IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 March 2023, accepted 31 March 2023, date of publication 5 April 2023, date of current version 8 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3264550

RESEARCH ARTICLE

KEMIM: Knowledge-Enhanced User Multi-Interest Modeling for Recommender Systems

FAN YANG[1], YONG YUE[1], GANGMIN LI[2], TERRY R. PAYNE[3], AND KA LOK MAN[1], (Member, IEEE)

[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China
[2]School of Computer Science and Technology, University of Bedfordshire, LU2 8DL Luton, U.K.
[3]Department of Computer Science, University of Liverpool, L69 7ZX Liverpool, U.K.

Corresponding author: Yong Yue (yongyue@xjtlu.edu.cn)

This work was supported in part by the Basic Public Welfare Research Project of Zhejiang, China, under Grant LGF20G020001; in part by the AI University Research Centre (AI-URC) through the Xi'an Jiaotong-Liverpool University Key Program Special Fund under Grant KSF-A-17; and in part by the Suzhou Municipal Key Laboratory for Intelligent Virtual Engineering under Grant SZS2022004.

ABSTRACT Researchers typically leverage side information, such as social networks or the knowledge graph, to overcome the sparsity and cold start problem in collaborative filtering. To tackle the limitations of existing user interest modeling, we propose a knowledge-enhanced user multi-interest modeling for recommender systems (KEMIM). First, we utilize the user-item historical interaction as the knowledge graph's head entity to create a user's explicit interests and leverage the relationship path to expand the user's potential interests through connections in the knowledge graph. Second, considering the diversity of a user's interests, we adopt an attention mechanism to learn the user's attention to each historical interaction and each potential interest. Third, we combine the user's attribute features with interests to solve the cold start problem effectively. With the knowledge graph's structural data, KEMIM could describe the features of users at a fine granularity and provide explainable recommendation results to users. In this study, we conduct an in-depth empirical evaluation across three open datasets for two different recommendation tasks: Click-Through rate (CTR) prediction and Top-K recommendation. The experimental findings demonstrate that KEMIM outperforms several state-of-the-art baselines.

INDEX TERMS Multi-interest, user modeling, knowledge graph, recommender systems.

I. INTRODUCTION

The rapid development of the Internet has brought an explosive growth in data, resulting in a challenge for users to find the content they want, often referred to as the information overload problem [1]. As an effective method to alleviate this problem, recommender systems play a vital role in online services by analyzing users' historical behavior data to determine user preferences and to recommend personalized content.

Nowadays, most recommendation methods can be characterized by five general approaches [2], [3]: (i) content-based recommendation; (ii) collaborative filtering; (iii) matrix factorization; (iv) hybrid recommendation; and (v) Deep Learning-based recommendation. Content-based approaches [4] typically generate recommendation results by calculating the similarity of extracted item features; however the performance can be significantly reduced if the item content attribute is missing. The underlying principle behind collaborative filtering [5] is to calculate the similarity between users or items and then generate personalized recommendations according to thia similarity. Although collaborative filtering has achieved some success in the industry, there is still scope for considerable improvement, such as the need for more interactive data, the difficulty in expanding user interests, and the challenges associated with the cold start problem, resulting from the inclusion of new users or items within the recommendation system. Utilizing the sparsity of the score matrix, Matrix Factorization [6] employs a row-column transformation to learn the feature matrix of users and items

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

while assuming that user preferences are mostly influenced by a small number of critical factors. Although Matrix Factorization successfully addresses the issue of missing values for the user and item scoring matrix, the decomposed matrix cannot correspond to concepts in the real world, and can only be understood as abstract semantic spaces. This can make it difficult to provide reasonable explanations for the resulting recommendations. In practical applications, different models are often combined or fused synergistically to achieve better results, compared to using a specific model in isolation [7]. Although there are many such fusion strategies used for generating recommendations, there is no guarantee that any fused method will perform better than the original method when solving a specific problem. The way in which a fused model can be optimized should also be considered. Deep learning-based recommendation approaches [8] use Neural Networks to model user preferences and generate personalized recommendations. However, due to the need for adequate auxiliary information, cold start and interpretability remain significant challenges.

In 2012, Google proposed the concept of a ''knowledge graph''. Since then, the use of knowledge graphs has become widespread across a variety of industries and application areas, including search, recommendation, question answering, text understanding, and text generation. Thus, a recommender system can benefit from the knowledge graph's construction of the relationships between entities with explicit semantic links. These include: (i) enhancing user-item and item-item connections to reduce the sparsity of user behavior; (ii) enhancing item attributes to improve the accuracy of recommendation learning and the completion of item representations; and (iii) making an explainable recommendation using the knowledge graph's semantic relationships. Currently, knowledge graph-based recommendation algorithms can be divided into two broad categories: embedding-based methods [9], [10], [11], [12], [13], [14] and path-based methods [15], [16], [17], [18], [19]. The embedding-based methods utilize the representation vectors of nodes in the learned knowledge graph to determine the similarity between users and items for recommendations. These methods only consider the direct relationship between entities and require a multi-level relationship between entities. Path-based methods make recommendations by extracting and modeling the path between users and items within the knowledge graph itself. However, the models are often complex and need to be more scalable. Additionally, recommendation methods based on knowledge graphs only capture the explicit relationship features between users and items that exist in the knowledge graph itself, as it can be challenging to extract the implicit relationship features outside the knowledge graph.

To provide personalized services and improve recommendation performance, we propose a Knowledge-Enhanced user Multi-Interest Modeling for recommendation (KEMIM), which uses a knowledge graph to enrich and mine the potential information of users and items, and organically combines embedding-based and path-based recommendation methods to more reasonably express user interests. Our main contributions are as follows:

- We model user interests across several dimensions. We model explicit user interest by the user-item interaction and expand potential user interest by linking the relevant entities corresponding to the knowledge graph's relationship path to improve the recommendation's accuracy. At the same time, we use the semantic relationships defined within the knowledge graph to provide explainable recommendation results.
- We use the attention mechanism to model user behaviors on explicit and potential interest. We analyze the impact of different interactions on user preferences and effectively extract dynamic preferences in the user interaction process.
- We combine user attribute features with user interest features to solve the cold start problem, and improve recommendation performance.
- The proposed KEMIM system outperforms various state-of-the-art recommendation approaches across diverse settings, based on an empirical analysis across three open datasets. An additional ablation analysis supports the validity of our main contributions.

