Sequential Recommendation Based on Multivariate Hawkes Process Embedding with Attention

Dongjing Wang *Member, IEEE,* Xin Zhang*, Zhengzhe Xiang, Dongjin Yu *Senior Member, IEEE,*
Guandong Xu *Member, IEEE,* and Shuiguang Deng *Senior Member, IEEE*

***Abstract*—Recommender systems are important approaches for dealing with the information overload problem in big data era, and various kinds of auxiliary information, including time and sequential information, can help to improve the performance of retrieval and recommendation tasks. However, it is still a challenging problem how to fully exploit such information to achieve high-quality recommendation results and improve users' experience. In this work, we present a novel sequential recommendation model named Multivariate Hawkes Process Embedding with attention (MHPE-a), which combines a temporal point process with attention mechanism to predict the items that the target user may interact with according to her/his historical records. Specifically, the proposed approach MHPE-a can model users' sequential patterns in their temporal interaction sequences accurately with a multivariate Hawkes process. Then, we perform accurate sequential recommendation to satisfy target users' real-time requirement based on their preferences obtained with MHPE-a from their historical records. Especially, an attention mechanism is used to leverage users' long/short-term preferences adaptively to achieve accurate sequential recommendation. Extensive experiments are conducted on two real-world datasets (lastfm and gowalla), and the results show that MHPE-a achieves better performance than state-of-the-art baselines.**

***Index Terms*—recommender system, sequential recommendation, multivariate Hawkes process, attention, embedding**

I. Introduction

NOWADAYS, the prevalence of Information Technology (IT) promotes the rapid growth in digital services and contents available on the Internet, thereby causing an information overload problem. For example, Apple iTunes provides over 70 million songs (https://www.apple.com/my/apple-music/). As a result, it becomes more and more difficult for users to obtain digital contents or services they actually need. As one important solution for information overload problem, recommender systems [1], [2] can reduce search cost effectively and offer users personalized contents or services from enormous amounts of available data. Generally, traditional recommendation methods include collaborative filtering, content-based, context-aware and hybrid models [3]. All kinds of recommendation methods are widely applied in many fields, such as music recommendation [4], point of interests (POI) recommendation [5], [6], groups recommendation [7], [8], business process recommendation [9], E-Commerce [10] and so on. Most traditional recommendation techniques perform recommendation based on users' general interest, which represents users' long-term static preferences. For example, some certain users may prefer pop music to other genres in most situations.

In fact, users' preferences may change dynamically over time, and the next behavior or item generally depends on their recent behaviors. Therefore, the sequential patterns are quite important in capturing their requirement and improving the performance of recommendation. In this case, sequential patterns represent a user's short-term dynamic preference. For example, a certain user may enjoy rock songs when doing exercise, though he/she likes pop music better in general. In this case, rock music is the user's short-term preference while pop music is her/his long-term preference. Sequential recommender systems [11] can model users' sequential patterns and predict their next action/item based on their long/short-term preferences, which can be obtained from their historical behavior sequences. However, traditional methods mainly focus on users' interaction records while ignore some important information in sequences, such as time and context, which limits such methods' performance and applications. Besides, it is still an important and challenging task to deal with complex interactions and relationships in sequences and capture users' real-time preferences accurately.

A toy example of the studied scenario is given in Figure 1, where the colored line of dashes indicates the base intensity of event/record A, B and C, and the colored solid line is used to represent the real intensity of the events/records in real time. Meanwhile, due to the occurrence of events/records, a positive or negative effect may be triggered among A, B, and C. For example, A has a negative effect on the occurrence of C, and vice versa, so the occurrence of e_1 or e_3 has a negative effect on the occurrence of A. Besides, event B has a positive effect on C's probability of occurrence, and vice versa. Note that the correlations between events A and B are omitted for simplicity. Specifically, the base intensity (base rate) can be seen as users' long-term static preferences for specific item, and events/records that occurred recently indicate users' short-term dynamic preferences. Furthermore, the prediction or recommendation of events/records depends on both long- and short-term preferences, although they may

D. Wang is with School of Computer Science and Technology, Hangzhou Dianzi University, China. e-mail: (Dongjing.Wang@hdu.edu.cn).

X. Zhang*, corresponding author, is with School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, 310018, China. e-mail: (zhangxin@hdu.edu.cn).

Z. Xiang is with School of Computer & Computing Science, Zhejiang University City College, China. e-mail: (xiangzz@zucc.edu.cn).

D. Yu is with School of Computer Science and Technology, Hangzhou Dianzi University, China. e-mail: (yudj@hdu.edu.cn).

G. Xu is with Advanced Analytics Institute, University of Technology Sydney, Australia. e-mail: (Guandong.xu@uts.edu.au)

S. Deng is with School of Computer Science and Technology, Zhejiang University, China. e-mail: (dengsg@zju.edu.cn)

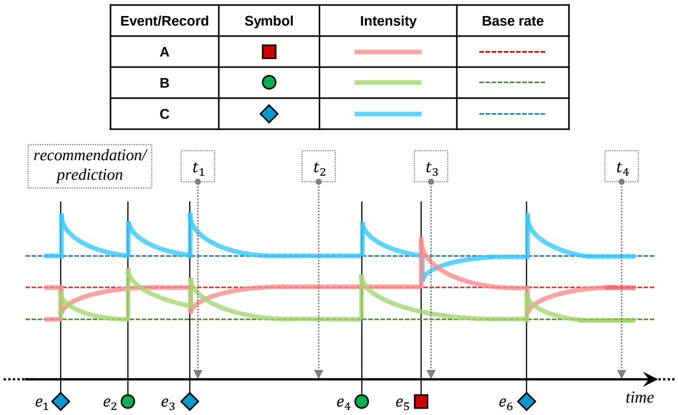


Fig. 1. Illustration of complex interactions and relationships in sequences. Each event/record has a basic probability of occurring (base rate or long-term preferences). At the same time, the occurrence of various events/records may have positive or negative effects (short-term preferences) on the target events/records, causing their actual probability (Intensity) to deviate from its basic probability.

have varying degrees of impact in different situations. For example, the prediction at t_1 and t_3 mainly depends on short-term preference, while the prediction at t_2 or t_4 is mostly up to long-term preference.

In this work, we propose a novel sequential recommendation model named Multivariate Hawkes Process Embedding with attention (MHPE-a), which combines a temporal point process and attention mechanism to predict items that a user would interact with according to her/his historical interaction records. Especially, compared with traditional sequential recommendation methods, temporal point process (TPP) explicitly incorporates important temporal information and model timestamped behaviors in continuous time space. As one variant of TPP, multivariate Hawkes process can learn the low-dimensional feature representations (embeddings) of items and capture the feature interactions as well as latent sequential behavior patterns for better recommendation. Specifically, MHPE-a consists of three main steps. Firstly, MHPE-a models the sequential relationships among items in sequences effectively, and learns the sequential patterns in users' interaction sequences accurately with a multivariate Hawkes process. Secondly, an attention mechanism specifically tailored for this task is used to enhance MHPE-a in extracting users' preferences as well as modeling complex sequences, which can further improve the performance. Thirdly, MHPE-a can perform sequential recommendation to satisfy target users' real-time requirement according to their long-term static preferences as well as short-term dynamic preferences.

Compared with existing approaches, MHPE-a can: 1) effectively model users' behaviors in sequences and precisely capture the complex dynamic relevance among items/records in behavior sequences with multivariate Hawkes process, 2) use attention mechanism to learn and leverage users' long/short-term preferences adaptively to perform accurate sequential recommendation.

The main contributions of this work are summarized as follows:

- We devise a multivariate Hawkes process-based method to model users' behavior sequences and learn the complex sequential relationships among items as well as the corresponding features in a user's behavior sequence.
- We present a sequential recommendation model named MHPE-a that can recommend appropriate items in accordance with the target user's long/short-term preferences to satisfy their real-time needs.
- Comprehensive experiments on two real-world datasets, and the results show that MHPE-a outperforms several state-of-the-art baseline methods.

The rest of this paper is organized as follows. Section II introduces the related work. In Sections III and IV, we describe the problem definition and the proposed model MHPE-a in detail. Then, comprehensive experiments of MHPE-a against state-of-the-art baselines are given in Section V. At last, the conclusion and future works are provided in Section VI.

II. Related Work

In this part, we introduce existing work on sequential recommendation, and related works including temporal point process and attention mechanism which inspire this study.

