebook," in *Proc. 8th Int. Workshop Data Mining Online Advertising (ADKDD)*, E. Saka, D. Shen, K. Lee, and Y. Li, Eds. New York, NY, USA, Aug. 2014, pp. 5:1–5:9. [Online]. Available: https://doi.org/10.1145/2648584.2648589

[18] H. B. McMahan *et al.*, "Ad click prediction: A view from the trenches," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1222–1230, doi: 10.1145/2487575.2488200.

[19] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: Estimating the click-through rate for new ads," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, Eds. Banff, AB, Canada, May 2007, pp. 521–530, doi: 10.1145/1242572.1242643.

[20] L. Shan, L. Lin, C. Sun, and X. Wang, "Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization," *Electron. Commerce Res. Appl.*, vol. 16, pp. 30–42, Mar. 2016, doi: 10.1016/j.elerap.2016.01.004.

[21] R. J. Oentaryo, E. Lim, J. Low, D. Lo, and M. Finegold, "Predicting response in mobile advertising with hierarchical importance-aware factorization machine," in *Proc. 7th ACM Int. Conf. Web Search Data Mining (WSDM)*, B. Carterette, F. Diaz, C. Castillo, and D. Metzler, Eds. New York, NY, USA, Feb. 2014, pp. 123–132, doi: 10.1145/2556195.2556240.

[22] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003. [Online]. Available: https://jmlr.org/papers/v3/bengio03a.html

[23] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piece-wise linear models from large scale data for ad click prediction," 2017, *arXiv:1704.05194*. [Online]. Available: https://arxiv.org/abs/1704.05194

[24] S. Rendle, "Factorization machines," in *Proc. 10th IEEE Int. Conf. Data Mining*, G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, Eds. Sydney, NSW, Australia: IEEE Computer Society, Dec. 2010, pp. 995–1000, doi: 10.1109/ICDM.2010.127.

[25] X. He and T. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, N. Kando, T. Sakai, H. Joho, H. Li, A. P. de Vries, and R. W. White, Eds. Shinjuku, Tokyo, Japan, Aug. 2017, pp. 355–364. [Online]. Available: https://doi.org/10.1145/3077136.3080777

[26] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, S. Sen, W. Geyer, J. Freyne, and P. Castells, Eds. Boston, MA, USA, Sep. 2016, pp. 191–198, doi: 10.1145/2959100.2959190.

[27] Y. Juan, Y. Zhuang, W. Chin, and C. Lin, "Field-aware factorization machines for CTR prediction," in *Proc. 10th ACM Conf. Recommender Syst.*, S. Sen, W. Geyer, J. Freyne, and P. Castells, Eds. Boston, MA, USA, Sep. 2016, pp. 43–50, doi: 10.1145/2959100.2959134.

[28] C. Cheng, F. Xia, T. Zhang, I. King, and M. R. Lyu, "Gradient boosting factorization machines," in *Proc. 8th ACM Conf. Recommender Syst.*, A. Kobsa, M. X. Zhou, M. Ester, and Y. Koren, Eds. Foster City, CA, USA, Oct. 2014, pp. 265–272, doi: 10.1145/2645710.2645730.

[29] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, C. Sierra, Ed. Melbourne, VIC, Australia, Aug. 2017, pp. 3119–3125, doi: 10.24963/ijcai.2017/435.

[30] Y. Qu *et al.*, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu, Eds. Barcelona, Spain: IEEE Computer Society, Dec. 2016, pp. 1149–1154, doi: 10.1109/ICDM.2016.0151.

[31] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proc. ADKDD*, Halifax, NS, Canada, Aug. 2017, pp. 12:1–12:7, doi: 10.1145/3124749.3124754.

[32] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Y. Guo and F. Farooq, Eds. London, U.K., Aug. 2018, pp. 1754–1763, doi: 10.1145/3219819.3220023.