II. RELATED WORK

A. USER MULTI-INTEREST MODELING

User multi-interest modeling is a type of user interest modeling that focuses on predicting and analyzing the multiple interests of users; in contrast to the more traditional user interest modeling, which typically focuses on a single interest. Thus, user multi-interest modeling aims to understand a user's full range of interests, and studies have shown that user multi-interest modeling can enhance the effectiveness of personalized recommender systems. By considering multiple interests, these systems can provide a more diverse and comprehensive set of recommendations, leading to improved user satisfaction and engagement. Thus, this approach has received significant attention from both industry and academia, resulting in numerous methods for modeling using user behavior data.

Li et al. [23] proposed a novel method to learn user representations by representing a user with many representation vectors that encode various aspects of the user's interests. Xiao et al. [24] presented a Deep Multi-Interest Network (DMIN) that predicts click-through rates by modeling users' latent multiple interests, whereas a novel framework called Multi-Interest User Representation Model was proposed by Yang et al. [25]. As a recommendation model for news, Wang et al. [26] proposed a multi-interest news sequence (MINS). Using the advantages of multi-interest learning, Chai et al. [27] improve candidate matching performance by utilizing user profiles. They presented the User-Aware Multi-Intention Learning Framework (UMI) for assessing candidates based on user profiles and behavioral data. Portman et al. [28] introduced MiCRO, a generative statistical framework that models the preferences of multi-interest users and the representations of items that reflect multi-interest preferences over time. The Knowledge Enhanced Multi-Interest Network (KEMI) [29] utilizes knowledge graphs to learn users' interest representations using heterogeneous graph neural networks (HGNNs) and a unique dual memory network (Long- and Short-Term) which is extremely useful in capturing user interests. Yang et al. [30] provided a hierarchical model of user intentions and preferences based on relation-aware heterogeneous information network (HIN) embeddings for sequential recommendation to account for users' multiple interests.

A user's numerous possible interests need to be distinguished or accurately modeled by most existing approaches, which makes it difficult to propose the next item accurately.

B. KNOWLEDGE GRAPH-BASED RECOMMENDATION

Knowledge graph-based recommendation refers to using knowledge graphs within personalized recommendation systems, such that a knowledge graph helps to enhance the personalized recommendation systems' accuracy, relevance, and explainability [37], [38]. Such knowledge graph-based recommendations mainly exploit a knowledge graph representation or knowledge graph meta-path extraction. The methods based on knowledge graph representation mainly use the original dataset to construct the knowledge graph and represent the nodes and relations in the vector space, which is computed to generate low-dimensional dense vectors. Zhang et al. [31] used embeddings to encode interaction information, text, and images, and from these, they extracted multiple semantic features fused with the recommended items. Wang et al. [32] proposed a collaborative deep learning (CDL) model, which uses knowledge graph representation learning to obtain structured information about items, and uses a denoising encoder network to learn textual representation vectors at the coding layer. Subsequent work [33] resulted in the proposal of a Deep knowledge-aware network model that uses knowledge graph representation techniques and convolutional neural networks to learn sentences, and adds attention mechanisms to achieve news recommendations.

Those methods based on knowledge graph meta-path extraction differ as they mainly use the connectivity between nodes to extract path information and combine it with a collaborative filtering model to make recommendations. Qian et al. [34] proposed a new path similarity calculation method, PathSim, by calculating the path similarity between nodes. Shi et al. [17] proposed a meta-path-based strategy to calculate the path similarity using weighted meta-paths and constructed a regularized term loss function with weights for model training. The RippleNet model [9] adds the path and node information in the knowledge graph as additional knowledge to the recommendation algorithm and understands the user's preference for an item as a water wave propagation. However, these models could have effectively solved the problems of algorithm interpretability and cold start.

III. METHODOLOGY

A. PROBLEM DEFINITION

Definition 1: The user-item interaction matrix $\mathbb{Y} = \{y_{ui} | u \in \mathbb{U}, i \in \mathbb{I}.\ \mathbb{U} = \{u_1, u_2, \ldots\}$ and $\mathbb{I} = \{i_1, i_2, \ldots\}$ represent the set of users and the set of items.

Where $y_{ui} = 1$ indicates implicit feedback between user u and item i, e.g., click, view, collect.

Definition 2: Knowledge Graph $\mathcal{G}$, which is composed of a large number of entity-relationship-entity triples (head entity, relationship, tail entity). $\mathcal{G} = \{(e_h, r, e_t) | e_h \in \mathcal{E}, r \in \mathcal{R}, e_t \in \mathcal{E}\}$, $\mathcal{E}$ and $\mathcal{R}$ represents the set of entities and the set of relationships in knowledge graph $\mathcal{G}$. Meanwhile, item i in the set $\mathbb{I}$ can be matched with one or more entities in $\mathcal{G}$, which can be expressed as $\mathbb{I} \subseteq \mathcal{E}$.

Definition 3: User u's historical interaction item set is $\delta_u = \{i_1, i_2, \ldots, i_{N_j}\}$.

Definition 4: The set of entities related to the user u's interacted items is $\mathcal{E}_u = \{e_t \mid (e_h, r, e_t) \in \mathcal{G}, e_h \in \delta_u\}$.

Definition 5: The related triplet set of user u is $\mathcal{S}_u = \{(e_h, r, e_t) \mid (e_h, r, e_t) \in \mathcal{G}, e_h \in \delta_u\}$.

To sum up, we can describe the recommendation problem as follows: given interaction matrix $\mathbb{Y}$, knowledge graph $\mathcal{G}$, and user attribute information, for any pair of u, i, learn its prediction probability $\hat{y}_{ui} = \mathcal{F}(u, i; \theta) \in [0, 1]$, where θ represents the parameter of function $\mathcal{F}$.