A. Sequential Recommendation

Users' interactive actions/events recorded in web systems and applications play an important role in understanding their underlying intents and preferences. Many approaches have been proposed to model their interactions in a sequential manner for prediction or recommendation. Typically, the factorized personalized Markov chains [12] model uses the first-order Markov chains and matrix factorization to learn sequential information from users' historical behavior for next-basket recommendation. Wang et al. [13] propose a hierarchical representation model (HRM) to model complicated interactions and perform next basket recommendation. Especially, the adaptability of HRM is enhanced with different aggregation operations, especially nonlinear ones. However, these methods mainly mine the local sequential patterns between adjacent behaviors [14], and they can hardly mine long-term dependence between records in behavior sequence.

Recurrent Neural Networks (RNN) as well as their variants [15] are often used in sequential behavior prediction and recommendation due to their success in sequence modeling [16]. Hidasi et al. [17] apply RNN on session-based recommender systems, and they also propose several variants to classic RNN to improve the performance on specific tasks. However, such approaches generally represent user's historical behaviors with one latent vector or hidden state, while each record may have different importance in prediction or recommendation tasks. Zhu et al. [18] present a modified version of Long Short-Term Memory (LSTM), namely Time-LSTM, to model users' sequential actions. Specifically, Time-LSTM can incorporate time intervals with time gates, and capture both of users' long/short-term preferences, so as to achieve accurate sequential recommendation. Liu et al. [19] propose an approach named Short-Term Attention/Memory Priority Model, which uses different memory to capture users' long/short-term preferences from session context and last-clicks, separately. Ying et al. [20] present a sequential recommendation model, which uses a hierarchical attention network to model users' long/short-term interests. Tang et al. [21] propose a unified and flexible model named Convolutional Sequence Embedding Recommendation (Caser) to learn users' general preferences from their behavior sequences and capture their sequential behavior patterns for top-N sequential recommendation. Kang et al. [22] address the next item recommendation task with a self-attention based sequential model (SASRec), which models users' longer-term semantics as well as their recent actions simultaneously for accurate prediction. Ma et al. [23] combine hierarchical gating network with the Bayesian Personalized Ranking (BPR) to capture users' long/short-term interests for the sequential recommendation. Especially, feature/instance gating modules are adopted to extract latent features and items that are important for prediction/recommendation. However, the proposed approach can capture the dynamic relevance and complex relationships among items in timestamped behavior sequences, and leverage users' long/short-term preferences with attention mechanism adaptively to predict next new behavior/item.

B. Temporal Point Process

A temporal point process (TPP) can model event sequences in continuous time space by learning the time dependency between events. Especially, several variants of TPP have been proposed for specific form of dependency in different tasks. For example, Hawkes processes [24], [25] assumes that past events temporarily increase the probability of future events rather than using a fixed probability, which is known as self-exciting effects. Specifically, the exciting effects are positive, additive over the past events, and exponentially decaying with time [26]. However, in real-world, sequential patterns may violate these assumptions when one event inhibits another instead of exciting it. On the contrary, a self-correcting process [27] assumes that the occurrence of an event reduces the probability of other events by a certain amount. Recurrent Marked TPP [28] combines RNN with marked TPP to predict the probability and time of next event will occur. Mei et al. [26] apply continuous-time LSTM in a multivariate point process and propose Neural Hawkes process to model event sequences in continuous time.

TPP model and its variants span a wide range of applications as event data is prevalent and becoming increasingly available such as online advertisement [29], [30], detection [31], pattern mining [32], information diffusion [33], prediction and recommendation [34]–[37]. For example, Xu et al. [35] develop a framework for modeling the transition events of patient flow via point process, which can be used to predict patients' destination care units and duration days. Dutta et al. [31] propose a fake retweeters detector named HawkesEye, which combines Hawkes process and topic model to fully utilize textual content data and time information for better detection performance. Cai et al. [37] present a long- and short-term Hawkes process (LSHP) model, which combines two Hawkes processes to capture "mutual-influence" of different behaviors as well as "self-influence" of behaviors of the same type. Yang et al. [36] design a novel Recurrent Spatio-Temporal Point Process (RSTPP) to learn the latent dependencies of event times over behavior sequences. Compared with other methods, RSTPP can utilize abundant spatio-temporal information of precedent records for predicting the time of users' next check-in behaviors. Okawa et al. [34] propose a model named Deep Mixture Point Processes (DMPP), which uses deep learning method and point process intensity to capture the effects of unstructured contextual features on target value. Du et al. [38] combine self-exciting point processes with low-rank models to explicitly capture users' patterns from their temporal behaviors and predict relevant items/events or returning-time. Bai et al. [39] present a Demand-aware Hawkes Process (DHP) framework to infer users' requirements from their behavior records. Specifically, users' long/short-term preferences are captured by attention mechanism and convolutional neural network, respectively. Vassøy et al. [40] use a hierarchical RNN to model inter-session relations and capture users' long-term preferences for time and item predictions. Especially, the point process model is adopted to incorporate temporal aspects of user-item interactions for further improving the performance.

C. Attention Mechanism

An attention [41], [42] is intuited from visual attentions of human beings (incline to be attracted by more important parts of a target object). Attention is widely used in many fields, including object detection [43], [44], prediction [45], query suggestion [46], and recommendation [4]. In brief, attention can be used to increase the interpretability and adaptivity of complex models such as neural networks by calculating the weights of different data/information automatically. Recently, many studies attempt to apply an attention mechanism in recommendation. For example, Li et al. [47] combine an encoder with an attention mechanism to capture users' preferences in the current context from their sequential behaviors. Chen et al. [48] combine item-level and component-level attention with Collaborative Filtering to perform sequential recommendation. Xiao et al. [49] propose a model named Attentional Factorization Machines, which uses an attention model and Factorization Machines to model the importance of different features and their interactions. Wang et al. [4] present a content- and context-aware recommendation model called CAME, which use an attention model and Convolutional Neural Network to learn the content features adaptively for music recommendation. Especially, an attention mechanism enables CAME to model different aspects of music and enhanced its ability of capturing music pieces' dynamic features. Huang et al. [8] propose a novel multiattention-based recommendation model, which utilizes multiattention-based deep neural network structures to capture internal social features for accurate group recommendation. Han et al. [50] present a novel recommendation method named Adaptive Deep Latent Factor Model, which can learn users' preferences adaptively from their rated item descriptions, and experimental results show it is effective in recommendation task.

TABLE I
SYMBOLS USED IN THIS WORK

Symbol	Description
U, I	user set and item set
$u \in U$, $i \in I$	a user and an item
S_u	user u's behavior sequence
$H_{u,t} \subseteq S_u$	user u's historical behavior sequence before time t
$\mathbf{U} \in \mathcal{R}^{\lvert U\rvert \times d}$	user embedding matrix
$\mathbf{V} \in \mathcal{R}^{\lvert I\rvert \times d}$	item embedding matrix
$\mathbf{v} \in \mathcal{R}^{d}$	the d-dimensional feature vector representation (embedding) of user or item
$\mu_{u,i}$	the base rate (long-term static preference) of user u for item i
$\alpha_{h,i}$	the degree to which historical item h initially excites the target item i
$\kappa\left(\cdot\right)$	the exponential kernel function that calculate the exponentially decaying of historical influence with time
$\delta_u \geq 0$	the decay rate of historical influence
$\mathbf{W}$, $\mathbf{b}$, $\mathbf{W}_l$, b_l, $\mathbf{W}_s$, b_s	model parameters
β_u^l, β_u^s	adaptive weights of user u's long/short-term preferences
$\tilde{\lambda}_{i\mid u}\left(t\right)$	the predicted preference of u for item i at time t
$>_{u,t}$	the ranking of candidate item for user u at time t

III. Definition

As shown in Table I, we give the key notions and symbols of sequence data used in this paper.

Definition 1. **Record**. Let $U = \left\{u_1,\ u_2,\ ...,\ u_{\lvert U\rvert}\right\}$ denote the user set, $I = \left\{i_1,\ i_2,\ ...,\ i_{\lvert I\rvert}\right\}$ represent the item set, and T be a time domain. A record r is a triplet $(u, i, t) \in U \times I \times T$, which represents the interaction record between user u and item i at time t .

Definition 2. **Sequence**. Let S be the collection of all users' behavior sequences, and as for a user $u \in U$, her/his behavior sequence $S_u \in S$ is formally defined as $S_u := \left[\left(u, i_1, t_1\right), \left(u, i_2, t_2\right), \cdots, \left(u, i_{\lvert S_u\rvert-1}, t_{\lvert S_u\rvert-1}\right)\right]$, where $(u, i, t) \in S$ and $S_u \subseteq S$. Here, an item can refer to a piece of music, a point of interests (POI), or an action on websites/applications.

Definition 3. **Sequential Recommendation**. Given a user $u \in U$ and her/his historical behavior sequence $H_{u,t} \subseteq S_u$ before time t, predict u's preference for item i at t and make a recommendation.

