IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 25, 2018, accepted October 21, 2018, date of publication November 9, 2018,
date of current version December 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2880197

A Survey of Recommender Systems Based on Deep Learning

RUIHUI MU[1,2]
[1]College of Computer and Information, Hohai University, Nanjing 210098, China
[2]College of Computer and Information Engineering, Xinxiang University, Xinxiang 453003, China
(muruihui@126.com)

This work was supported in part by the National Key Research and Development Plan Key Projects of China under Grant 2017YFC0405800, in part by the National Natural Science Foundation of China under Grant 60971088, Grant 60571048, Grant 61432008, and Grant 61375121, and in part by the Natural Science Foundation of the Colleges and Universities in Jiangsu Province of China under Grant 17KJB520022.

ABSTRACT In recent years, deep learning's revolutionary advances in speech recognition, image analysis, and natural language processing have gained significant attention. Deep learning technology has become a hotspot research field in the artificial intelligence and has been applied into recommender system. In contrast to traditional recommendation models, deep learning is able to effectively capture the non-linear and non-trivial user-item relationships and enables the codification of more complex abstractions as data representations in the higher layers. In this paper, we provide a comprehensive review of the related research contents of deep learning-based recommender systems. First, we introduce the basic terminologies and the background concepts of recommender systems and deep learning technology. Second, we describe the main current research on deep learning-based recommender systems. Third, we provide the possible research directions of deep learning-based recommender systems in the future. Finally, concludes this paper.

INDEX TERMS Deep learning, recommender systems, deep learning-based recommender systems, machine learning, terminology.

I. INTRODUCTION

In recent years, with the rapid development of sensor technology, storage technology, computer technology, and network technology, the explosion of data has become increasingly intensified [1]. However, as the volume of data increases, individuals have to face the problem of excessive information, which makes it more difficult to make the right decisions. This phenomenon is known as information overload [2]. Using the technique of artificial intelligence to obtain abstract information from Big data and convert it into useful knowledge is one of the core issues in big data analysis [3]. In order to solve the problem of information overload, recommender system emerges as the times require. Its main idea is to analyze the user's historical behavior and preference information, establish a model, and automatically recommend the items or products of interest to the user, then get a personalized list for the user [4]. The recommender system can not only recommend items with similar preferences to the user according to the user's preference, but also can even recommend unknown items of interest to the user without user's preferences. The recommender systems can alleviate these problems by effectively finding users' potential requirements and selecting desirable items from a huge amount of candidate information.

The current research direction of recommender system can be divided into three categories [5]: content-based recommendation, collaborative filtering-based recommendation, and hybrid recommendation methods. Content-based recommendation methods make full use of the user's profile and description of the product to generate recommended items. Collaborative filtering-based recommendation methods [6] makes full use of the behavior information and preference information generated by the user in the past without using the user's personal information and product description information, such as the user's rating of the item to generate the recommended item. Hybrid recommendation methods [7] seek to obtain the best recommendation results by combining content-based recommendation methods and collaborative filtering-based recommendation methods. As personal privacy issues are getting more and more attention from Internet users, it has become more and more difficult to collect user profiles for content-based recommendation methods.

 This work is licensed under a Creative Commons Attribution 3.0 License. For more information, see http://creativecommons.org/licenses/by/3.0/

Collaborative filtering-based recommendation methods also have some limitations. When the rating matrix is very sparse, the recommendation accuracy usually drops very obviously. More importantly, it cannot generate recommendations for a new item that has not received user ratings. Therefore, collaborative filtering-based recommendation methods make use of some auxiliary information become irreversible, such as commodity description information [8].

At the same time, in recent years, the development of deep learning technology has made rapid progress and development, becoming a hot and emerging field in the data mining and machine learning community [9]. Deep learning was originally used in the fields of image recognition, speech recognition, and natural language processing, and achieved the most advanced results in these fields [10]. By constructing a multi-layer, nonlinear, layer-to-layer interconnection network structure, deep learning can use the network structure to approach a complex multivariate function as a training target, and to learn the original features of the dataset from many unlabeled training data set. Deep learning is an emerging field of machine learning research. It aims to research how to automatically extract multi-level feature representations from data. Its basic idea is to extract features from data by combining low-level features to form denser high-level semantic abstractions, thus solving the problem of manually design features in the traditional machine learning [10]. For the case where the image and voice data contain a large amount of unlabeled data, the deep learning model can learn more effective features from the large number of unlabeled training data for identification.

Deep learning has been revolutionizing the recommendation architectures dramatically and brings more opportunities to improve the performance of recommender. Recent advances in deep learning-based recommender systems(DLRS) have gained significant attention by overcoming obstacles of conventional models and achieving high recommendation quality. Deep learning is able to effectively capture the non-linear and non-trivial user-item relationships and enable the codification of more complex abstractions as data representations in the higher layers [12]–[24]. Furthermore, it catches the intricate relationships within the data itself, from abundant accessible data sources such as contextual, textual and visual information.

The remainder of the paper is organized as follows. Section 2 gives the basic terminologies and background concepts of recommender systems and deep learning technology. Section 3 describes the main current research on deep learning-based recommender systems. In section 4, we present the possible research directions in the future. Finally, section 5 concludes our work.

II. TERMINOLOGIES AND BACKGROUND CONCEPTS

In this section, we describe the basic terminologies and concepts regarding recommender system and deep learning technology.

A. RECOMMENDER SYSTEM

In this part, we describe the traditional recommender systems, such as content-based recommender systems, collaborative filtering(CF) recommender systems and hybrid recommender systems.

The recommender systems can alleviate these problems by effectively finding users' potential requirements and selecting desirable items from a huge amount of candidate information [25]. The core idea of the recommender system is the recommendation algorithm. Traditional recommender systems are mainly divided into three types [26]: content-based recommender systems [27], collaborative filtering recommender systems [28], and hybrid recommender systems [29].

1) CONTENT-BASED RECOMMENDER SYSTEMS

Content-based recommender systems utilize the contents of items and finds the similarities among them. After analyzing sufficient numbers of items that one user has already shown favor to, the user interests profile is established. Then the recommender system could search the database and choose proper items according to this profile [30].

The difficulty of these algorithms lies in how to find user preferences based on the contents of items. Many approaches have been developed to solve this problem in the areas of data mining or machine learning [31]. For example, in order to recommend some articles to a specific reader, a recommender system firstly obtains all the books this reader has already read and then analyzes their contents. Key words can be extracted from the text with the help of text mining methods, such as the well-known TF-IDF. After integrating all the key words with their respective weights, a book can be represented by a multi-dimensional vector. Specific clustering algorithms [32] can be implemented to find the centers of these vectors which represent the interests of this reader.

2) COLLABORATIVE FILTERING (CF) RECOMMENDER SYSTEMS

Collaborative filtering (CF) has become one of the most influential recommendation algorithms. Unlike the content-based approaches, CF only relies on the item ratings from each user. It is based on the assumption that users who have rated the same items with similar ratings are likely to have similar preferences. Collaborative filtering recommends item based on the interest of other like-minded users or identify items similar to those previously rated by the target user. It uses statistical techniques to find the similarity between the user or item vector. CF methods can be classified into two categories Memory-Based and Model-Based [33].

3) HYBRID RECOMMENDER SYSTEMS

Hybrid recommendation systems are divided into monolithic hybrid recommendation, parallel hybrid recommendation, and pipeline hybrid recommendation [34]. Monolithic hybrid recommendation is a hybrid recommendation method that integrates several recommendation strategies into one algorithm [35]. The remaining two hybrid recommendations require at least two different recommendation methods and then combine them. According to the input, the parallel hybrid recommendation operates independently of each other, respectively generating a recommendation list, and then the output data is combined into the final recommendation set. The pipeline hybrid recommendation connects multiple recommender systems in pipelined architecture, with the output of the previous recommender system becoming the input portion of the latter recommender system. Of course, the latter recommendation unit can also choose to use part of the original input data.