B. MODEL FRAMEWORK

Our proposed KEMIM is outlined in Figure 1, which consists of four critical modules: (i) Explicit user interests; (ii) Potential user interests; (iii) User attributes; and (iv) Fuse user interest representation and attribute representation to generate recommendation results. Specifically, the model initially takes the item knowledge graph and user attributes as auxiliary information, and the user-item historical interaction as the explicit interests U_E, and then links related entities through the relationship structure of the knowledge graph as the potential interests U_P, before using the attention mechanism to model them respectively. Meanwhile, to address the cold start challenge, the user attribute features are embedded as U_A. The resulting user embedding is obtained by fusing the user interest representation and attribute representation. As a final step, we concatenate the user embedding U and the candidate item embedding I_j together and feed it into the neural network and the sigmoid function to calculate the click probability $\hat{y}_{ui_j}$.

C. EXPLICIT USER INTEREST MODELING

Modeling user interests is the basis of many recommendation systems, and capturing more user preference information is crucial if the accuracy of recommendations is to be enhanced. The user's historical behavior involves a variety of features, e.g., clicking, browsing, and buying. The recommender system based on deep learning represents these historical behaviors through the embedding layer. It uses combination methods to fuse them into a vector that represents user

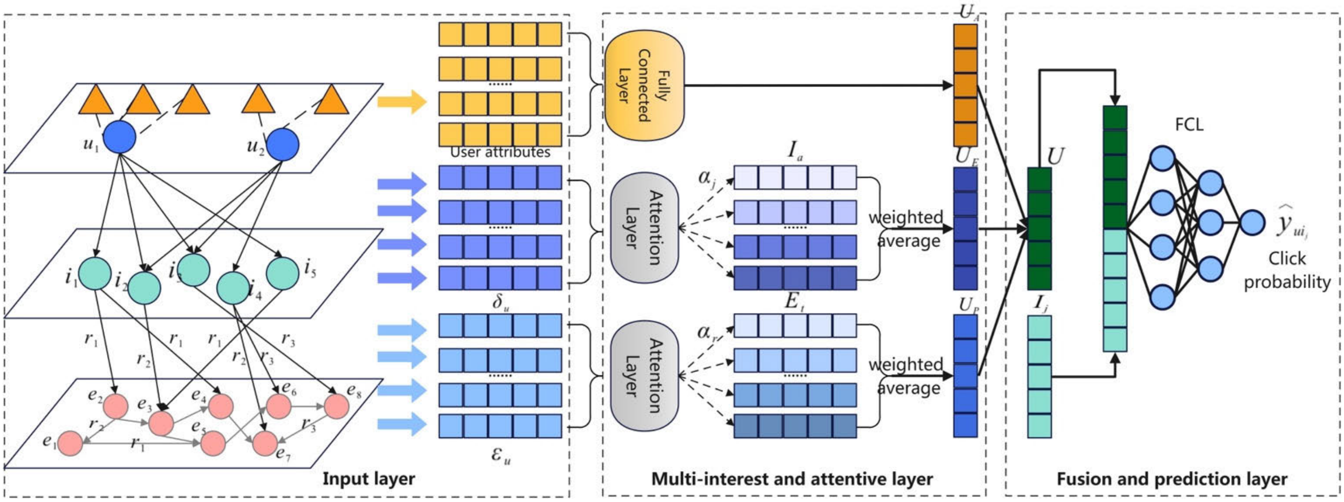

FIGURE 1. The overall framework of KEMIM.

interests. Our approach uses the user-item historical interaction to extract the explicit relationship features between users and items through the attention network as its explicit interest representation. We assume a historical interaction set δ_u of user u, and since each item matches an entity in the knowledge graph, for each item i_a $(a = 1, 2, \ldots, N_a)$, we learn the corresponding vector representation $I_a \in \mathbb{R}^d$ through knowledge representation learning, where d is the size of a vector representation.

The user-item interaction is diversified. The user will be more interested in this candidate item if many items are similar to some candidate item in the historical interaction set. Therefore, we use the attention mechanism to calculate the relevant weight between each interactive item and the candidate item. Given this weight, the vector of each historical item is weighted and summed to represent the user's historical interest dynamically. Specifically, based on the candidate item representation $I_j \in \mathbb{R}^d$, each i_j in the user u interaction history item set δ_u is assigned a different weight and weighted average to obtain the user's explicit interest representation U_E:

$$U_E = att\left(I, I_a\right) = \frac{1}{N_a}\sum\nolimits_{a=1}^{N_a} \alpha_j I_a \tag{1}$$

where α_j is the weight factor of attention. We can use function $\rho\left(x, y\right)$ to fit the correlation between item i_j and i_a in the form of an inner product and convert the correlation into a weight factor through *the softmax* function:

$$\alpha_j = softmax\left(\rho\left(I_a; I_j\right)\right) = \frac{exp\left(\rho\left(I_a; I_j\right)\right)}{\sum_{a=1}^{N_a} exp\left(\rho\left(I_a; I_j\right)\right)} \tag{2}$$

D. POTENTIAL USER INTEREST MODELING

The knowledge graph contains rich entity information, and the association between different entities can be used to mine potential user interests. Consider the scenario where user m has clicked on movie i_m because he is interested in the leading actor in this movie. To discover the potential user interests, we use the knowledge graph to link the items that users have interacted with to related entities through the relationship path of the knowledge graph to mine the potential interests of the related entities. Considering the different degrees of user interests in different relationships, we put the knowledge path representation vectors into the attention network to calculate the items that users have interacted with to their related entities through the relationship path.