IV. Methodology

A. Temporal Point Process

Temporal Point Process (TPP) can model event sequences in continuous time space by learning the time dependency between events. In this paper, an event (record) (u, i, t) is that a user u interacted with an item i at time $t \in \mathcal{R}^+$ (a set of non-negative real numbers).

A typical TPP models the probability of an event occurs at time t (more precisely, in the infinitesimally wide interval $[t, t + \Delta t))$ as $\lambda\left(t\right)\Delta t$. Specifically, $\lambda\left(t\right) \geq 0$ is known as the intensity function, which represents the arrival rate of sequential events. As a well-known generalization of TPP, Hawkes process models the events sequences as well as the interactions between events in a sequence. Specifically, the conditional intensity function in the Hawkes process characterizes the arrival rate of a current event given past events and models the effects between historical events and the current one, which is formally defined as

$$\lambda\left(t\right) = \mu + \int_0^t \alpha\kappa\left(t - s\right) dN\left(s\right), \qquad (1)$$

where $\mu \geq 0$ is the base rate(base intensity) of the current event, depicting the generating rate of events, α is excitation rate of past events on the current event, and $\kappa\left(t - s\right) dN\left(s\right)$ is a kernel function that describes excitation of historical events $N\left(s\right)$ on the current one at time t. Specifically, the excitation in a typical TPP is positive, additive over the historical events, and exponentially decaying with time [26]. Furthermore, the conditional intensity function is extended into a multi-dimensional one where each dimension represents one type of event. Therefore, a Hawkes process can deal with different types of events. The excitation of event type k' on event type k in a multivariate Hawkes process is captured with excitation rate parameter $\alpha_{k',k}$.

Formally, a conditional intensity function can be used to represent the arrival rate of events, and it can be formally defined as the frequency of events in a small time window $[t, t + \Delta t)$ given all past events $H\left(t\right)$ in historical sequence.

Our goal is to predict users' future actions from their historical behavior sequences. Especially, we propose a novel sequential recommendation approach named Multivariate Hawkes Process Embedding with attention (MHPE-a). It well combines a TPP with embedding and attention mechanism to 1) effectively model users' behavior sequences, 2) precisely capture dynamic relevance and complex relationships among items in sequences, 3) incorporating and leveraging users' long-term static preferences and short-term dynamic preferences adaptively to achieve accurate sequential recommendation.

B. Multivariate Hawkes Process Embedding with Attention

The proposed approach named Multivariate Hawkes Process Embedding with attention (MHPE-a) model is based on the following characteristics of sequential patterns: 1) there exist complex correlations among items in users' behavior sequences, which may have different impacts on the prediction of next behavior or item, and 2) users have both static long-term preferences and dynamic short-term preferences, which may have different influences on sequential recommendation for each user.

The framework of the proposed approach MHPE-a is given in Figure 2. Specifically, MHPE-a consists of three components: 1) an embedding layer for learning representation of users and items, 2) a multivariate Hawkes process for modeling complex behavior sequences, and 3) an attention mechanism for capturing and leveraging users' long/short-term preferences and performing recommendation.

The basic idea of MHPE-a is to learn the correlations among items in behavior sequences of different users for accurate prediction and recommendation. Specifically, we first learn the

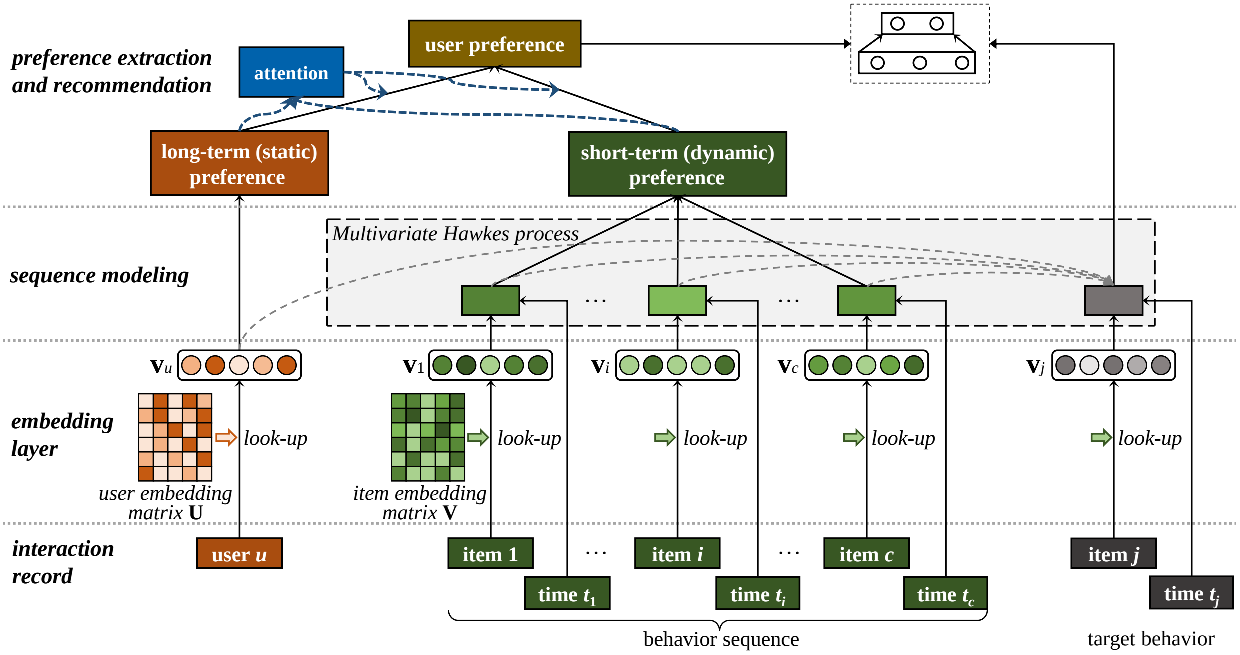


Fig. 2. The framework of MHPE-a consists of three main components: 1) an embedding layer for representation learning, 2) a multivariate Hawkes process for sequence modeling, and 3) an attention mechanism for users' preferences capturing and recommendation.

low-dimensional denser feature representations (embeddings) of users and items, which is more informative and effective than the users/items' id or one-hot representations. Then, a multivariate Hawkes process model is devised to model behavior sequences in continuous time space by learning the time dependency and correlations between events. Especially, an attention mechanism is adopted to enhance MHPE-a's ability of modeling complex sequences as well as leveraging users' long/short-term preferences. At last, we can perform next item recommendation based on users' historical behavior sequences.

Compared with existing methods, MHPE-a can incorporate both long/short-term user preferences in an adaptive way. Besides, the dynamic and complex impact of historical items in behavior sequences on the next item can be effectively modeled by its Hawkes process with attention mechanism. Next, we introduce each of its component in detail.

*1) **Representation Learning with Embedding Layer:*** In traditional recommendation models, user and item are generally represented with one-hot vectors, whose dimension is the same as the size of item set. However, one-hot representation suffers from serious dimensional disaster and data sparsity problems [51], especially when the size of item set reach millions or even larger.

In this work, the proposed model first learns the informative low-dimensional denser feature representations (embeddings) of users and items, which capture both items' features and relationships. Formally, each item $i \in I$ in the behavior sequences is transformed into corresponding embedding $\mathbf{v}_i \in \mathcal{R}^d$ with an item embedding matrix $\mathbf{V} \in \mathcal{R}^{|I| \times d}$, where d represents the dimension of items' embeddings and I is the item set. Specifically, $\mathbf{v}_i = \mathbf{v}_i' \mathbf{V}$, where $\mathbf{v}_i' \in \mathcal{R}^{1 \times |I|}$ is i's one-hot representation that consists of "0" in all dimensions with the exception of a single "1" in one dimension used uniquely to identify the item. Similarly, user u's embedding $\mathbf{v}_u \in \mathcal{R}^d$ can be obtained by looking up the user embedding matrix $\mathbf{U} \in \mathcal{R}^{|U| \times d}$ in the embedding layer.

Although recommendation methods like matrix factorization or latent factor models [52] can also learn the feature vectors of users and items, the proposed model can capture more high-level dynamic key features by adopting a Hawkes process and attention mechanism.