[33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473

[34] S. Zhai, K. Chang, R. Zhang, and Z. M. Zhang, "Deepintent: Learning attentions for online advertising with recurrent neural networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds. San Francisco, CA, USA, Aug. 2016, pp. 1295–1304, doi: 10.1145/2939672.2939759.

[35] Y. Lassoued, J. Monteil, Y. Gu, G. Russo, R. Shorten, and M. Mevissen, "A hidden Markov model for route and destination prediction," in *Proc. 20th IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 1–6.

[36] A. V. Khezerlou, X. Zhou, L. Tong, Y. Li, and J. Luo, "Forecasting gathering events through trajectory destination prediction: A dynamic hybrid model," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 991–1004, Mar. 2021.

[37] X. Zhang, Z. Zhao, Y. Zheng, and J. Li, "Prediction of taxi destinations using a novel data embedding method and ensemble learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 68–78, Jan. 2020.

[38] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. Gong, "Trust-enhanced collaborative filtering for personalized point of interests recommendation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6124–6132, Sep. 2020.

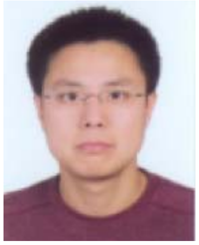
Yu Li received the Ph.D. degree from The Hong Kong Polytechnic University in 2015. She worked as a Post-Doctoral Researcher with The Hong Kong Polytechnic University. She is currently working with Hangzhou Dianzi University, China. Her research interests include crowdsourcing, spatial recommendation, database optimization, and cloud computing.

Fei Xiong received the B.E. degree in automation from Wuhan University, Wuhan, China, in 2016, and the M.E. degree in control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019. He is currently working as a Senior Algorithm Engineer with the Alibaba Group, participating in the development of recommendation system and search engine. His research interests include data mining, machine learning, and applications in artificial intelligence.

Ziyi Wang received the M.E. degree in computer science and technology from Nanjing University in June 2019. He is currently working with the Alibaba Group. His current research interest includes personalized ranking algorithm in e-commerce (travel scenario) search and recommendation.

Zulong Chen received the M.E. degree in information engineering from Northeastern University. He worked with Baidu in 2013. In 2014, he joined the Alibaba Group, working on personalized click-through rate estimation and crowd mining algorithms. He is currently working with Fliggy, focusing on search recommendation and advertising. He has published articles in KDD, CIKM, and WWW.

Chuanfei Xu received the B.E. degree from the Shenyang University of Technology in 2007 and the M.E. and Ph.D. degrees in computer science from Northeastern University, in 2009 and 2013, respectively. He is currently a Post-Doctoral Research Fellow with Concordia University, Canada. His research interests include data mining, NLP, and uncertain data management.



Yuyu Yin received the Ph.D. degree in computer science from Zhejiang University in 2010. He is currently a Professor with the College of Computer, Hangzhou Dianzi University. He is also a Supervisor of master's students with the School of Computer Engineering and Science, Shanghai University, Shanghai, China. He has published more than 40 articles in journals and refereed conference papers, such as *Sensors*, *Entropy*, *International Journal of Software Engineering* and *Knowledge Engineering*, *Mobile Information Systems*, ICWS, and SEKE. His research interests include service computing, cloud computing, and business process management. He is a member of the China Computer Federation (CCF) and the CCF Service Computing Technical Committee. He has organized more than ten international conferences and workshops, such as FMSC 2011/2017 and DISA 2012 and 2017/2018. He has served as a Guest Editor for the *Journal of Information Science and Engineering* and *International Journal of Software Engineering and Knowledge Engineering* and a Reviewer for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Journal of Database Management*, and *Future Generation Computer Systems*.

Li Zhou received the master's degree from Hangzhou Dianzi University. She is currently working as an Associate Professor with Hangzhou Dianzi University. She has hosted the Natural Science Foundation of Zhejiang Province, the sub-projects of the National Natural Fund, and the sub-projects of Zhejiang Province's Key Research and Development projects. She has published more than 30 articles indexed by SCI/EI.