Given the knowledge graph $\mathcal{G}$, we take the user u's historical interaction item set δ_u as the head entity and relationship along the knowledge path to get the related entities $\mathcal{E}_u$ and triplet set $\mathcal{S}_u$ of historical items. The inner product function $\rho(x, y)$ is used to calculate the weight factor α_r of candidate item i_j and the triple $\left(e_{h_j}, r_j, e_{t_j}\right)$, under the relation r_j:

$$\begin{aligned}\alpha_r &= softmax\left(\rho\left(I_j; E_{h_j}, R_j\right)\right)\\ &= \frac{exp\left(\rho\left(I_j; E_{h_j}, R_j\right)\right)}{\sum_{\left(e_{h_j}, r_j, e_{t_j}\right)\in S_u} exp\left(\rho\left(I_j; E_{h_j}, R_j\right)\right)}\end{aligned} \tag{3}$$

where $R_j \in \mathbb{R}^d$ is the vector representation of relation r_j, E_{h_j} is the vector representation of the head entity matched by the item. By calculating the relevant weight factors of all triples in S_u, the relevant entities that users link to are weighted average to represent user potential interests U_P:

$$U_P = att\left(\left(R_j, E_{t_j}\right), I\right) = \sum\nolimits_{\left(e_{h_j}, r_j, e_{t_j}\right)\in S_u} \alpha_r E_{t_j} \tag{4}$$

E. USER ATTRIBUTES MODELING

Traditional recommendations ignore the attribute aspects of users themselves and concentrate on the interaction between users and items. We comprehensively consider the features of user attributes and user interests to identify users better, and help solve the cold start problem for new users.

Attribute features (based on the inherent information of users) can be used to represent users effectively when users have no interaction with items. To extract these attribute features for each user, we convert the attribute information into sparse vectors through the one-hot encoder. Because the vectors of one-hot type are too sparse, it will lead to too many parameters and different lengths of vectors, which is not conducive to the fusion of features. We therefore use the embedding layer to compress the sparse vector representation of attribute features into a low dimensional dense vector of uniform length, and then map different features to the same hidden space with user interest features through the fully connected layer mapping function $\mathcal{Z}$. User attributes representation U_A is given as:

$$U_A = \mathcal{Z}\left(U_{age}, U_{gender}, U_{loc}, U_{job}, \ldots\right) \tag{5}$$

F. RECOMMENDATION GENERATION AND OPTIMIZATION

User vector representations can be obtained by fusing explicit and potential user interests and integrating user attribute features, as follows:

$$\boldsymbol{U} = U_E + U_P + U_A \tag{6}$$

Given the vector representation $\boldsymbol{U}$ of user u and the vector representation I_j of candidate items i_j, and given the fact that the simple inner product is not enough to mine the complex non-linear association between user features and item features, we use neural network $\mathcal{F}$ to get $\grave{y}_{ui_j}$ corresponding to the probability of their interaction:

$$\begin{aligned}\grave{y}_{ui_j} &= \mathcal{F}\left(\mathbf{U}; I_j\right)\\ &= \sigma\left(a_L\left(W_L^T\left(\cdots a_2\left(W_2^T a_1\left(W_1^T f_{\text{concat}}\left(U, I_j\right) + b_1\right)\right.\right.\right.\right.\\ &\qquad\left.\left.\left.\left. + b_2\right)\cdots\right) + b_L\right)\right)\end{aligned} \tag{7}$$

where $\sigma(x) = \left(1 + e^{-x}\right)^{-1}$ is a *sigmoid* function, a_n represents the activation function of the n^{th} layer in the neural network, and W_n and b_n are the weight and bias of the n^{th} layer, respectively.

G. MODEL TRAINING

The loss function of KEMIM is shown in Formula (8):

$$\mathcal{L} = \mathcal{L}_{RS} + \mathcal{L}_{KG} + \mathcal{L}_{REG} \tag{8}$$

The probability distribution of the target user interacting with candidate items is output through the *softmax* function in the neural network, and the gap is between the predicted value $\grave{y}_{uv}$ and the actual score y_{uv}:

$$\mathcal{L}_{RS} = -\sum y_{uv} log\, \grave{y}_{uv} + (1 - y_{uv})\, log\left(1 - \grave{y}_{uv}\right) \tag{9}$$

We use the random gradient descent method to optimize the loss function iteratively. Compared with other loss functions, the curve of cross-entropy loss is monotonic—the more significant the loss, the greater the gradient, which is convenient for model optimization.

We give the observed interactions a greater reward than the unobserved interactions for the loss in the second item:

$$\mathcal{L}_{KG} = \sum\nolimits_{(u,i,j)\in O} -ln\, \sigma\left(\grave{y}_{ui} - \grave{y}_{uj}\right) \tag{10}$$

where O is the training set, there are the observed and unobserved interactions between users and items.

Finally, for the regularization term $\mathcal{L}_{REG}$, we take all the framework parameters into account using L_2 regularization:

$$\mathcal{L}_{REG} = \lambda_1 \parallel W \parallel_2^2 + \frac{\lambda_2}{2}(\|V\|_2^2 + \|E\|_2^2 + \|R\|_2^2) \tag{11}$$

where λ_1 and λ_2 are the balancing parameters.

IV. EXPERIMENTS

A comparative evaluation of KEMIM with other recommendation based approaches was conducted across three real-world scenarios; recommendations for movies, books, and music. Before presenting the experimental findings, we first describe the datasets, baselines, and experimental setup, as well as completing an ablation experiment to validate each module in KEMIM. The selection of hyper-parameters is also discussed.

A. EXPERIMENTS SETTINGS

1) DATASETS

For recommendation data, we use the MovieLens-20M, Book-Crossing and Last.FM datasets. MovieLens-20M is a frequently used dataset for recommendation tasks, comprising 130000 users, more than 27000 movies, and nearly 20 million movie rating data points. The Book-Crossing data set consists of user-book interactions that were collected from the Book-Crossing Community. It includes explicit and implicit feedback from more than 10000 users, as well as one million ratings on more than 19000 books. Last.FM is a dataset of user-music interactions taken from the Last.FM music service, which includes approximately 2000 users and nearly 20,000 musical scores from almost 100,000 musical compositions.

In addition to user-item interaction, it was necessary to construct a knowledge graph for entities/items in each dataset, using Microsoft Satori. This consisted of initially choosing the subgraphs of the relevant disciplines (such as the knowledge graphs of movies, books, and music), which have a confidence level greater than 0.9 throughout the whole knowledge graph. For each field, the names of the items in the dataset were compared to the names of the entities in the knowledge graph, and after this comparison, new IDs were given to all of the entities and relationships. Unrelated items, entities, and user-item interactions were then deleted, to keep things simple. Table 1 displays the fundamental data for the three datasets after having completed the entity matching process.