*2) **Sequence Modeling based on Multivariate Hawkes Process:*** A multivariate Hawkes process is adopted to model each user's behavior sequence, and then predict/recommend her/his next behavior (target item) according to historical records. Especially, compared with traditional methods, the multivariate Hawkes process used in MHPE-a can capture important temporal information and the complex feature interactions between different records/items in users' timestamped behavior sequences for better recommendation. Specifically, the prediction of a target item can be done according to her/his historical behaviors. Formally, given user $u \in U$ as well as u's historical behavior sequence, the conditional intensity function for the arrival of target item $i \in I$ at time t is formally defined as follows:

$$\tilde{\lambda}_{i|u}(t) = \mu_{u,i} + \sum_{h \in H_{u,t}} \alpha_{h,i} \kappa_u (t - t_h), \qquad (2)$$

where $\mu_{u,i}$ is the base rate (u's long-term static preferences for item i, and $H_{u,t} \subseteq S_u$ denotes u's most recent historical behavior sequence before time t. Especially, $H_{u,t}$ is truncated from S_u as a subsequence with fixed length to reduce computation cost. $\alpha_{h,i}$ represents the degree to which h initially excites i. $\kappa_u(\cdot)$ is an exponential kernel function for u, which calculates the exponentially decaying of historical influence with time, i.e.,

$$\kappa_u\left(t-t_h\right)=\exp\left(-\delta_u\left(t-t_h\right)\right), \tag{3}$$

where $\delta_u \geq 0$ denotes the decay rate of historical influence, and it is a user-dependent parameter since each user's preference may decay in different rates. Especially, we relax the positivity constraint on α and μ and allow them to range over $\mathcal{R}$ so we can model effects of inhibition ($\alpha < 0$) and inertia ($\mu < 0$) [26]. In other words, two terms in Equation (2) model user u's long-term static preference and short-term dynamic interest, respectively. Specifically, when a user interacts with an item, the intensities of all items are elevated or inhibited first, and then approach their base rates μ as the influence of a previous event decays towards 0.

However, since the positivity constraint on α and μ is relaxed to $\mathcal{R}$, the result of intensity function $\tilde{\lambda}_{i|u}(t)$ could now be negative. Therefore, we define the probability that user u is interested in target item i at time t as

$$p_{i|u}(t)=\frac{\exp\left(\tilde{\lambda}_{i|u}(t)\right)}{\sum\limits_{k\in I}\exp\left(\tilde{\lambda}_{k|u}(t)\right)}. \tag{4}$$

For each target node $i \in I$, Equation (4) defines a conditional distribution $p_{\cdot|u}(t)$ over the entire item set I.

Generally, the inputs of a traditional multivariate Hawkes process are sequences of the original user or item IDs, which have very limited representation capacity. Instead, we feed the d-dimensional embeddings of users and items into the intensity function. Specifically, $\mu_{u,i}$, the base rate of user u interacting with item i, depends on the matching degree between u's preferences and i's features, which is a mapping function$f'(\cdot):\mathcal{R}^d\times\mathcal{R}^d\rightarrow\mathcal{R}$. It can be formally defined as a vector dot product as follows:

$$\mu_{u,i}=f'(\cdot)={\mathbf{v}_u}^{\mathrm{T}}\mathbf{v}_i, \tag{5}$$

where $\mathbf{v}_u$ is user u's embedding and $\mathbf{v}_i$ is the embedding of item i. To note that other metrics between vectors, such as cosine similarity and Euclidean distance, can also be adopted here.

Besides, $\alpha_{h,i}$, the degree to which historical item h in a behavior sequence initially excites current item i, depends on features of h and i, i.e.,

$$\alpha_{h,i}=f'(\cdot)={\mathbf{v}_h}^{\mathrm{T}}\mathbf{v}_i, \tag{6}$$

where $\mathbf{v}_h$ and $\mathbf{v}_i$ are the embeddings of historical item h and item i. Then, the multivariate Hawkes process defined in Equation (2) can summarize the embeddings of all historical items and calculate their influence on the prediction/recommendation of target item.

*3) **Preference Extraction and Recommendation via Attention:*** In sequential recommender systems, users have long-term general static preferences as well as short-term dynamic interest. Both of them can help accurate recommendation. In [53], long/short-term preferences are combined in a concise but static way, e.g., $p_u = p_u^{long} + p_u^{short}$, although they may have different roles and influencing mechanisms for each user.

Different from them, we present an adaptive method based on attention for leveraging and fusing users' preferences. Specifically, we use an attention model to determine the dynamic weights of users' long/short-term preferences. They are formally defined as:

$$\begin{aligned}\beta_u^l&=\operatorname{relu}\left(\mathbf{W}_l\mathbf{v}_u+b_l\right),\\ \beta_u^s&=\operatorname{relu}\left(\mathbf{W}_s\bar{\mathbf{v}}_H+b_s\right),\end{aligned} \tag{7}$$

where $\mathbf{W}_l\in\mathcal{R}^d$, $\mathbf{W}_s\in\mathcal{R}^d$, $b_l\in\mathcal{R}$, $b_s\in\mathcal{R}$ are model parameters, relu is the rectified linear unit, and $\bar{\mathbf{v}}_H$ is the aggregation of historical items' embeddings, which is obtained via the average pooling strategy as follows:

$$\bar{\mathbf{v}}_H=\sum_{h\in H_{u,t}}\mathbf{v}_h\Big/|H_{u,t}|. \tag{8}$$

Then, the preference of $u\in U$ for target item $i\in I$ at time t given S_u is defined as

$$\tilde{\lambda}_{i|u}(t)=\beta_u^l\mu_{u,i}+\beta_u^s\sum_{h\in H_{u,t}}\alpha_{h,i}\kappa\left(t-t_h\right). \tag{9}$$

At last, we can perform recommendation according to the ranking scores of two item i and i' in MHPE-a, which is calculated as

$$i>_{u,t}i' :\Leftrightarrow p_{i|u}(t)>p_{i'|u}(t). \tag{10}$$

C. Learning

In a learning process, Equation (4) is maximized over all users' behavior sequences in the training dataset. However, the soft-max function in Equation (4) has high complexity, which is proportional to the item set size $|I|$. Especially, item set size may reach millions in real-world applications.

Therefore, negative sampling [54] is used to calculate the original soft-max function in Equation (4) approximately, which is computationally efficient. Then, the log probability can be defined as:

$$\log p_{i|u}(t)\propto\log\sigma\left(\tilde{\lambda}_{i|u}(t)\right)+n\cdot E_{i'\sim P_I}\left[\log\sigma\left(-\tilde{\lambda}_{i'|u}(t)\right)\right], \tag{11}$$

where $\sigma(x)$ is a *sigmoid* function, n is "negative" sample count, and i' is the item sampled from item set based on P_I, which is a noise distribution defined with empirical unigram distribution over items.

Here, the count of negative samples n is much smaller than item set size $|I|$, and the training time is independent of the item set size $|I|$. Traditional optimization methods, such as stochastic gradient descent algorithms, can be adopted to optimize the objective function defined in Equation (11).

V. Experiments

In this section, comprehensive experiments of the proposed approach MHPE-a and baselines are conducted on real-world datasets. In detail, we first describe the experimental designs, including the datasets, baseline models, and the evaluation metrics. Next, we show the learned embeddings in quantitative way, and evaluate the impact of the embeddings' dimension on recommendation accuracy. Then, MHPE-a is evaluated against

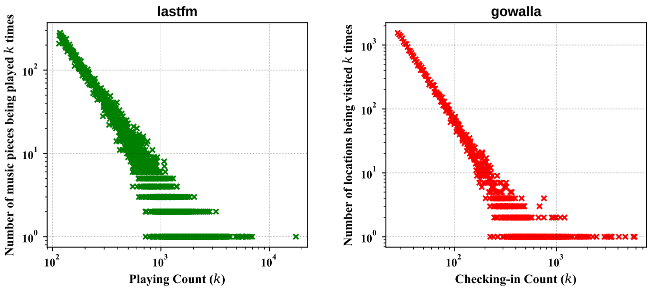


Fig. 3. Popularity analysis of datasets

state-of-the-art methods on two real-world datasets. At last, we evaluate how the attention component and the sparsity of datasets influence recommendation performance.

A. *Experimental Designs*

The detailed experimental designs are firstly introduced, including dataset and task, baselines, and evaluation metrics.

*1) **Dataset and Task:*** Two real-world datasets and the corresponding statistics information are listed in Table II.

Lastfm[1] dataset is extracted from Last.fm[2], which is an international online music service website. If the action that user u listens to music i at time t, there is a tuple record $(u, i, t) \in S_u$, where S_u is u's behavior sequence. Especially, we have filtered users or music pieces with few interactions. As shown in Table II, the final lastfm dataset contains 857,242 listening records by 938 users to 30,000 music pieces.

Gowalla[3] dataset is extracted from a location-based social network where users can check in at specific location to share it. Specifically, users with few checking-in records or unpopular locations are filtered, and the final dataset contains 473,628 annotating records by 1,594 users to 30,000 locations.

Moreover, Figure 3 gives popularity information (logarithm) of all datasets, which shows that most items are not very popular, and only a small number of items are interacted with frequently, which are consistent with the power law distribution [55].