Hybrid recommender systems are used either to leverage the power of multiple data sources or to improve the performance of existing recommender systems within a particular data modality [36]. An important motivation for the construction of hybrid recommender systems is that different types of recommender systems, such as collaborative filtering-based, content-based methods, have different strengths and weaknesses. Some recommender systems work more effectively at cold start, whereas other work more effectively when sufficient data are available [37]. Hybrid recommender systems attempt to leverage the complementary strengths of these systems to create a system with greater overall robustness [38].

B. DEEP LEARNING TECHNOLOGY

Deep learning stems from the study of artificial neural networks [39]. A multilayer perceptron with multiple hidden layers is a deep learning structure. The concept of deep learning was proposed by Hinton *et al.* [40] in 2006. He proposed an unsupervised greedy layer-by-layer training algorithm based on deep belief network (DBN), which brought hope to solve the optimization problem related to deep structure. Then he proposed the deep structure of multi-layer autoencoder [41]. In addition, the convolutional neural network proposed by Lecun *et al.* [42]. It is the first true multi-layer structure learning algorithm, which uses spatial relative relations to reduce the number of parameters to improve training performance. Deep learning combines low-level features to form more abstract high-level representation attribute categories or features to discover distributed feature representations of data [43].

Deep learning is a new field in machine learning research [44]. It mimics the mechanism of the human brain to interpret data such as images, sounds and texts [45]. Like the machine learning method, the deep learning method also divided into supervised learning and unsupervised learning [46].

In this section, we describe commonly used deep learning models. First, we introduce the autoencoder (AE). Then, we present the details of restricted Boltzmann machine (RBM). Next, we present recurrent neural network (RNN) and convolutional neural network (CNN). Finally, we describe deep belief network (DBN).

1) AUTOENCODER

Autoencoder (AE) is an unsupervised model attempting to reconstruct its input data in the output layer. In general, the bottleneck layer (the middle-most layer) is used as a salient feature representation of the input data. Autoencoder can be seen as a variant of the traditional multi-layer perceptron, first proposed by Rumelhart *et al.* [47].

Autoencoder reconstructs the input data to learn the latent feature of the data through coding and decoding process. The structure is shown in Figure 1.

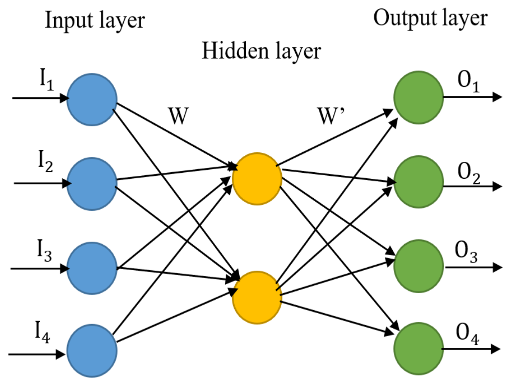

FIGURE 1. Autoencoder.

Autoencoder consists of a three-layer network in which the number of neurons in the input layer is equal to the number of neurons in the output layer, and the number of neurons in the middle layer is less than that of the input layer and the output layer. During network training, for each training sample, a new signal is generated at the output layer through the network. The purpose of network learning is to make the output signal and input signal as similar as possible. This similarity is represented by the reconstruction error. Autoencoder can form a deep structure by cascading and layer-by-layer training. After trained the deep model by layer-by-layer optimization, fine tuning can also be performed by allowing the entire network to reconstruct the input signal. There are many variants of autoencoders such as denoising autoencoder, marginalized denoising autoencoder [48], sparse autoencoder, stacked denoising autoencoder (SDAE) [49], contractive autoencoder and variational autoencoder (VAE).

There are two general ways of applying autoencoder to recommender system: (1) using autoencoder to learn lower-dimensional feature representations at the bottleneck layer; or (2) filling the blanks of the interaction matrix directly in the reconstruction layer. Almost all the autoencoder variants such as denoising autoencoder, variational autoencoder, and marginalized autoencoder can be applied to recommendation task [50]–[55].

2) RESTRICTED BOLTZMANN MACHINE

Restricted Boltzmann Machine (RBM) is an extension of Boltzmann machine [56]. Restricted Boltzmann

Machine (RBM) is a two-layer neural network consisting of a visible layer and a hidden layer [57], [58]. It is one of the earliest artificial neural networks capable of solving complex learning problems by learning the inherent intrinsic expression of data. The learning efficiency is greatly improved because the connection between the same layers is removed. Restricted here means that there are no intra-layer communications in visible layer or hidden layer. It can be easily stacked to a deep net. The structure of restricted Boltzmann machine is shown in Figure 2 [56].

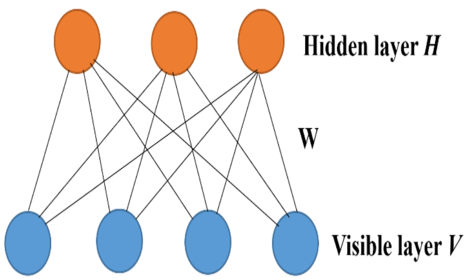

FIGURE 2. Restricted Boltzmann machine.

RBM has only two layers of neurons, one called the visible layer V, which consists of visible units for inputting training data. The other layer is called the hidden layer H and is composed of hidden units and used as feature detectors. The nodes between the two layers are fully connected, and the nodes in the same layer are not connected to each other. The joint probability distribution between the visible layer and the hidden layer is defined as:

$$P(v,h) = \frac{1}{Z} e^{-(a^T v + b^T h + h^T wv)} \tag{1}$$

Where a is the biases of visible layer, b is the biases of hidden layer, Z is the normalized function and W represents the connection weight between the visible layer and the hidden layer. Optimization goal of this model is based on the maximum likelihood estimation:

$$\arg max \sum_{v \in V} \log P(v,h) \tag{2}$$

Compared with other models such as sparse Autoencoder, RBM model is very fast, and only requires a simple forward encoding operation. But if we use the traditional sampling method to solve the problem, the number of iterations is too much, and the efficient is low. In order to overcome this problem, Smolensky [57] Hinton [58] proposed a fast algorithm called contrastive divergence (CD).

By cascading multiple single-layer restricted Boltzmann machine models, a deep structure can be formed. That is, the hidden layer of the previous layer is used as the visible layer of the current layer. The optimization of the network adopts a layer-by-layer optimization method. Some extended models modify the RBM's structure and probability distribution model so that it can simulate more complex probability distributions.

Mnih *et al.* [59], Phung and Venkatesh [60], Georgiev and Nakov [61], Jie-Yue and Be [62], and Nguyen and Lauw [63] proposed RBM-based recommender system. To the best of our knowledge, it is the first recommendation model that built on neural networks. The visible unit of RBM is limited to binary values, therefore, the rating score is represented in a one-hot vector to adapt to this restriction.

3) RECURRENT NEURAL NETWORK

Recurrent Neural Network (RNN) is suitable for modeling sequential data. Unlike feedforward neural network, there are loops and memories in RNN to remember former computations. RNNs have been widely used in machine translation [64], speech recognition [65], and label generation [66].

An ordinary full-connected network or a convolutional neural network is a structure which from the input layer to the hidden layer to the output layer, the layers are fully connected between layers, and the nodes are disconnected in each layer. This structure of neural network is often incompetent to model the sequential data. In a fully connected DNN and CNN network, the signals of each layer can only propagate to the next layer, and the processing of the samples is independent at each time. When the RNN accepts a new input, it combines the implied state vector with the new input to produce an output that depends on the entire sequence. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations.