2) BASELINES

To evaluate the effectiveness of KEMIM, we compared it with a number of state-of-the-art baselines. Other than LibFM, all of the following baselines use knowledge graphs as auxiliary information:

CFKG [20] is a collaborative filtering recommendation system. It integrates user behavior and the knowledge graph

TABLE 1. Basic statistics of the three datasets.

Datasets		MovieLens-20M	Book-Crossing	Last.FM
user-item	# users	127,538	17,860	1,872
	# items	24,633	14,967	3,846
	# interaction	19,580,275	139,746	42,346
knowledge graph	# entities	98,097	19,876	9,270
	# relations	32	18	60
	# triples	472,241	57,103	15,492

of items into a unified graph and uses TransE to learn entity embedding in the graph.

LibFM [21] is a feature-based factorization model. As input for LibFM, we concatenate user ID, item ID, and the appropriate averaged entity embeddings learned through TransR.

RippleNet [9] is a network structure similar to water ripple propagation. The user preference is propagated to the entity set along the user-item interaction path using knowledge data from the knowledge graph. The user preferences are dispersed to generate the recommendation results.

MKR [22] is a multi-task feature learning method. It mainly embeds knowledge from the knowledge graph into the tasks to support the recommendation.

KGCN [10] aims to encode high-order dependent contexts with respect to the semantic data in the knowledge graph. The main goal of KGCN is to represent entities by aggregating messages with a neighborhood information bias.

KGAT [13] develops an attentive message-passing method across the knowledge-aware collaborative graph for embedding fusion. It employs an attention mechanism to discriminate the importance of the neighbors.

HKIPN [35] presents a new heterogeneous propagation method that simultaneously propagates knowledge and user interest in a user-item-knowledge graph. It considers information decay in the process of propagation.

CG-KGR [36] encapsulates historical interactions to construct an interactive information summarization, and adopts Collaborative Guidance Mechanism to extract information.

FIRE [39] models multi-granular high-order feature interactions by convolutional neural networks (CNNs) and the users' latent intent factors by utilizing a two-level attention mechanism to improve user and item representation learning.

3) EXPERIMENT SETUP

Our evaluation focuses on two recommendation tasks: Top-K recommendation and Click-Through Rate (CTR) prediction. To ensure reliable and consistent experimental results, we divided each dataset into training, evaluation, and test sets with a 6:2:2 ratio and performed the random split process five times.

We use the trained model in the Top-K recommendations to rank K items according to each user's highest predicted scores. To evaluate the Top-K recommendation performance of KEMIM, we uitilise two popular evaluation metrics: **Recall@K**(the percentage of relevant items the system selected in the top K items) and **NDCG@K** (Normalised Discounted Cumulative Gain for the top K items). We first transform $\hat{y}_{ui_j}$ through the sigmoid function in the CTR prediction process. The click rate is then determined by comparing the rescaled $\hat{y}_{ui_j}$ to a critical value of 50%,and categorizing it as either 1 or 0. To evaluate the prediction's accuracy, we adopt the **AUC** (area under the ROC curve) metric and **F1** (harmonic mean of recall and precision) metric.

For the hyper-parameters of our proposed model, we select the entity embedding dimension d and the relationship embedding dimension k from 8, 16, 32, 64, 128}. The maximum number of neighbors of the node K id is selected from the set 2, 4, 8, 16, 24, 32}. The selection range of regularization parameter λ is $10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 10^{-2}$, and learning rate η is tuned within $10^{-4}, 5 \times 10^{-4}, 10^{-3}, 10^{-2}, 5 \times 10^{-1}$. The model employs a batch training method and has a constant data size per batch of 512. All models are optimized utilizing the Adam optimization algorithm.

For the hyper-parameters of baselines, the embedding size of TransE in the CFKG is 32, the entity embedding size $d = 8$, and the learning rate $\eta = 0.1$. For RippleNet, $d = 8$, $\eta = 0.01$, $\lambda_1 = 10^{-6}$, $\lambda_2 = 0.01$, $H = 2$. For MKR, $d = 16$, $\eta = 0.01$, $\lambda_1 = 10^{-5}$, $\lambda_2 = 0.01$. For KGCN, $H = 2$, $d = 64$, $\eta = 0.01$, $\lambda_2 = 0.01$. For KGAT, $H = 2$, $d = 128$, $\eta = 0.01$, $\lambda_2 = 0.01$. The hyper-parameter settings of other baseline approaches are the same as those specified in the original paper or the provided code.

B. PERFORMANCE COMPARISON

Table 2 displays the comparison with the baseline approaches for the CTR prediction. The second-best performances are underlined, while the top results are shown in bold. The experimental comparison of NDCG and Recall for different K values in accordance with the Top-K guideline is shown in Figure 2.

From our empirical evaluations, we make the following set of observations:

- KEMIM generally outperforms all other approaches in terms of performance, and the performances on AUC and F1 are significantly improved compared with other baselines. Specifically, for the Movie dataset, these evaluation metrics increased by 0.13% - 17.31% and 0.15% - 18.45%, respectively. Likewise for the Book dataset, they increased by 1.20% - 23.55% and 0.77% - 16.56%, respectively. For Top-K recommendation, KEMIM has also achieved a good performance for NDCG@K and Recall@K values, which further demonstrates the effectiveness of this proposed model.
- The results obtained using KEMIM for the Book Crossing and MovieLens-20M datasets are better than that of the baseline approaches. This is because the data in the book dataset and Movie dataset are much sparser than

that in the music dataset. The performance of KEMIM remains effective when faced with sparse data scenarios.