Each dataset is divided into a training set and a test set, which are non-overlapping. Specifically, the training set consist of the 80% users' behavior sequences of (random selected) and first half of the remaining 20% users' historical behavior sequences, while the second half of the remaining 20% users' sequences are used as the test set. Specifically, each user u's behavior sequence $S_u := \left[(u, i_1, t_1), (u, i_2, t_2), \cdots, \left(u, i_{|S_u|-1}, t_{|S_u|-1}\right)\right]$ in the test set generates $|S_u| - 1$ test cases, where the k-th test case is to perform recommendation at time t_{k+1} given u's historical sequences $S_u := [(u, i_1, t_1), (u, i_2, t_2), \cdots, (u, i_k, t_k)]$ with the ground truth i_{k+1}. Note that the task studied in this work is sequential new recommendation, which predicts the target user's next new behavior/item that has not appeared in her/his historical sequences. Especially, the sequential new recommendation task is an important but more challenging task compared with traditional sequential recommendation.

TABLE II
COMPLETE STATISTICS OF TWO DATASETS

Dataset	#(Users)	#(Items)	#(Records)
Lastfm	938	30,000	857,242
Gowalla	1,594	30,000	473,628

*2) **Baselines:*** The proposed approach MHPE-a is evaluated against the following baselines, including basic and state-of-the-art models:

- Popularity-based Method (PM) performs recommendation based on items' popularity in training data.
- Factorizing Personalized Markov Chains (FPMC) [12] combines matrix factorization and first-order Markov chain to learn sequential information simultaneously from users' historical behavior sequences for sequential recommendation.
- Hierarchical Representation Model (HRM) [13] encodes sequential information and users' general taste as one vector with hierarchical representation learning model for next basket recommendation. Especially, two kinds of aggregation operations, i.e., max pooling and average pooling, are adopted by HRM to learn the representations of users' preferences, which correspond to HRM-max and HRM-avg, respectively.
- Recommendation based on Distributed Representation (RDR) [56] can learn the feature vectors of items from behavior sequences with a skip-gram model [54], and acquire users' preferences from their historical behaviors for personalized recommendation.
- Sequential Hierarchical Attention Network-based method (SHAN) [20] uses a hierarchical attention mechanism to mine long/short-term preferences for sequential recommendation.
- Convolutional Sequence Embedding Recommendation model (Caser) [21] embeds items in users' behavior sequences into an "image" in the time and latent spaces and learns both general preferences and sequential patterns with convolutional filters for recommendation.
- Self-Attention based Sequential Recommendation method (SASRec) [22] models users' longer-term semantics as well as their recent actions simultaneously for accurate next item recommendation
- Hierarchical Gating Network (HGN) [23] adopts a feature gating module and an instance gating module to select informative latent features and items, and captures both the long- and short-term user interests for sequential recommendation.

*3) **Evaluation Metrics:*** In the evaluation step, every method generates a recommendation list of k items (top-k recommendation), which is evaluated by two quality metrics, i.e., recall and Mean Reciprocal Rank (MRR).

Recall is the fraction of the total amount of hits in all testcases. Specifically, a hit means the target item (ground truth) appears in the recommendation list. For instance, if there

[1] http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html

[2] http://www.last.fm

[3] http://snap.stanford.edu/data/loc-gowalla.html

exists a record (u, i, t) in the test set and the recommended list of u contains i, then it is called a hit. Recall is formally defined as:

$$Recall@k = \#(hit)/\#(testcase)$$

where k is the length of a recommendation list, $\#(hit)$ is the amount of hits, and $\#(testcase)$ is the amount of all testcases.

MRR is a ranking evaluation metrics, which indicates the average of the reciprocal ranks of target items in a recommendation list, i.e.,

$$MRR@k = \left(\sum 1/rank_n\right) \Big/ \#(testcase)$$

where $rank_n$ is the ranking of the n_{th} test case's target item in the generated recommendation list.

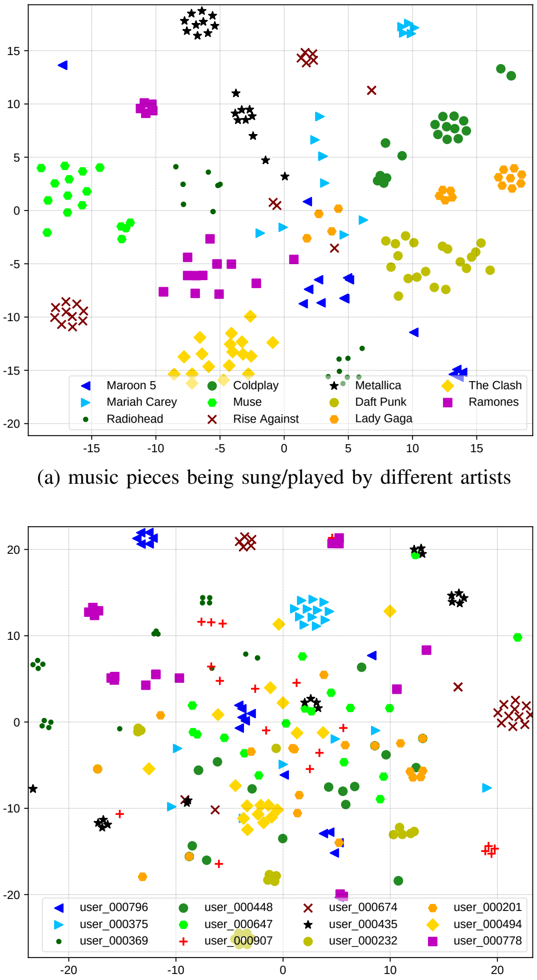
*4) **Implementation Details:*** In the training phase, we set the batch size to 512, negative sample count to 5, dimension of embedding to 256, number of epochs to 200. Besides, the parameters in MHPE-a are updated via Adam optimizer [57] with the learning rate $1e-3$. Especially, during the training process, the weight decay in Adam optimizer is set as $3e-3$ to prevent over-fitting. The proposed approach MHPE-a was implemented using the PyTorch 1.5.0 framework with Python 3.6, and all experiments were conducted on a server with Intel(R) Xeon(R) Silver 4108 CPU, GeForce RTX 2080Ti GPU, 128GB memory, and running Ubuntu 18.04.

B. Visual Illustration of Embedding

In this section, we visually illustrate the learned embeddings with t-SNE [58], which show high-dimensional vectors in 2-dimensional space via dimensionality reduction. Note that we only give results on lastfm dataset for brevity.

The illustration of the embeddings of music pieces by eleven famous artists is given in Figure 4a, and Table III lists these artists and their tags (genre information) in Last.FM. we can see that music pieces played/sung by same or similar artists are relatively close in the 2-dimensional space, and it is because each singer/musician has her/his own genre, which is also reflected in their music pieces. Especially, MHPE-a can effectively learn music pieces' intrinsic features from music listening sequences.

In addition, Figure 4b illustrates the embeddings of several users' listening records (music pieces), and the results show that users have different behavior patterns and preferences. For example, user_000375 and user_000674 have relatively focused interest (long-term static preferences) and their records cluster tightly in 2-dimension space. In contrast, user_000647 has a broader range of interests (short-term dynamic preferences). Especially, MHPE-a can effectively leveraging users' long/short-term preferences adaptively with attention mechanism to achieve better recommendation.



(a) music pieces being sung/played by different artists

(b) music pieces from different users' listening records

Fig. 4. Visualization of music pieces' embeddings in two-dimensional space

TABLE III
GENRE OF ARTISTS

No.	Artist	Tags
1	Maroon 5	pop, rock, pop rock, alternative
2	Mariah Carey	pop, rnb, female vocalists, soul
3	Radiohead	alternative, rock, alternative rock, indie, electronic
4	Coldplay	alternative, rock, alternative rock, britpop
5	Muse	alternative rock, rock, alternative, progressive rock
6	Rise Against	punk rock, melodic hardcore, punk, hardcore, rock
7	Metallica	thrash metal, heavy metal, metal, hard rock
8	Daft Punk	electronic, house, dance, techno, electronica
9	Lady Gaga	pop, dance, electronic, female vocalists
10	The Clash	punk,punk rock, rock, british, classic rock
11	Ramones	punk rock, punk, 70s, classic rock, rock

C. Impacts of Parameter Settings

*1) **Dimension:*** The dimension of embeddings is quite important in both sequence modeling and recommendation. Specifically, the embeddings with high dimension can capture more useful information at the cost of more computation resources and time. Therefore, we firstly evaluate the proposed model MHPE-a with different dimensions (16, 32, 64, 128 and 256) to investigate the impact of embedding's dimension on recommendation performance and then determine the proper dimension to achieve performance balance between accuracy and efficiency. As shown in Figure 5, MHPE-a with larger embedding dimension achieve better performance in metrics of