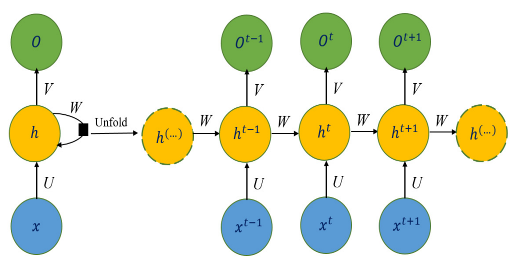

FIGURE 3. A recurrent neural network and the unfolding in time of the computation involved in its forward computation.

RNNs can model any length of sequential data in theory, but in practice it is often assumed that the current state is only related to the historical state of the previous time, which helps to reduce the complexity of the model. RNNs structure including input units, output units, and hidden units. As shown in Figure 3. The most important feature of RNNs is that the nodes in the hidden layer are connected. It calculates the output of the hidden layer at the current time by obtaining the output of the input layer and the hidden layer state at the previous time, that is, RNNs can remember the past information.

RNN can be applied directly to itself at the next timestamp, i.e. the input of the i-th neuron at time t, except for the output of the (i-1) layer neuron at time t-1, including its own input at time t.

Where x is the input, h is the hidden layer unit, o is the output, L is the loss function, and y is the label of the training set. The t in the upper right corner of these elements represents the state at time t. It should be noted that the performance of the decision unit h is determined not only by the input of this moment but also by the time before time t. V, W, and U are weights.

There are some RNN models for different application requirements, such as long short-term memory (LSTM) [67], gated recurrent unit (GRU) [68], memory network [69], stack-augmented recurrent Net [70], neural turing machines (NTM) [71], differentiable neural computer (DNC) [72] and so on. LSTM is a long short-term memory network. It is a time recurrent neural network suitable for processing and predicting important events with relatively long intervals and delays in time series. LSTM combines short-term memory with long-term memory through subtle gate control, solves the problem of gradient disappearance to some extent. The difference between LSTM and RNN is that it adds a "processor" to the algorithm to judge whether the information is useful or not. The structure of this processor is called cell. Three cells are placed in a cell, called input gate, forget gate and output gate. Only information that complies with the algorithm's certification will be retained, and information that does not match will be forgotten through the forgot gate.

RNNs are extremely suitable for sequential data processing [73]–[80]. As such, it becomes a natural choice for dealing with the temporal dynamics of interactions and sequential patterns of user behaviors, as well as side information with sequential signals, such as texts, audio, etc. Chen *et al.* [81] proposed point-of-interest recommendation in location-based social networks.

4) CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network (CNN) [42], [82], [83] is a special kind of feedforward neural network with convolution layers and pooling operations. It can capture the global and local features and significantly enhancing the efficiency and accuracy. It performs well in processing data with grid-like topology. CNN is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. Convolutional neural network is a multi-layer perceptron. In recent years, convolutional neural networks(CNN) have become a hot topic in the field of image understanding.

Compared to traditional multi-layer perceptron, CNN uses pooling operation to reduce the number of neurons in the model and is more robust shift invariant or space invariant. In addition, CNN's shared-weights architecture can reduce the number of parameters in the model, reduce the complexity of the model and enhance the generalization ability of the model. CNN is mainly used to process two-dimensional image data. The basic structure of a convolutional neural network is composed of input layer, convolutional layer,

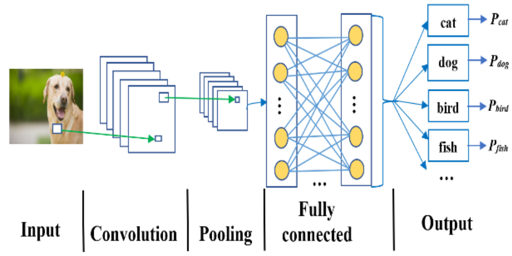

FIGURE 4. Convolutional neural network.

subsampling layer (pooling layer), fully connected layer, and output layer, as shown in Figure 4.

Convolution Neural Networks are powerful in processing unstructured multimedia data with convolution and pool operations. Most of the CNNs based recommendation models utilize CNNs for feature extraction [84]–[89].

5) DEEP BELIEF NETWORK

Deep Belief Network (DBN) was a generation model proposed by Geoffrey Hinton *et al.* [40]. The components of the DBN are Restricted Boltzmann Machines (RBM). The process of training DBN is done layer by layer [90]. In each layer, the data vector is used to infer the hidden layer, and this hidden layer is treated as the data vector of the next layer (higher layer). Its structure is shown in Figure 5.

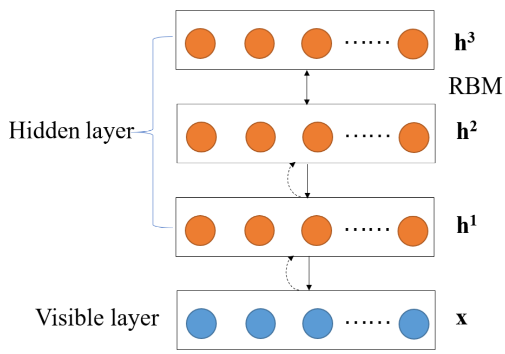

FIGURE 5. Deep belief network.

The structure of DBN can be thought of as consisting of multiple restricted Boltzmann machine stacks. The hidden layer of the previous RBM is considered as the visible layer of the next RBM in the network. In this way, during the training of DBN, each RBM can use the output of the previous RBM alone to train, so compared with the traditional neural network, the training of DBN is simpler.

Deep neural networks can be formed by increasing the number of hidden layers and the corresponding number of nodes. Deep neural networks generally refer to fully connected neural networks.

Such neural network models are often applied to extract the feature representation of music and thus perform music recommendation [91], image and speech recognition, etc.

The large number of parameters limits the depth and breadth of the full connection neural network model structure.

In this section, we describe commonly used deep learning models, Table 1 lists the summary of commonly used deep learning models.

TABLE 1. The summary of commonly used deep learning models.

Types of models	Deep learning models	Features
Unsupervised learning	AE	Encoder + decoder Dimensionality reduction Noise reduction
	RBM, DBN	Depth generation model (pre-training)
Supervised learning	CNN	Convolution transformation Pooling operation Processing grid data
	RNN	LSTM Bidirectional RNN Processing sequential data

III. DEEP LEARNING-BASED RECOMMENDER SYSTEM

In this section, we describe deep learning-based recommender systems. First, we introduce deep learning in content-based recommender systems. Then, the details of deep learning-based collaborative filtering recommender systems are provided. Next, we present deep learning-based hybrid recommender systems and deep learning in social network-based recommender systems. Finally, we describe deep learning in context-aware recommender systems.

Deep learning-based recommender systems can be divided into two broad categories: integration model and neural network model. Considering whether it integrates traditional recommendation models with deep learning or relies solely on deep learning technique, we classified Integration model into two models: integrate deep learning with traditional recommendation model and recommend rely solely on deep learning model. Integrate deep learning with traditional recommendation model attempt to combine deep learning methods with traditional recommendation techniques in one way or another. Recommend rely solely on deep learning model rely on deep learning techniques without any forms of help from traditional recommendation models.

In accordance with the types of employed deep learning techniques. We divide neural network model into two categories: models using single deep learning technique and deep composite model. The motivation of deep composite model is that different deep learning techniques can complement one another and enable a more powerful hybrid model. The basic architecture of deep learning-based recommender system as shown in Figure 6.

A. DEEP LEARNING IN CONTENT-BASED RECOMMENDER SYSTEMS

Content-based recommender system is mainly based on items' and users' auxiliary information. A various range of auxiliary information such as texts, images and videos can be taken into account. Deep learning in content-based recommender systems are mainly used to effectively capture the non-linear and non-trivial user-item relationships and enable the codification of more complex abstractions as data representations in the higher layers [80]. Furthermore, it catches the intricate relationships within the data itself, from abundant accessible data sources such as contextual, textual and visual information.