- KEMIM comes in third place on the AUC metric for the Last.FM dataset. However, it does not exhibit more extraordinary performance on the F1 and Recall metrics. As the ratio of KG triplets to items in the music dataset is the lowest, our KEMIM cannot always successfully extract the entities of KG to enhance the item embeddings. Another reason is that we put the critical value at 50% to decide if item i would be offered to user u after normalizing the expected score $\hat{y}_{ui_j}$. As the distribution of positive and negative samples in music is uneven and restricted, a critical value of 50% may not be suitable for binary categorization.
- The performance of the baseline approaches using the knowledge graph is better than that of the approaches that do not use a knowledge graph. It supports the claim that that using the knowledge graph as additional information to the recommender system will help to improve the recommendation performance.
- MKR shares information through multi-task learning in the recommendation and knowledge representation, which can improve recommendation accuracy and optimize knowledge representation. However, exact knowledge representation makes the recommendation result too single, and the recall rate of the model decreases.
- HKIPN increases the effectiveness of the propagation process by simultaneously propagating knowledge and user interest in a unified graph. However, simply integrating KGs into recommendation models would not improve its performance; it may even reduce its capacity as an overall model.
- The fact that FIRE has the best performance in the music dataset could be attributed to its ability to effectively explore user latent intent and item fine-grained high-order feature interactions. However, it needs to improve its effectiveness in the movie and book datasets, which are much sparser. It was also noted that the time taken for recommendations was longer than other methods, which is an obvious shortcoming.
- CG-KGR and KGAT perform well in CTR prediction and are better than most baselines in Top-K recommendation. This is because they exploit end-to-end joint learning models that simultaneously use the knowledge graph's features and structural information. This has the advantage that it combines knowledge graph representation learning and recommendation tasks more efficiently. However, these methods need more modeling of user attribute information. The implicit user vector generated only by the knowledge graph and user-item interaction needs to be more comprehensive to fully describe user interest.

In brief, our proposed KEMIM uses user-item historical interaction and item-related entities in the knowledge graph to mine user interests at multiple levels. At the same time, KEMIM combines user attribute features to make up for the cold start defect, which significantly affects the improvement of recommendation performance.

TABLE 2. Average results of CTR prediction task.

Model	MovieLens-20M AUC (%)	MovieLens-20M F1 (%)	Book-Crossing AUC (%)	Book-Crossing F1 (%)	Last.FM AUC (%)	Last.FM F1 (%)
CFKG	89.24	78.95	61.32	57.36	63.27	58.09
LibFM	86.16	81.70	60.04	58.78	61.89	60.16
RippleNet	92.73	90.83	71.93	64.78	80.37	72.11
MKR	83.50	81.32	67.09	61.79	78.20	70.10
KGCN	90.37	90.12	67.14	62.08	78.92	68.37
KGAT	96.98	92.01	68.45	66.07	81.63	74.29
HKIPN	96.80	92.90	73.30	65.40	80.50	70.10
CG-KGR	97.82	93.38	73.18	66.35	81.00	71.24
FIRE	92.50	85.10	66.29	61.74	**82.00**	**75.20**
KEMIM	**97.95**	**93.52**	**74.18**	**66.86**	81.27	71.35
% Gain	0.13%	0.15%	1.20%	0.77%	N/A	N/A

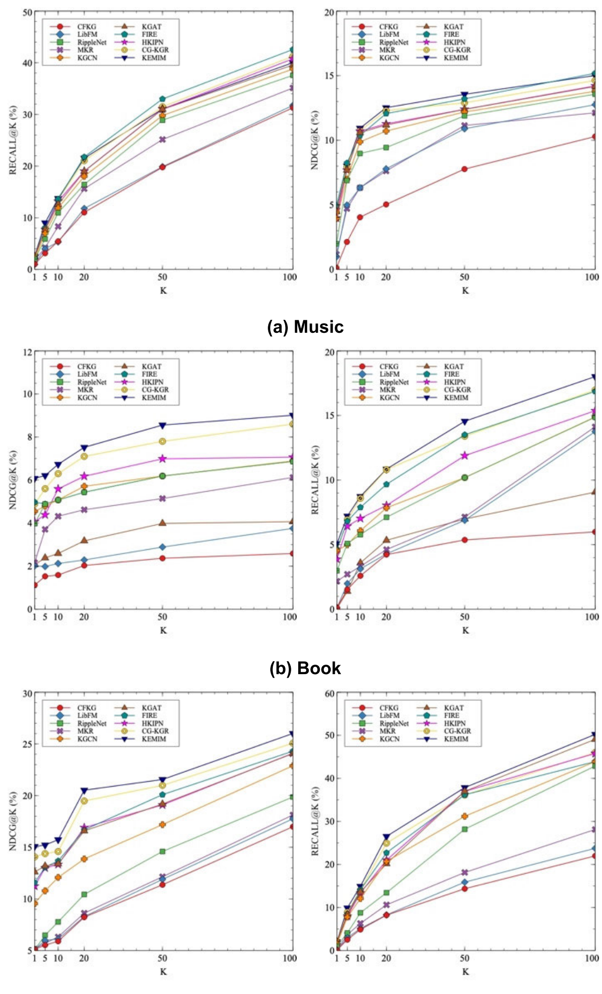

FIGURE 2. Average results of Recall@K and NDCG@K on Top-K recommendation.

C. ABLATION STUDY

To discuss the impact of different modules in our proposed model, we designed some variants for use in conducting

TABLE 3. Sensitivity analysis of d and λ.