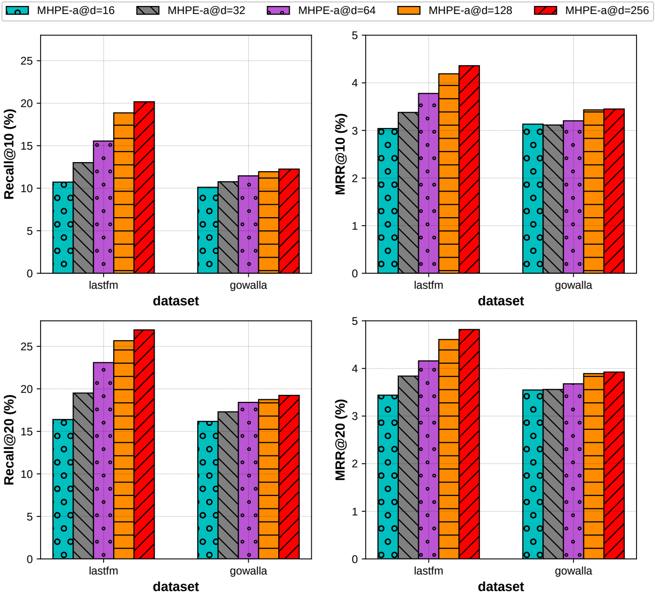


Fig. 5. Experimental results of the dimension's impact

recall and MRR, which shows that high-dimensional embeddings can indeed learn useful features and represent users and items accurately. Besides, the accuracy tends to get relatively stable when the dimension reaches 256. Therefore, we set the embeddings' dimension as 256 for subsequent experiments.

*2) **Sequence Length**:* The sequence length parameter is designed to truncate user u's whole historical behavior sequence S_u into a fixed length sequence $H_{u,t}$ before time t. As shown in Figure 6, the proposed approach MHPE-a is evaluated with historical sequence length c varying from 1 to 5. We can see that the performance of MHPE-a in terms of recall and MRR firstly increases and then drops when c gets larger. Especially, the optimal performance of most cases is achieved when c is 3. Besides, the performance is relative stable when $c \geq 3$, because the attention mechanism in MHPE-a helps to capture key items in sequences that are important for tasks of prediction and recommendation. Therefore, we set the sequence length c as 3 for subsequent experiments.

Furthermore, we have presented how the average weights of users' long/short-term preferences change with different sequence length c during recommendation. Specifically, as shown in Figure 7, the proposed model MHPE-a focuses more on users' long-term preferences when c is small. The reason is that it is challenging to infer the user' preference accurately only from her/his most recent record, and the prediction/recommendation relies more on the long-term preference. Besides, the weight of short-term preferences increases when c gets larger. In addition, the results show the interpretability of MHPE-a during the process of recommendation.

D. Comparison with Baselines

In this section, the proposed method MHPE-a is compared with basic and state-of-the-art baselines, including Popularity-based Method (PM), Factorizing Personalized Markov Chains (FPMC) [12], Hierarchical Representation Model (HRM-max and HRM-avg) [13], Recommendation based on Distributed Representation (RDR) [56], Sequential Hierarchical Attention Network-based method (SHAN) [20], Convolutional Sequence Embedding Recommendation model (Caser) [21], Self-Attention based Sequential Recommendation method (SASRec) [22], and Hierarchical Gating Network (HGN) [23]. The results on all datasets are given in Figure 8 and 9, respectively.

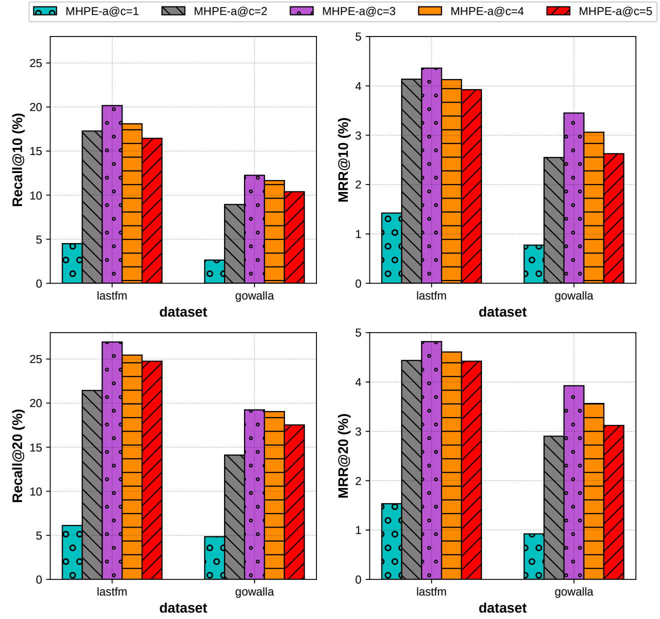


Fig. 6. Experimental results of the historical sequence length's impact

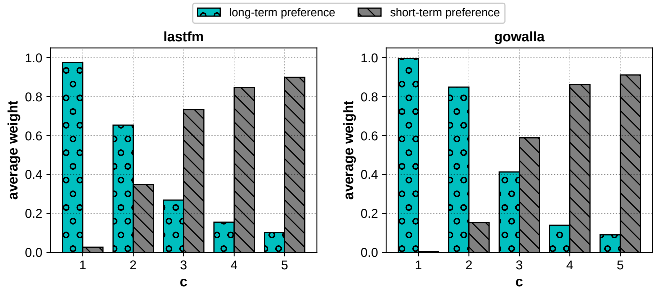


Fig. 7. Experimental results of the historical sequence length's impact on the weights of long- and short-term preferences

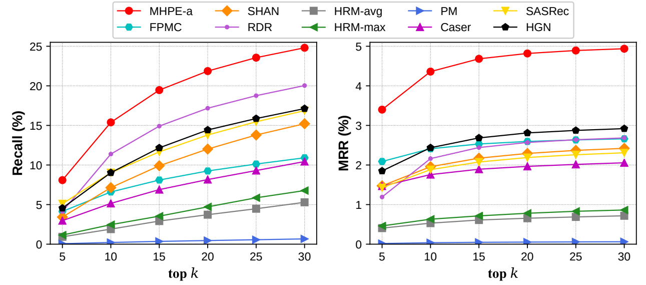


Fig. 8. Comparison with baselines on lastfm dataset

We can observe that MHPE-a achieves higher accuracy than baselines in both recall and MRR. The improvements indicate that the user's short-term dynamic preferences are

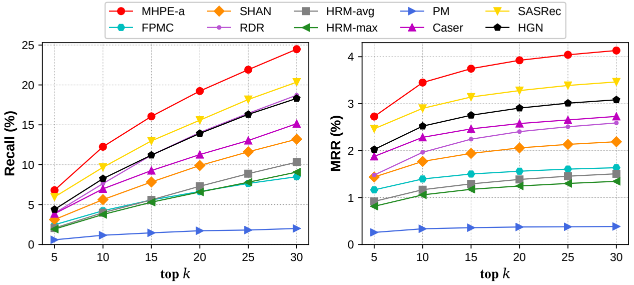


Fig. 9. Comparison with baselines on gowalla dataset

TABLE IV
STATISTICS OF DATASETS WITH DIFFERENT SPARSITY

Lastfm	**#(Users)**	938	934	926	919
	#(Items)	30,000	25,000	20,000	15,000
	#(Records)	857,242	849,789	839,019	822,436
	Sparsity	96.95%	96.36%	95.47%	94.03%
Gowalla	**#(Users)**	1,594	1,538	1,462	1,359
	#(Items)	30,000	25,000	20,000	15,000
	#(Records)	473,628	444,826	409,964	366,414
	Sparsity	99.01%	98.84%	98.60%	98.20%

indeed important in improving sequential recommendation. Specifically, it outperforms other embedding-based methods (RDR, HRM-avg, and HRM-max) because it can fully exploit complex sequential information with a multivariate Hawkes process and learn the embeddings effectively from users' behavior sequences. Especially, the attention mechanism enables MHPE-a to model users' long-term static preferences and short-term dynamic preferences adaptively, and enhances its ability to capture the features which are important in sequential recommendation. Besides, MHPE-a outperforms sequential recommendation methods (FPMC, SHAN, Caser, SASRec, and HGN), and the reason is two-fold. Firstly, MHPE-a can capture more key features other than the correlations between adjacent items with multivariate Hawkes process, and fully exploit sequential information by adopting embedding layer and attention mechanism. Secondly, the task of sequential new recommendation, especially on sparse datasets, is a more challenging task compared with traditional sequential recommendation, which may influence the performance of some baselines. Furthermore, MHPE-a achieves better performance than PM, because PM only uses popularity information, and ignores users' preferences and sequential patterns.