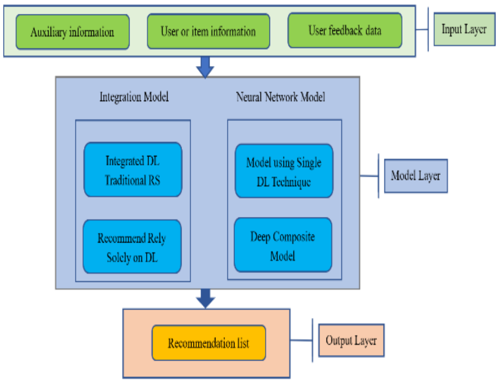

FIGURE 6. The architecture of deep learning-based recommender systems.

B. DEEP LEARNING IN COLLABORATIVE FILTERING RECOMMENDER SYSTEMS

Collaborative filtering(CF) is a widely used approach in recommender systems to solve many real-world problems. Traditional CF-based methods employ the user-item matrix which encodes the individual preferences of users for items for learning to make recommendation. In real applications, the rating matrix is usually very sparse, causing CF-based methods to degrade significantly in recommendation performance. Some improved CF methods utilize the increasing amount of side information to address the data sparsity problem as well as the cold start problem. However, the learned latent factors may not be effective due to the sparse nature of the user-item matrix and the side information. Some researchers utilize advances of learning effective representations in deep learning, propose deep learning-based collaborative filtering methods, which is a kind of model-based collaborative filtering recommendation methods.

1) AUTOENCODER-BASED COLLABORATIVE FILTERING METHOD

Autoencoder-based Collaborative Filtering (ACF) [50] is the first autoencoder-based collaborative recommendation model. Instead of using the original partial observed vectors, it decomposes them by integer ratings. Sedhain *et al.* [50] proposed autoencoder-based collaborative filtering recommendation method(AutoRec), the input of model is user-based ratings or item-based ratings in the rating matrix R, produces an output through encoding and decoding process and optimizes the model parameters by minimizing the reconstruction error. For example, if the rating score is integer in the range of [1-5], each r_{ui} will be divided into five partial vectors.

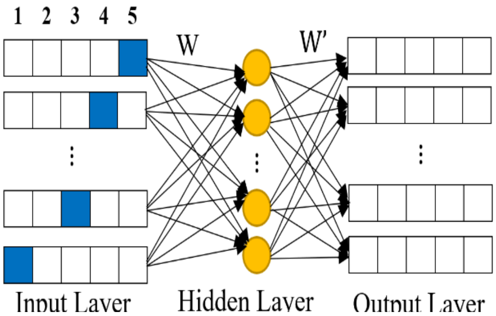

FIGURE 7. AE-based collaborative filtering model.

Figure 7 presents an example of AE-based collaborative filtering model, where rating scale from 1 to 5. The blue entries indicate that the user has rated that item as the corresponding rating (e.g. the user gives 5 for item 1, 1 for item 5). The cost function of autoencoder-based collaborative filtering recommendation method aims at reducing the mean squared error. The rating prediction of autoencoder-based collaborative filtering recommendation method is calculated by summarizing each entry of the five vectors, then scaled by the maximum rating K. It uses RBM to pretrain the parameters as well as to avoid local optimum. Stacking several autoencoders together also enhances the accuracy slightly. However, there are two demerits of autoencoder-based collaborative filtering recommendation method: it fails to deal with non-integer ratings; the decomposition of partial observed vectors increases the sparseness of input data and leads to worse prediction accuracy.

Collaborative Denoising Auto-Encoder (CDAE) [92] is principally used for ranking prediction. The input of CDAE is user partial observed implicit feedback. The entry value is 1 if the user likes the movie, otherwise 0. It can also be regarded as a preference vector which reflects user's interests to items. The input of CDAE is corrupted by Gaussian noise.

CDAE initially updates its parameters using SGD over all feedback. However, the authors argued that it is impractical to take all ratings into consideration in real world applications, so they proposed a negative sampling technique to sample a small subset from the negative set (items with which the user has not interacted), which reduces the time complexity substantially without degrading the ranking quality.

2) RESTRICTED BOLTZMANN MACHINE-BASED COLLABORATIVE FILTERING METHOD

Restricted Boltzmann Machine (RBM) is a two-layer neural network consisting of a visible layer and a hidden layer. It is one of the earliest artificial neural networks capable of solving complex learning problems by learning the inherent intrinsic expression of data. The learning efficiency is greatly improved because the connection between the same layers is removed. Mnih *et al.* [59] proposed a restricted Boltzmann machine-based recommender system. The researchers further proposed using a conditional RBM to incorporate the implicit feedback information. The visible unit of RBM is limited to binary values, therefore, the rating score is represented in a one-hot vector to adapt to this restriction. The structure of the model is shown in Figure 8.

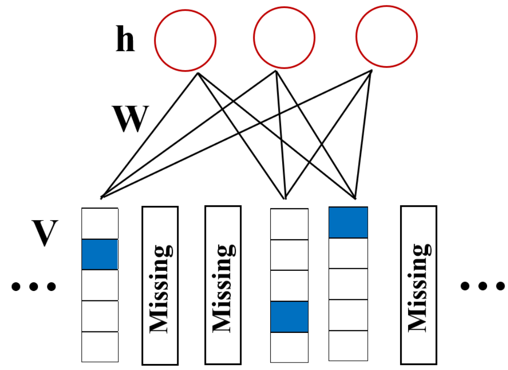

FIGURE 8. RBM-based collaborative filtering model.

Every RBM has the same number of hidden units, but an RBM only has visible softmax units for the movies rated by that user, so an RBM has few connections if that user rated few movies. Each RBM only has a single training case, but all of the corresponding weights and biases are tied together, so if two users have rated the same movie, their two RBM's must use the same weights between the softmax visible unit for that movie and the hidden units. The binary states of the hidden units can be quite different for different users. Suppose user rated N movies, the number of visible units is n, Let M be a $X \times N$ matrix where $r_{ui}^x = 1$ if user u rated movie i as x and $r_{ui}^x = 0$ otherwise. Then:

$$p\left(r_{ui}^x = 1 \mid \mathbf{h}\right) = \frac{\exp(b_i^x + \sum_{j=1}^{T} h_j w_{ij}^x)}{\sum_{l=1}^{X} \exp(b_i^l + \sum_{j=1}^{T} h_j w_{ij}^l)} \quad (3)$$

$$p\left(h_j = 1 | \mathbf{M}\right) = \sigma\left(b_j + \sum\nolimits_{i=1}^{N} \sum\nolimits_{k=1}^{T} r_{ui}^x w_{ij}^x\right) \quad (4)$$

where $\sigma(x) = 1/\left(1 + e^{-x}\right)$ is the logistic function, w_{ij}^x is a symmetric interaction parameter between hidden unit j and rating x of movie i, b_i^x is the bias of rating x for movie i and b_j is the bias of hidden unit j.

3) RECURRENT NEURAL NETWORK-BASED COLLABORATIVE FILTERING METHOD

Recurrent neural network (RNN) is suitable for modeling sequential data. Unlike feedforward neural network, there are loops and memories in RNN to remember former computations. Variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) network are often deployed in practice to overcome the vanishing gradient problem. The basic idea of RNN-based collaborative filtering is to use RNN to model the effect of user historical sequence behavior on the user current behavior, then recommends items for the user and predicts user's behavior [78]. Figure 9 is a basic RNN-based collaborative filtering method framework [78]. Suppose the input is $\{I_1, I_2, \ldots, I_t\}$, $O_t = \sigma\left(g\left(W \cdot h_{t-1} + V \cdot I_t\right) \cdot V\right)$, g is the activation function, O_t is the output, it denotes the probability of selecting a specific item at time t, it is usually calculated by the hidden vector at time t. σ is softmax function. Recent advancements have demonstrated the efficacy of RNN in solving this session-based recommendation.