Parameter	$d = 4$		$d = 8$		$d = 16$		$d = 32$		$d = 64$	
	AUC	R@10	AUC	R@10	AUC	R@10	AUC	R@10	AUC	R@10
$\lambda = 0.001$	93.45	13.97	95.41	13.99	96.67	14.02	96.58	14.02	96.01	14.00
$\lambda = 0.005$	93.78	14.11	95.08	14.20	96.78	14.35	96.62	14.30	96.07	14.15
$\lambda = 0.01$	93.89	14.32	95.58	14.47	97.95	14.73	97.24	14.56	96.86	14.43
$\lambda = 0.05$	93.63	14.27	95.27	14.39	97.75	14.58	97.18	14.43	96.53	14.31
$\lambda = 0.1$	93.32	14.19	95.12	14.24	97.63	14.37	97.03	14.29	96.17	14.21
$\lambda = 0.5$	93.28	14.10	94.98	14.10	97.58	14.12	96.95	14.10	95.89	14.08
$\lambda = 1$	92.81	14.03	95.01	14.04	97.37	14.06	96.87	14.05	95.64	14.04

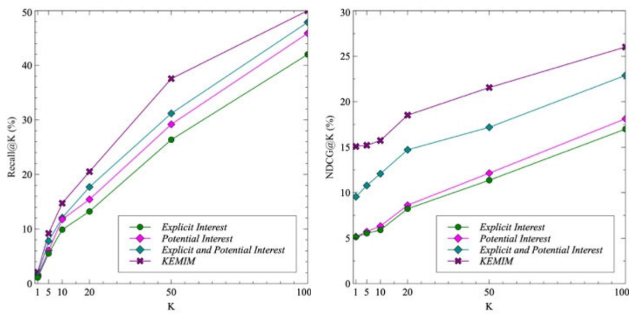

FIGURE 3. Ablation study of the components on Top-K recommendation.

ablation experiments: explicit interest, potential interest, and both. We use different variants to represent users and compare them with KEMIM. Experiments were conducted by varying the number of recommended items K with the values 1,5,10,20,50,100, for the experimental dataset MovieLens-20M. Figure 3 shows the values of Recall@K and NDCG@K, respectively.

Potential interest in using knowledge graph structure information has significantly improved the recall rate. The results of the two metrics are improved by combining explicit interest with potential interest compared with individual modules. When the value of K is small, the NDCG value with attribute information increases significantly, suggesting that the fusion of attribute features can improve the overall recommendation performance to a certain extent. KEMIM, which combines explicit interests, potential interests, and user attributes, is superior to the other variants considered in this empirical study across each metric.

We conducted a comparative evaluation to investigate the impact of attention networks on recommendations. The first involved retaining the attention layer to capture the degree of user interest, whereas the other involved removing the attention layer and directly embedding nodes by weighted averaging. The Recall@K values of these two models across different datasets is shown in Figure 4.

From the above results, the recommendation performance of the model retaining the attention network is equivalent to or better than that obtained when removing the attention network. The more the value of K, the more pronounced the recommendation improvement by the attention network. This comparative evaluation shows that the attention mechanism

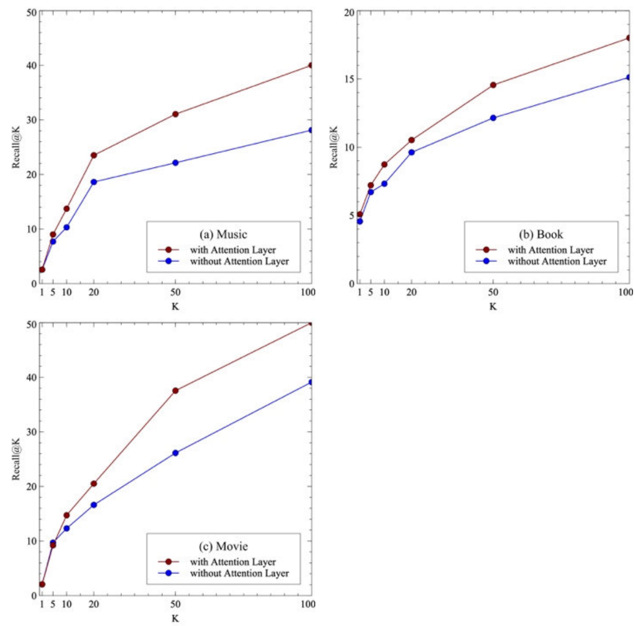

FIGURE 4. Ablation study of the attention mechanism on Top-K recommendation.

used in KEMIM can effectively improve the performance of recommendations.

D. PARAMETER ANALYSIS

We conducted an analysis to verify the values of the vector representation dimension d and the training weight λ of knowledge graph feature learning. Values for d ranged from 4 to 64, the values for λ ranged from 0.001 to 1, and the other parameters remained constant. Table 3 shows in the MovieLens-20M dataset, with different dimensions and different λ, the value of AUC (%) and R@10 (%). It has a similar trend in the other two datasets. We can see that when λ is constant, AUC and Recall increase with d as vectors with larger dimensions can encode more helpful information. However, when d is greater than 16, both AUC and Recall begin to decline possibly because of overfitting. In addition, when $\lambda = 0.01$, AUC and Recall perform best. Because when it is smaller than 0.01, it cannot provide practical regularization constraints, and when it is more significant than 0.01, it can have an enormous impact on the objective function.

V. CONCLUSION

The knowledge graph is a powerful mechanism for intelligent recommendation, but the existing recommendation methods combined with the knowledge graph have many apparent shortcomings. We propose a knowledge-enhanced user multi-interest modeling for recommendation (KEMIM), which models user interests across three different dimensions: explicit interest, potential interest, and user attributes, making full use of the rich semantics and distinctive network structure contained in the knowledge graph, to mine users' multi-level interests and address the problem of cold start. Additionally, we adopt the attention mechanism to weigh the typical user's explicit and prospective interests to reflect the diversity of interests, considering that various activities influence user interests. Finally, we provide the optimization strategy and loss function and test it over three open datasets. By comparing the performance of existing state of the art models and ablation experiments, we could confirm that the model has significant effects on the recommendation performance and analyze the impact of each module and parameters used in the model.

Future work will seek to create more effective and explainable recommendation models based on knowledge graphs and graph neural networks. Mainly, we will focus on simplifying and improving the knowledge extraction in the KG-based recommender system, which is one shortcoming of our work. In addition, we will incorporate self-supervised learning into KG-based recommendations.

REFERENCES

[1] L. W. Huang, B. T. Jiang, and S. Y. Lu, "Review of recommendation systems based on deep learning," *Chin. J. Comput.*, vol. 41, no. 7, pp. 1619–1647, 2018.

[2] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers Comput. Sci.*, vol. 14, pp. 1–21, Aug. 2019.