In conclusion, the comparison with baselines shows that MHPE-a is effective in inferring both the users' long/short-term preferences from behavior sequences as well as in incorporating them into sequential recommendation.

E. Impacts of Data Sparsity

In this section, the proposed approach MHPE-a and baselines are evaluated on datasets with different sparsity, which are generated via filtering items with low interacting frequency. Specifically, the statistics information of all datasets is given in Table IV.

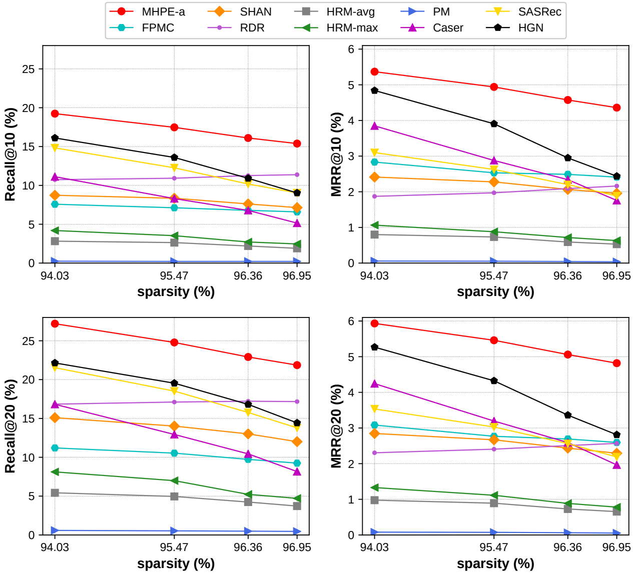


Fig. 10. Performance on lastfm datasets with different sparsity

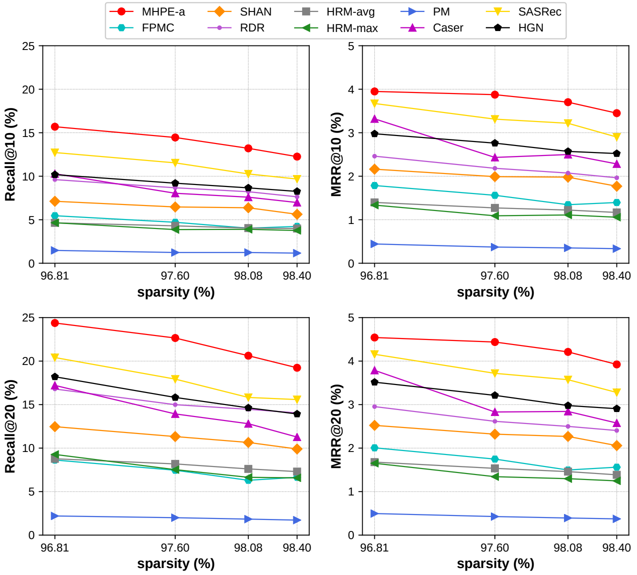


Fig. 11. Performance on gowalla datasets with different sparsity

The results on all datasets are given in Figure 10 and 11, respectively. We can observe that the proposed approach MHPE-a still achieves better performance than baselines in terms of recall and MRR over all datasets with different sparsity. The reason is that MHPE-a can fully exploit complex sequential information to alleviate the influence of sparse interactions in datasets. Besides, the performance of some methods on gowalla datasets is not as good as that on lastfm datasets. The reason is that the gowalla datasets are sparser than lastfm datasets, and the correlations among sequential records in gowalla datasets are not as strong as lastfm datasets. In conclusion, the results show MHPE-a can effectively handle datasets with different sparsity.

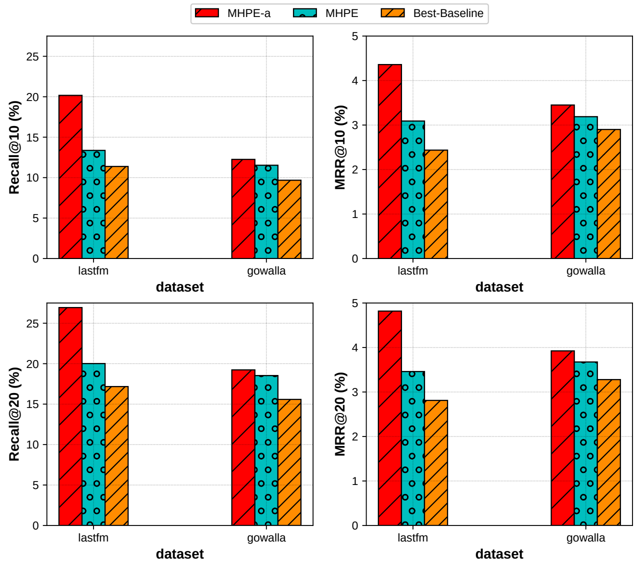


Fig. 12. Experimental results of attention component

F. Effects of Attention

In order to investigate the effect of attention mechanism used in MHPE-a, we also conduct ablation experiments to compare MHPE-a and its variant MHPE that has no attention component, with the baseline that has best performance (Best-Baseline).

The results are given in Figure 12, and we can observe that MHPE-a outperform MHPE on two datasets. Taking recall@10 as an example, the relative performance improvements by MHPE-a over MHPE on lastfm dataset is 50.91%. Besides, the performance gap between MHPE-a and MHPE on gowalla dataset is not as large as it on lastfm dataset, and the reason is that gowalla dataset is much sparser than lastfm dataset, which influence the performance improvement achieved by attention component in MHPE-a. Moreover, both MHPE-a and MHPE outperform Best-Baseline, which further shows the effectiveness of multivariate Hawkes process.

In conclusion, the key components in MHPE-a, including attention mechanism and multivariate Hawkes process, can indeed help to achieve accurate recommendation.

VI. Conclusion and Future Works

In this work, we present a novel sequential recommendation method named Multivariate Hawkes Process Embedding with attention (MHPE-a). It relies on a temporal point process with embedding and an attention mechanism to recommend items that users may interact with based on their historical data. Specifically, it consists of three main components: 1) an embedding layer for learning representation of users and items, 2) a multivariate Hawkes process for modeling complex behavior sequences, and 3) an attention mechanism for capturing and leveraging users' long/short-term preferences and performing recommendation.

Compared with existing approaches, MHPE-a can: 1) effectively model complex behavior sequences with temporal point process, 2) precisely capture dynamic relevance and complex relationships among items in sequences, 3) incorporate and leverage users' long/short-term preferences adaptively for accurate sequential recommendation. Comprehensive experiments are conducted on two real-world datasets (lastfm and gowalla), and the results show that it outperforms state-of-the-art methods.

In future work, we plan to incorporate more auxiliary information, such as users' social data [59], to alleviate the problem of data sparsity and cold start. Besides, we will try combining advanced deep learning techniques [60] to enhance ability of sequence modeling for better sequential recommendation. Moreover, we will also extend the proposed approach for other application scenarios, such as heterogeneous behavior modeling on e-commercial datasets.

Acknowledgements

This research was supported by Natural Science Foundation of Zhejiang Province under No. LQ20F020015, LQ21F020015, and LQ21F020007, and the Fundamental Research Funds for the Provincial University of Zhejiang by Hangzhou Dianzi University under No. GK199900299012-017.

References

[1] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE transactions on cybernetics*, vol. 49, no. 3, pp. 1084–1096, 2019.

[2] X. Luo, M. Zhou, S. Li, Y. Xia, Z. You, Q. Zhu, and H. Leung, "An efficient second-order approach to factorize sparse matrices in recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 946–956, 2015.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] D. Wang, X. Zhang, D. Yu, G. Xu, and S. Deng, "Came: Content-and context-aware music embedding for recommendation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[5] W. Luan, G. Liu, C. Jiang, and L. Qi, "Partition-based collaborative tensor factorization for poi recommendation," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 437–446, 2017.

[6] D. Li, Z. Gong, and D. Zhang, "A common topic transfer learning model for crossing city poi recommendations," *IEEE transactions on cybernetics*, vol. 49, no. 12, pp. 4282–4295, 2019.

[7] W. Wang, G. Zhang, and J. Lu, "Hierarchy visualization for group recommender systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1152–1163, 2019.

[8] Z. Huang, X. Xu, H. Zhu, and M. Zhou, "An efficient group recommendation model with multiattention-based neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[9] S. Deng, D. Wang, Y. Li, B. Cao, J. Yin, Z. Wu, and M. Zhou, "A recommendation system to facilitate business process modeling," *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1380–1394, 2017.

[10] C.-D. Wang, Z.-H. Deng, J.-H. Lai, and S. Y. Philip, "Serendipitous recommendation in e-commerce using innovator-based collaborative filtering," *IEEE transactions on cybernetics*, vol. 49, no. 7, pp. 2678–2692, 2019.