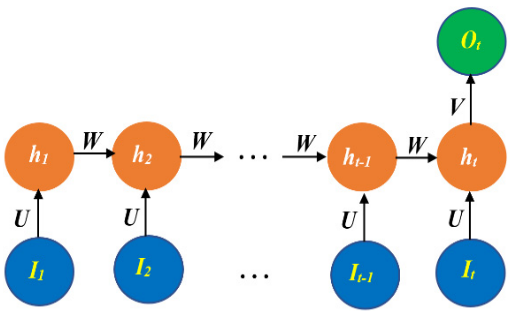

FIGURE 9. RNN-based collaborative filtering model.

4) GENERATIVE ADVERSARIAL NETWORK-BASED COLLABORATIVE FILTERING METHOD

Generative Adversarial Network (GAN) is a generative neural network which consists of a discriminator and a generator. The two neural networks are trained simultaneously by competing with each other in a minimax game framework. Information retrieval generative adversarial network(IRGAN) [93] is the first model which applies GAN to Information Retrieval area. Traditional GAN consists of a discriminator and a generator. Likely, there are two schools of thinking in information retrieval: the generative retrieval focusing on predicting relevant documents given a query, while discriminative retrieval focusing on predicting relevancy given a query-document pair. On one hand, the discriminative model, aiming to mine signals from labeled and un-labeled data, provides guidance to train the generative model towards fitting the underlying relevance distribution over documents given the query. On the other hand, the generative model, acting as an attacker to the current discriminative model, generates difficult examples for the discriminative model in an adversarial way by minimizing its discrimination objective.

Thus, inspired by the idea of GAN, IRGAN unify these two different types of IR models by letting them play a minimax game: the generative retrieval model would try to generate (or select) relevant documents that look like the ground truth relevant documents and therefore could fool the discriminative retrieval model, whereas the discriminative retrieval model would try to draw a clear distinction between the ground-truth relevant documents and the generated ones made by its opponent generative retrieval model.

$$J^{G^*D^*} = \min_{\theta} \max_{\emptyset} \sum_{n=1}^{N} \left(E_{d \sim P_{true}(d \mid q_n, r)} \left[\log D\left(d \mid q_n\right)\right] + E_{d \sim P_{\theta}(d \mid q_n, r)} \left[\log\left(1 - D\left(d \mid q_n\right)\right)\right]\right) \quad (5)$$

where the generator G is directly written as $P_\theta\left(d \mid q_n, r\right)$, and the probability of document d being relevant to query q is given by the sigmoid function of the discriminator score

$$D\left(d \mid q\right) = \frac{\exp(f_\theta(d, q))}{1 + \exp(f_\theta(d, q))} \quad (6)$$

From formula (4), we can see that the optimal parameters of the generative retrieval model and the discriminative retrieval model can be learned iteratively by maximizing and minimizing the same objective function, respectively. In collaborative filtering, a widely adopted model is matrix factorization, following which they define scoring function for the preference of user u (i.e. the query) to item i (i.e. the document) as

$$s\left(u, i\right) = b_i + W^T V \quad (7)$$

where b_i is the bias term for item i, W and V are the latent vectors of user u and item i respectively, defined in the k-dimensional continuous space. Here we omit the global bias and user bias as they are reduced in the task of top-N item recommendation for each user.

Despite the great empirical success of GAN, there are still many questions with regard to its theoretical foundation remaining to be answered by the research community. For example, it is "not entirely clear" why GAN can generate sharper realistic images than alternative techniques.

C. DEEP LEARNING-BASED HYBRID RECOMMENDER SYSTEMS

Traditional CF-based methods employ the user-item matrix which encodes the individual preferences of users for items for learning to make recommendation. In real applications, the rating matrix is usually very sparse, causing CF-based methods to degrade significantly in recommendation performance. In this case, some improved CF methods utilize the increasing amount of side information to address the data sparsity problem as well as the cold start problem. However, the learned latent factors may not be effective due to the sparse nature of the user-item matrix and the side information. The basic idea of deep learning-based hybrid recommendation method is to combine the content-based recommendation methods with collaborative filtering recommendation methods, integrate the feature learning of the user or item and the process of recommendation into a unified framework.

Wang and Blei utilize [94] advances of learning effective representations in deep learning, propose a stacked denoising autoencoder-based hybrid model which jointly performs deep users and items' latent factors learning from side information and collaborative filtering from the rating matrix.

An autoencoder is a specific form of neural network, which consists of an encoder and a decoder component. The encoder takes a given input and maps it to a hidden representation, while the decoder maps this hidden representation back to a reconstructed version of input. The parameters of the autoencoder are learned to minimize the reconstruction error,

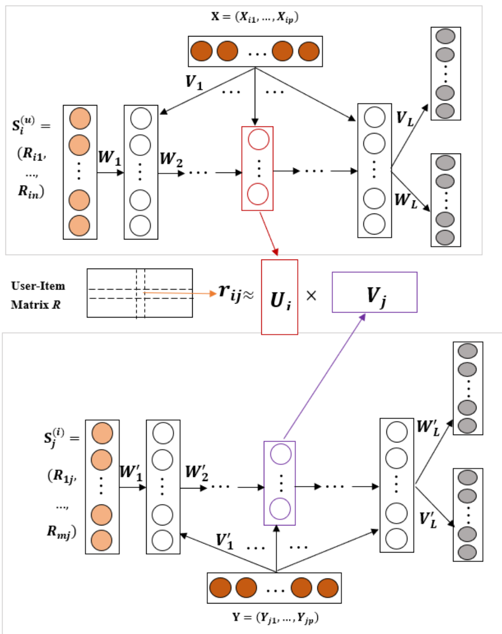

FIGURE 10. **The architecture of stacked denoising autoencoder -based hybrid recommendation method.**

measured by some loss function. However, denoising autoencoders (DAE) incorporate a slight modification to this setup, which reconstructs the input from a corrupted version with the motivation of learning a more effective representation from the input. A denoising autoencoder is trained to reconstruct the original input from its corrupted version by minimizing loss function. Usually, choices of corruption include additive isotropic Gaussian noise or binary masking noise. Moreover, various types of autoencoders have been developed in several domains to show promising results.

Stacked denoising autoencoder-based hybrid recommendation model makes use of both rating matrix and side information and combines SDAE [53] and matrix factorization together. Matrix factorization is a widely used model-based CF method with excellent scalability and accuracy, and SDAE is a powerful way to extract high-level representations from raw inputs. The combination of these two models leverages their benefits for learning more expressive models. The model contains three components: the upper component and the lower component are two SDAEs which extract latent factor vectors for users and items respectively; the middle component decomposes the rating matrix **R** into two latent factor matrices. The basic architecture of stacked denoising autoencoder-based hybrid recommendation method is shown in Figure 10.

Given the user-item rating matrix **R**, we first transform **R** into the set $S^{(u)}$ containing m instances $\{\mathbf{S}_1^{(\mathbf{u})}, \dots, \mathbf{S}_m^{(u)}\}$, where $\mathbf{S}_i^{(u)} = \{R_{i1}, \dots, R_{in}\}$ is the n-dimensional feedback vector of user i on all the items. Similarly, we can obtain set $S^{(i)}$ with n instances $\{\mathbf{S}_1^{(i)}, \dots, \mathbf{S}_n^{(i)}\}$, where $\mathbf{S}_j^{(i)} = \{R_{1j}, \dots, R_{mj}\}$ is the m-dimensional feedback vector of item j rated by all the users. hybrid model learns user and item latent factors (i.e., **U** and **V**) from **R**, $S^{(u)}$, $S^{(i)}$ and the additional side information (i.e., **X** and **Y**) through the following optimization objective:

$$\arg\min_{\mathbf{U},\mathbf{V}} G(R, U^T V) + \alpha\left(\|U\|_F^2 + \|V\|_F^2\right) + \beta\sigma(S^{(u)}, X, U) + \gamma\sigma(S^{(i)}, Y, V) \quad (8)$$

where $\sigma(\cdot, \cdot, \cdot)$ is the function that connects the user or item side information with the latent factors,$G(\cdot, \cdot)$ is the loss function for decomposing the rating matrix **R** into two latent factor matrices **U** and **V**, α is a regularization parameter, β and γ are the trade-off parameters. Note that, the last two terms in Equation (8) devised using SDAE model which extracts latent factor matrix from the rating matrix and additional side information.