[3] X. Liang, *Actual Combat of Recommendation System*. Beijing, China: People's Posts and Telecommunications Press, 2012.

[4] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Germany: Springer, 2015.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[7] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[8] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for Youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.

[9] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 417–426.

[10] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, May 2019, pp. 3307–3313.

[11] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 968–977.

[12] W. L. Hamilton, R. Ying, and L. Jure, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[13] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 950–958.

[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018.

[15] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, Feb. 2014, pp. 283–292.

[16] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 917–922.

[17] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 453–462.

[18] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 297–305.

[19] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5329–5336.

[20] Y. Zhang, Q. Ai, X. Chen, and P. Wang, "Learning over knowledge-base embeddings for recommendation," 2018, pp. 1–4, *arXiv:1803.06540*.

[21] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–22, May 2012.

[22] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *Proc. World Wide Web Conf.*, May 2019, pp. 2000–2010.

[23] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2615–2623.

[24] Z. Xiao, L. Yang, W. Jiang, Y. Wei, Y. Hu, and H. Wang, "Deep multi-interest network for click-through rate prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2265–2268.

[25] B. Yang, K. Liu, X. Xu, R. Xu, Q. Sun, H. Liu, and H. Xu, "User multi-interest modeling for behavioral cognition," 2021, *arXiv:2110.11337*.

[26] R. Wang, S. Wang, W. Lu, and X. Peng, "News recommendation via multi-interest news sequence modelling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 7942–7946.

[27] Z. Chai, Z. Chen, C. Li, R. Xiao, H. Li, J. Wu, J. Chen, and H. Tang, "User-aware multi-interest learning for candidate matching in recommenders," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1326–1335.

[28] F. Portman, S. Ragain, and A. El-Kishky, "MiCRO: Multi-interest candidate retrieval online," 2022, *arXiv:2210.16271*.

[29] D. Liu, Y. Yang, M. Zhang, W. Wu, X. Xie, and G. Sun, "Knowledge enhanced multi-interest network for the generation of recommendation candidates," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 3322–3331.

[30] F. Yang, G. Li, and Y. Yue, "Hierarchical user intention–preference for sequential recommendation with relation-aware heterogeneous information network embedding," *Big Data*, vol. 10, no. 5, pp. 466–478, Oct. 2022.

[31] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362.

[32] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1235–1244.

[33] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," 2018, *arXiv:1801.08284*.

[34] F. Qian, S. Zhao, J. Tang, and Y. Zhang, "SoRS: Social recommendation using global rating reputation and local rating similarity," *Phys. A, Stat. Mech. Appl.*, vol. 461, pp. 61–72, Nov. 2016.

[35] Q. Chen, H. Tan, G. Lin, and Z. Wang, "A hierarchical knowledge and interest propagation network for recommender systems," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2020, pp. 119–126.

[36] Y. Chen, Y. Yang, Y. Wang, J. Bai, X. Song, and I. King, "Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 299–311.

[37] R. Zhang, H. Ma, Q. Li, Z. Li, and Y. Wang, "A knowledge graph recommendation model via high-order feature interaction and intent decomposition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–7.

[38] X. Zhang, H. Ma, Z. Gao, Z. Li, and L. Chang, "Exploiting cross-session information for knowledge-aware session-based recommendation via graph attention networks," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 7614–7637, Oct. 2022.

[39] R. Zhang, H. Ma, Q. Li, Y. Wang, and Z. Li, "FIRE: Knowledge-enhanced recommendation with feature interaction and intent-aware attention networks," *Appl. Intell.*, pp. 1–21, Dec. 2022.

FAN YANG is currently pursuing the Ph.D. degree with Xi'an Jiaotong-Liverpool University. Her research interests include recommender systems, user modeling, knowledge graph, and machine learning.

YONG YUE received the B.Eng. degree Northeastern University, China, and the Ph.D. degree Heriot-Watt University, U.K.

He was the Head of the Department of Computer Science and Software Engineering (2013–2019). He is currently a Professor with the Department of Computing, Xi'an Jiaotong-Liverpool University (XJTLU), and the Director of the Virtual Engineering Centre (VEC) and the Suzhou Municipal Key Laboratory for Intelligent Virtual Engineering. Before joining XJTLU, he had held various positions in industry and academia in China and the U.K., including an Engineer, a Project Manager, a Professor, the Director of Research, and the Head of the Department. He has experience in learning and teaching, research and enterprise, and management. He has led various research and professional projects supported by primary funding bodies and industry. His research interests include operation research, computer vision, and virtual reality. He is a fellow of IET, IMechE, and HEA. He is a C.Eng.

GANGMIN LI is currently a Senior Lecturer in computer science with the University of Bedfordshire. He is an highly educated academic with over 35 years of research and teaching experience in four British and two Chinese universities. He has obtained over 200K in funding from the government and industry. He has published over 80 journals and conference papers. His research interests include data science, distributed AI, knowledge engineering (KE), agent and multi-agent systems, grid computing, and HCI.

He has served as many academic journal editors and international conference technical committee members.

TERRY R. PAYNE has been actively working in distributed AI and multi-agent systems research, since 1994. He is currently a Senior Lecturer with the University of Liverpool, where he heads the Robotics and Autonomous Systems Research Group. In addition to teaching final year research-led subjects (on robotics and multi-agent systems), he is the Academic Lead for Student Recruitment, Admissions and Widening Participation with the Faculty of Science and Engineering. His research interests include ontological knowledge to support agent/service discovery and provision. He has also worked on coalition formation, self-organizing agent communities, and machine learning for adaptive agents.

He is the winner of several research and teaching awards.

KA LOK MAN (Member, IEEE) received the Dr.Eng. degree in electronic engineering from Politecnico di Torino, Turin, Italy, in 1998, and the Ph.D. degree in computer science from Technische Universiteit Eindhoven, Eindhoven, The Netherland, in 2006. He is currently a Professor of computer science and software engineering with Xi'an Jiaotong-Liverpool University, Suzhou, China. His research interests include formal methods and process algebras, embedded system design and testing, and photovoltaics.