[11] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–36, 2018.

[12] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 811–820.

[13] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 403–412.

[14] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 729–732.

[15] G. Bao, Y. Zhang, and Z. Zeng, "Memory analysis for memristors and memristive recurrent neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 96–105, 2020.

[16] Y. Zhang, Y. Wang, and G. Luo, "A new optimization algorithm for non-stationary time series prediction based on recurrent neural networks," *Future Generation Computer Systems*, vol. 102, pp. 738–745, 2020.

[17] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[18] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to do next: Modeling user behaviors by time-lstm." in *IJCAI*, vol. 17, 2017, pp. 3602–3608.

[19] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: Short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2018, p. 1831–1839.

[20] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention network," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, ser. IJCAI'18. AAAI Press, 2018, p. 3926–3932.

[21] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 565–573.

[22] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 197–206.

[23] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 825–833.

[24] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[25] P. Embrechts, T. Liniger, and L. Lin, "Multivariate hawkes processes: an application to financial data," *Journal of Applied Probability*, vol. 48, no. A, pp. 367–378, 2011.

[26] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems*, 2017, pp. 6754–6764.

[27] R. Rotondi and E. Varini, "Failure models driven by a self-correcting point process in earthquake occurrence modeling," *Stochastic environmental research and risk assessment*, vol. 33, no. 3, pp. 709–724, 2019.

[28] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1555–1564.

[29] K. Parmar, S. Bushi, S. Bhattacharya, and S. Kumar, "Forecasting ad-impressions on online retail websites using non-homogeneous hawkes processes," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1089–1098.

[30] J. Zhang, Z. Wei, Z. Yan, M. Zhou, and A. Pani, "Online change-point detection in sparse time series with application to online advertising," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1141–1151, 2019.

[31] H. S. Dutta, V. R. Dutta, A. Adhikary, and T. Chakraborty, "Hawkeseye: Detecting fake retweeters using hawkes process and topic modeling," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2667–2678, 2020.

[32] J. Pang, J. Huang, X. Yang, Z. Wang, H. Yu, Q. Huang, and B. Yin, "Discovering fine-grained spatial pattern from taxi trips: Where point process meets matrix decomposition and factorization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3208–3219, 2017.

[33] M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network evolution," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 1305–1353, 2017.

[34] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda, "Deep mixture point processes: Spatio-temporal event prediction with rich contextual information," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 373–383.

[35] H. Xu, W. Wu, S. Nemati, and H. Zha, "Patient flow prediction via discriminative learning of mutually-correcting processes," *IEEE transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 157–171, 2017.

[36] G. Yang, Y. Cai, and C. K. Reddy, "Recurrent spatio-temporal point process for check-in time prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 2203–2211.

[37] R. Cai, X. Bai, Z. Wang, Y. Shi, P. Sondhi, and H. Wang, "Modeling sequential online interactive behaviors with temporal point process," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 873–882.

[38] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in *Advances in Neural Information Processing Systems*, 2015, pp. 3492–3500.

[39] T. Bai, L. Zou, W. X. Zhao, P. Du, W. Liu, J.-Y. Nie, and J.-R. Wen, "Ctrec: A long-short demands evolution model for continuous-time recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 675–684.

[40] B. Vassøy, M. Ruocco, E. de Souza da Silva, and E. Aune, "Time is of the essence: a joint hierarchical rnn and point process model for time and item predictions," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 591–599.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[42] X. Zhao, Y. Chen, J. Guo, and D. Zhao, "A spatial-temporal attention model for human trajectory prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 965–974, 2020.

[43] S. Chen, B. Wang, X. Tan, and X. Hu, "Embedding attention and residual network for accurate salient object detection," *IEEE transactions on cybernetics*, vol. 50, no. 5, pp. 2050–2062, 2020.

[44] X. Wang and H. Duan, "Hierarchical visual attention model for saliency detection inspired by avian visual pathways," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 540–552, 2019.

[45] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, and A. Aït-Bachir, "Period-aware content attention rnns for time series forecasting with missing values," *Neurocomputing*, vol. 312, pp. 177–186, 2018.

[46] J. Song, J. Xiao, F. Wu, H. Wu, T. Zhang, Z. M. Zhang, and W. Zhu, "Hierarchical contextual attention recurrent neural network for map query suggestion," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1888–1901, 2017.

[47] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[48] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 335–344.

[49] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. AAAI Press, 2017, pp. 3119–3125.

[50] J. Han, L. Zheng, Y. Xu, B. Zhang, F. Zhuang, S. Y. Philip, and W. Zuo, "Adaptive deep modeling of users and items using side information for recommendation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 737–748, 2020.

[51] M. Shang, X. Luo, Z. Liu, J. Chen, Y. Yuan, and M. Zhou, "Randomized latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 131–141, 2019.

[52] X. Luo, M. Zhou, S. Li, D. Wu, Z. Liu, and M. Shang, "Algorithms of unconstrained non-negative latent factor analysis for recommender systems," *IEEE Transactions on Big Data*, pp. 1–1, 2019.

[53] D. Wang, S. Deng, and G. Xu, "Sequence-based context-aware music recommendation," *Information Retrieval Journal*, vol. 21, no. 2-3, pp. 230–252, 2018.

[54] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[55] L. A. Adamic and B. A. Huberman, "Power-law distribution of the world wide web," *Science*, vol. 287, no. 5461, pp. 2115–2115, 2000.

[56] D. Wang, S. Deng, S. Liu, and G. Xu, "Improving music recommendation using distributed representation," in *Proceedings of the 25th International Conference Companion on World Wide Web*, 2016, pp. 125–126.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[58] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[59] L. Cui, J. Wu, D. Pi, P. Zhang, and P. Kennedy, "Dual implicit mining-based latent friend recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 5, pp. 1663–1678, 2020.

[60] B. Bai, Y. Fan, W. Tan, and J. Zhang, "Dltsr: A deep learning framework for recommendation of long-tail web services," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 73–85, 2020.

Dongjing Wang (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2012 and 2018, respectively. He was co-trained at the University of Technology Sydney, Ultimo, NSW, Australia, for one year. He is currently a Lecturer with Hangzhou Dianzi University, Hangzhou. His current research interests include recommender systems, machine learning, and business process management.

Xin Zhang received the bachelor's and Ph.D. degrees in computer science and technology from Shandong University, Jinan, China, in 2012 and 2018, respectively. She was co-trained at the University of California at Davis, Davis, CA, USA, for one year. She is currently a Lecturer with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China. Her research interests include machine learning, image processing and computer vision.

Zhengzhe Xiang received the B.S. and Ph.D. degrees in computer science and technology from Zhejiang University, Hangzhou, China, in 2013 and 2020, respectively. He is currently a Lecturer with Zhejiang University City College, Hangzhou. His research interests include the fields of service computing, cloud computing, and edge computing.

Dongjin Yu (Senior Member, IEEE) is currently a Professor with Hangzhou Dianzi University, Hangzhou, China, where he is also the Director of the Institute of Big Data and the Institute of Computer Software. His research efforts include big data, business process management, and software engineering.

Dr. Yu is also a member of ACM and a Senior Member of the China Computer Federation (CCF). He is also a member of the Technical Committee of Software Engineering of CCF and the Technical Committee of Service Computing of CCF.

Guandong Xu (Member, IEEE) is a Full Professor in Data Science at School of Computer Science and Advanced Analytics Institute, University of Technology Sydney with PhD degree in Computer Science. His research interests cover Data Science, Data Analytics, Recommender Systems, Web Mining, User Modelling, NLP, Social Network Analysis, and Social Media Mining. He has published three monographs in Springer and CRC press, and 190+ journal and conference papers including TNNLS, TSC, IJCAI, AAAI, WWW, ICDE, and CVPR conferences.

He is the assistant Editor-in-Chief of World Wide Web Journal and has been serving in editorial board or as guest editors for several international journals. He has received a number of Industry Awards from Australian industry community, such as 2018 Top-10 Australian Analytics Leader Award.

Shuiguang Deng (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

He was a Visiting Scholar with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2014, and Stanford University, Stanford, CA, USA, in 2015. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. He has published over 80 articles in journals, such as the IEEE Transactions on Computers (TOC), the IEEE Transactions on Parallel and Distributed Systems (TPDS), the IEEE Transactions on Services Computing (TSC), the IEEE Transactions on Cybernetics (TCYB), and the IEEE Transactions on Neural Networks and Learning Systems, and refereed conferences. His research interests include service computing, mobile computing, and business process management.

Dr. Deng is an Associate Editor of IEEE ACCESS and the IET Cyber-Physical Systems: Theory & Applications.