D. COMBINATION OF DEEP LEARNING AND SOCIAL NETWORK-BASED RECOMMENDER SYSTEMS

Traditional recommender systems always ignore social relationships among users. But in our real life, when we are asking our friends for recommendations of nice digital cameras or touching movies, we are actually requesting verbal social recommendations. Social recommendation is a daily occurrence, and we always turn to our friends for recommendations [95]. Hence, in order to improve recommender systems and to provide more personalized recommendation results, we need to incorporate social network information among users. There are various types of social relationships among users in social networks, and users will interact with each other through social relationships.

The most important recommender system for social network-based recommendation is to improve the quality of recommender system by modeling the influence of social relationships between users. In social network, all items have location attributes, and user behavior has sequential patterns in time and space. Modeling this spatiotemporal sequence pattern is beneficial to improve the accuracy of point of interest recommendations. Therefore, recommender system based on the location social network needs to pay attention to the social relationship between users, and also needs to grasp the user's location influence, sequence movement mode and other factors. Recently, based on the intuition that users' trust relations can be employed to enhance traditional recommender systems, a few trust-aware recommendation methods have been proposed. These methods utilize the inferred implicit or observed explicit trust information to further improve traditional recommender systems. Trust-aware recommender systems move an important step forward in the research of recommender systems.

At present, the combination of deep learning and social network-based recommender systems has also triggered a series of research results, however, to achieve the goal of "social recommendation", these approaches still have several inherent limitations and weaknesses that need to be addressed.

E. CONTEXT-AWARE RECOMMENDER SYSTEMS BASED ON DEEP LEARNING

Context-aware recommender systems (CARS), which incorporates contextual information into recommender systems, has become one of the hottest topics in the domain of recommender systems.

Deep learning is applied to context-aware recommender systems. Deep learning method can effectively integrate the context information into recommender systems in many complex recommendation scenarios [96]; And through deep learning method to get the latent representation of the context information. Based on this, the context-aware recommendation can effectively integrate into all kinds of rough situation data to alleviate data sparsity in context-aware recommender systems [97]. At present, the application of context-aware recommender systems based on deep learning mainly focuses on how to use deep learning methods to effectively model the situation information.

Sequential information plays an important role in modeling user behaviors, various sequential recommendation methods have been proposed. Methods based on Markov assumption are widely-used, but independently combine several most recent components. Recently, Recurrent Neural Networks (RNN) based methods have been successfully applied in several sequential modeling tasks. However, for real world applications, these methods have difficulty in modeling the contextual information, which has been proved to be very important for behavior modeling.

At present, the application of context-aware recommender systems based on deep learning mainly focuses on how to use deep learning methods to effectively model the situation information. Some researchers propose a novel model, named context-aware recurrent neural networks (CA-RNN). Instead of using the constant input matrix and transition matrix in conventional RNN models, CA-RNN employs adaptive context-specific input matrices and adaptive context-specific transition matrices. The adaptive context specific input matrices capture external situations where user behaviors happen, such as time, location, weather and so on.

F. COMPARISONS

The past few decades have witnessed the tremendous successes of the deep learning in many application domains. The academia and industry have been in a race to apply deep learning to a wider range of applications due to its capability in solving many complex tasks while providing start-of-the-art results. Recently, deep learning has been revolutionizing the recommendation architectures dramatically and brings more opportunities in reinventing the user experiences for better customer satisfaction. Recent advances in deep learning-based recommender systems have gained significant attention by overcoming obstacles of conventional models and achieving high recommendation quality [98].

Deep learning-based recommender systems use deep learning methods to learn the latent representations of users and items to make recommendations, and effectively capture the non-linear and non-trivial user-item relationships, enable the codification of more complex abstractions as data representations in the higher layers [99]. Furthermore, it catches the intricate relationships within the data itself, from abundant accessible data sources such as contextual, textual and visual information. However different types of methods have differences in deep learning models, data types and recommendation objects.

The differences of deep learning-based recommender systems are as follows:

(a) Deep learning in content-based recommender systems. Deep learning in content-based recommender systems are mainly used to effectively capture the non-linear and non-trivial user-item relationships and enable the codification of more complex abstractions as data representations in the higher layers. Furthermore, it catches the intricate relationships within the data itself, from abundant accessible data sources such as contextual, textual and visual information.

(b) Deep learning in collaborative filtering recommender systems. Deep learning-based collaborative filtering methods utilize the user's explicit feedback or implicit feedback information and adopt deep learning to train a recommendation model. It is a kind of model-based collaborative filtering recommendation methods. The basic idea is to take user-item rating matrix as input, use the deep learning model to learn the latent representation of users or items, and use the loss function (point-wise loss and pair-wise loss) to construct optimization function to optimize the parameters of deep learning model. Finally, based on the latent representation to make recommendation.

(c) Deep learning in hybrid recommender systems. The basic idea of deep learning-based hybrid recommendation method is to combine the content-based recommendation methods with collaborative filtering recommendation methods, integrate the feature learning of the user or item and the process of recommendation into a unified framework.

(d) Deep learning in social network-based recommender systems. The most important recommender system for social network-based recommendation is to improve the quality of recommender system by modeling the influence of social relationships between users. In social network, all items have location attributes, and user behavior has sequential patterns in time and space. Modeling this spatiotemporal sequence pattern is beneficial to improve the accuracy of point of interest recommendations. Therefore, recommender system based on the location social network needs to pay attention to the social relationship between users, and also needs to grasp the user's location influence, sequence movement mode and other factors. The combination of deep learning and social network-based recommender systems has also triggered a series of research results, however, to achieve the goal of ''social recommendation'', these approaches still have several inherent limitations and weaknesses that need to be addressed.

(e) Deep learning in context-aware recommender systems. Deep learning method can effectively integrate the context information into recommender systems in many complex recommendation scenarios; And through deep learning method to get the latent representation of the context information. Based on this, the context-aware recommendation can effectively integrate into all kinds of rough situation data to alleviate data sparsity in context-aware recommender systems. At present, the application of context-aware recommender systems based on deep learning mainly focuses on how to use deep learning methods to effectively model the situation information.

IV. POSSIBLE RESEARCH DIRECTIONS

Deep learning-based recommendation methods can effectively use multi-source heterogeneous data to alleviate the data sparsity and cold start problems. In the past three years, the research on deep learning-based recommender systems has attracted more and more attention from the academia and industry [100]. This section discusses the possible research direction of deep learning-based recommender system.

A. CROSS-DOMAIN RECOMMENDER SYSTEM BASED ON DEEP LEARNING

Several existing works [101]–[103] indicate the efficacy of deep learning in catching the generalizations and differences across different domains and generating better recommendations on cross-domain platforms. Currently, using deep learning techniques, various types of data as a unified input by embedding representations, constructing deep-level prediction models for recommendation by integrating various types of cross-platform heterogeneous data.

Cross domain recommendation can assist target domain recommendation with the knowledge learned from source domains, provides a desirable solution for these problems. Single domain recommendation only focuses on one domain while ignores the user interests on other domains, which also exacerbates data sparsity and cold start problems. Deep learning is well suited to transfer learning as it learns high-level abstractions that disentangle the variation of different domains. One of the most widely studied topics in cross domain recommendation is transfer learning which aims to improve learning tasks in one domain by using knowledge transferred from other domains. Therefore, it is a possible research direction of deep learning-based recommender system.

B. SCALABILITY OF DEEP LEARNING BASED RECOMMENDER SYSTEM

Scalability is critical to the usefulness of recommendation models in real-world systems as increasing data volumes in the big data. Deep learning has demonstrated to be very effective and promising in big data analysis. We can balance of the model complexity and scalability with the exponential growth of parameters.

At present, the scale of the network is beginning to move toward deeper but fewer parameters. Another possible research direction is high-dimensional input data can be compressed to compact embedding to reduce the space and computation time during model learning. The future research is to compress the redundant parameters so that the network may have better accuracy but have fewer parameters.

C. EXPLAINABILITY OF DEEP LEARNING BASED RECOMMENDER SYSTEM

Deep learning-based recommender systems use the deep learning model to directly predict the user preferences by using multi-source heterogeneous data [104]. The result of the model training is to get the weights between the neurons of deep neural network. It is difficult to give a reasonable explanation directly to the recommendation results. So, making explainable recommendations seem to be impossible. One way is to make explainable predictions to users, allowing them to understand the factors behind the network's recommendations, display an appropriate recommendation reason to tell the user why the system considers such a recommendation to be reasonable. The other way is to focus on explainability to the practitioner, probing weights and activations to understand more about the model. Therefore, it is it is also another possible research direction of deep learning-based recommender system, and next step would to be to design better deep learning-based recommender models to provide conversational or generative explanations.

D. ATTENTION MECHANISM BASED RECOMMENDER SYSTEM

Attentional mechanism has more or less eased the non-interpretable concerns of deep learning models. Attention mechanism has led to greater extents of interpretability since the attention weights not only give insights about the inner workings of the model but are also able to provide explainable results to users. Attentional mechanism is not only capable of enhancing performance but enjoys greater explain-ability. Applying the attention mechanism to recommender systems can help recommender systems to grasp the most informative features of the item, recommend the most representative item, and enhance the interpretability of the model [105]–[111].

Currently, attention mechanism has been applied to deep learning models. For example, Attention mechanism provides a better solution and helps RNN to better memorize inputs. Attention-based CNNs can identify the most informative part of the problem from the input. Given that models are already capable of capturing the most informative elements of the inputs, we believe that is another possible research direction.

E. DEEP COMPOSITE MODELS BASED RECOMMENDER SYSTEM

In current recommender systems, many elements need to be modeled, including the impact of socialized relationships, user-item interactions, the dynamic evolution of user preferences, etc. More elements be modeled can improve the performance of recommender systems. Deep composite model combining multiple deep neural networks enables a more powerful tool for modeling the heterogeneous characteristics of the determining factors in recommender system. There already have been a few studies which integrate different deep learning techniques together for enhancing performances. Nevertheless, the attempts are limited compared to the possible extensions, because the model should be designed in a sensible way rather than arbitrarily and tailored for practical requirements. Therefore, the research of deep composite models-based recommender system is also one of the future directions.

In short, deep learning has become popular in recommender systems community both in academia and industry. Meanwhile, this area of research is very young, there is much room for improvement in the aforementioned research directions, but we also believe that deep learning will revolutionize recommender systems dramatically and bring more opportunities in reinventing the user experiences for better customer satisfaction in the near future.

V. CONCLUSION

The dramatic increase in the amount of data being produced by electronic and automated devices necessitates the need for intelligent techniques and applications that can properly and intelligently store, process, access and analyze information for maximum benefits to users. Deep learning-based recommender systems(DLRS) are of such leading solutions to these challenges, which are appropriate tools to quickly aid the process of information seeking.

Deep learning-based recommender systems can learn the latent representations of users and items from massive data, and then construct a recommendation model, finally generate an effective recommendation list for the user.

The main tasks of the deep learning-based recommender systems are how to organize the massive multi-source heterogeneous data, build more suitable user models according to user preferences requirements, and improve the performance and user satisfaction. Compared with traditional recommender systems, deep learning-based recommender systems can use deep learning technique to automatically learn the latent features of user and item by integrating various types of multi-source heterogeneous data, model the sequence patterns of user behavior, more effectively reflect the user's different preferences and improve the accuracy of recommendation.

It is hoped that this review will assist novice and new researchers to understand the development of DLRS. Moreover, expert researchers can use this review as a benchmark to develop DLRS and as a reference to the limitations of DLRS.

REFERENCES

[1] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Greenwich, CT, USA: Manning Publications, 2015.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," in *Proc. IDC iView, IDC Analyze Future*, 2012, pp. 1–16.

[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 Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[4] 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*, Sydney, NSW, Australia, 2015, pp. 1235–1244.

[5] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intell. Data Anal.*, vol. 21, no. 6, pp. 1487–1524, 2017.

[6] R. Sharma, D. Gopalani, and Y. Meena, "Collaborative filtering-based recommender system: Approaches and research challenges," in *Proc. Int. Conf. Comput. Intell. Commun. Technol.*, 2017, pp. 1–6.

[7] Y. Li, L. Lu, and X. Li, "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in e-commerce," *Expert Syst. Appl.*, vol. 28, no. 1, pp. 67–77, 2005.

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

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[10] Y.-X. Peng *et al.*, "Cross-media analysis and reasoning: Advances and directions," *Frontiers Inf. Technol.*, vol. 18, no. 5, pp. 44–57, 2017.

[11] A. Karatzoglou *et al.*, "RecSys'16 workshop on deep learning for recommender systems (DLRS)," in *Proc. RecSys*, 2016, pp. 415–416, doi: 10.1145/2959100.2959202.

[12] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 191–198.

[13] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, 2016, pp. 7–10.

[14] C.-Y. Wu, A. Ahmed, A. J. Smola, H. Jing, and A. Beutel, "Recurrent recommender networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Cambridge, U.K., 2017, pp. 495–503.

[15] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, 2015, pp. 278–288.

[16] Y. Song, A. M. Elkahky, and X. He, "Multi-rate deep learning for temporal recommendation," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Pisa, Italy, 2016, pp. 909–912.

[17] F. Vasile, E. Smirnova, and A. Conneau, "Meta-Prod2Vec: Product embeddings using side-information for recommendation," in *Proc. ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 225–232.

[18] M. Grbovic *et al.*, "E-commerce in your inbox: product recommendations at scale," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, 2015, pp. 1809–1818.

[19] C. K. Hsieh, L. Yang, T. Y. Lin, S. Belongie, D. Estrin, and Y. Cui, "Collaborative metric learning," in *Proc. the 26th Int. Conf. World Wide Web*, Perth, WA, Australia, 2017, pp. 193–201.

[20] S. Roy and S. C. Guntuku, "Latent factor representations for cold-start video recommendation," in *Proc. the 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 99–106.

[21] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Melbourne, VIC, Australia, 2015, pp. 811–820.

[22] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Cambridge, U.K., 2017, pp. 425–434.

[23] T. Bansal, D. Belanger, and A. McCallum, "*Ask the GRU*: Multi-task learning for deep text recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 107–114.

[24] Y. Wu, C. DuBois, M. Ester, and X. A. Zheng, "Collaborative denoising auto-encoders for top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, San Francisco, CA, USA, 2016, pp. 153–162.

[25] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 12, no. 4, pp. 4–9, 2009.

[26] K. Verbert *et al.*, "Context-aware recommender systems for learning: A survey and future challenges," *IEEE Trans. Learn. Technol.*, vol. 5, no. 4, pp. 318–335, Oct./Dec. 2012.

[27] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. 5th ACM Conf. Digit. Libraries*, San Antonio, TX, USA, 2000, pp. 195–204.

[28] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, Madison, WI, USA, 1998, pp. 43–52.

[29] M. Balabanović and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[30] J. Ben Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proc. 1st ACM Conf. Electron. Commerce*, Denver, CO, USA, 1999, pp. 158–166.

[31] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 73–105.

[32] T. Di Noia, R. Mirizzi, D. Romito, M. Zanker, and V. C. Ostuni, "Linked open data to support content-based recommender systems," in *Proc. ACM Int. Conf. Semantic Syst.*, 2012, pp. 1–8.

[33] S. Gong, H. Ye, and H. Tan, "Combining memory-based and model-based collaborative filtering in recommender system," in *Proc. IEEE Int. Conf. Pacific-Asia Circuits, Commun. Syst.*, May 2009, pp. 690–693.

[34] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[35] T. Hussein, T. Linder, J. Ziegler, and W. Gaulke, "Hybreed: A software framework for developing context-aware hybrid recommender systems," *User Model. User-Adapted Interact.*, vol. 24, no. 2, pp. 121–174, 2014.

[36] X. Li, M. Wang, and T.-P. Liang, "A multi-theoretical kernel-based approach to social network-based recommendation," *Decision Support Syst.*, vol. 65, pp. 95–104, Sep. 2014.

[37] D. Mukherjee, S. Banerjee, P. Misra, and S. Bhattacharya, "Method and system for context-aware recommendation," U.S. Patent 9 510 050 B2, Nov. 29, 2016.

[38] P. Kosmides, C. Remoundou, I. Loumiotis, E. Adamopoulou, M. Theologou, and K. Demestichas, "A location recommender system for location-based social networks," in *Proc. IEEE Int. Conf. Math. Comput. Sci. Ind.*, Sep. 2015, pp. 277–280.

[39] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[40] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[41] G. E. Hinton and A. Krizhevsky, "Using very deep autoencoders for content-based image retrieval," in *Proc. ESANN*, 2011, pp. 262–270.

[42] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[43] Y. Shan, R. T. Hoens, H. Wang, D. Yu, J. C. Mao, and J. Jiao, "Deep Crossing: Web-scale modeling without manually crafted combinatorial features," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, 2016, pp. 255–262.

[44] J. Zhu, Y. Shan, J. C. Mao, D. Yu, H. Rahmanian, and Y. Zhang. (Mar. 2017). "Deep embedding forest: Forest-based serving with deep embedding features." [Online]. Available: https://arxiv.org/abs/1703.05291

[45] H. Guo, R. Tang, Z. Li, X. He, and Y. Ye, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, 2017, pp. 1725–1731.

[46] G. Zhou *et al.* (Jun. 2017). "Deep interest network for click-through rate prediction." [Online]. Available: https://arxiv.org/abs/1706.06978

[47] D. E. Williams, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–538, 1986.

[48] Y. Bengio, P. Lamblin, H. Larochelle, and D. Popovici, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2007, pp. 153–165.

[49] P. Vincent, H. Larochelle, P.-A. Manzagol, and Y. Bengio, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 1096–1103.

[50] S. Sedhain, A. K. Menon, L. Xie, and S. Sanner, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, 2015, pp. 111–112.

[51] H. Wang, X. Shi, and D.-Y. Yeung, "Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 415–423.

[52] H. Wang, X. Shi, and D.-Y. Yeung, "Relational stacked denoising autoencoder for tag recommendation," in *Proc. 29th Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 3052–3058.

[53] X. Dong, L. Yu, Y. Sun, L. Yuan, F. Zhang, and Z. Wu, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 1309–1315.

[54] J. Wei, J. He, Y. Zhou, Z. Tang, and K. Chen, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, Mar. 2017.

[55] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, 2017, pp. 305–314.

[56] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA, USA: MIT Press, 1986.

[57] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," Ph.D. dissertation, Dept. Comput. Sci., Univ. Colorado Boulder, Boulder, CO, USA, 1986.

[58] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, no. 1, pp. 926–928, 2010.

[59] A. Mnih, G. Hinton, and R. Salakhutdinov, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvalis, OR, USA, 2007, pp. 791–798.

[60] D. Q. Phung and S. Venkatesh, "Ordinal Boltzmann machines for collaborative filtering," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, Richmond, VA, USA, 2009, pp. 548–556.

[61] K. Georgiev and P. Nakov, "A non-IID framework for collaborative filtering with restricted Boltzmann machines," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 1148–1156.

[62] J. Y. He and B. Ma, "Based on real-valued conditional restricted Boltzmann machine and social network for collaborative filtering," (in Chinese), *Chin. J. Comput.*, vol. 39, no. 1, pp. 183–195, 2016.

[63] T. T. Nguyen and H. W. Lauw, "Representation learning for homophilic preferences," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 317–324.

[64] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3104–3112.

[65] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 1764–1772.

[66] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3128–3137.

[67] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[68] K. Cho, M. B. van Merrienboer, Y. Bengio, and D. Bahdanau. (Oct. 2014). "On the properties of neural machine translation: Encoder-decoder approaches." [Online]. Available: https://arxiv.org/abs/1409.1259

[69] J. Weston, S. Chopra, and A. Bordes. (Oct. 2014). "Memory networks." [Online]. Available: https://arxiv.org/abs/1410.3916

[70] A. Joulin and T. Mikolov, "Inferring algorithmic patterns with stack-augmented recurrent nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 190–198.

[71] A. Graves, G. Wayne, and I. Danihelka. (Dec. 2014). "Neural turing machines." [Online]. Available: https://arxiv.org/abs/1410.5401

[72] A. Graves *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[73] Q. Liu, S. Wu, T. Tan, and L. Wang, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 194–200.

[74] Q. Liu, S. Wu, and L. Wang, "Multi-behavioral sequential prediction with recurrent log-bilinear model," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1254–1267, Jan. 2017.

[75] C. Yang, M. Sun, Z. Liu, E. Y. Chang, and X. W. Zhao, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, pp. 36–45, 2017.

[76] B. Hidasi, A. Karatzoglou, D. Tikk, and L. Baltrunas. (Nov. 2015). "Session-based recommendations with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1511.06939

[77] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, 2016, pp. 17–22.

[78] C. Wu, J. Wang, W. Liu, and J. Liu, "Recurrent neural network based recommendation for time heterogeneous feedback," *Knowl.-Based Syst.*, vol. 109, pp. 90–103, Oct. 2016.

[79] H. Dai, Y. Wang, L. Song, and R. Trivedi, "Recurrent coevolutionary feature embedding processes for recommendation," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, 2016, pp. 29–34.

[80] P.-S. Huang, X. He, L. Deng, A. Acero, L. Heck, and J. Gao, "Learning deep structured semantic models for Web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, San Francisco, CA, USA, 2013, pp. 2333–2338.

[81] C. Chen, X. Meng, T. Lukasiewicz, and Z. Xu, "Location-aware personalized news recommendation with deep semantic analysis," *IEEE Access*, vol. 5, pp. 1624–1638, 2017.

[82] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[83] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, USA, 2012, pp. 1097–1105.

[84] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, USA, 2013, pp. 2643–2651.

[85] X. Geng, H. Zhang, T.-S. Chua, and J. Bian, "Learning image and user features for recommendation in social networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 4274–4282.

[86] D. Kim, C. Park, S. Lee, H. Yu, and J. Oh, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 233–240.

[87] Y. S. Rawat and M. S. Kankanhalli, "ConTagNet: Exploiting user context for image tag recommendation," in *Proc. ACM Multimedia Conf.*, Amsterdam, The Netherlands, 2016, pp. 1102–1106.

[88] C. Lei, D. Liu, Z.-J. Zha, H. Li, and W. Li, "Comparative deep learning of hybrid representations for image recommendations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 2545–2553.

[89] S. Wang, Y. Wang, K. Shu, S. Ranganath, H. Liu, and J. Tang, "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in *Proc. 26th Int. Conf. World Wide Web*, Perth, WA, Australia, 2017, pp. 391–400.

[90] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[91] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 627–636.

[92] F. Strub and J. Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs," in *Proc. NIPS Workshop Mach. Learn. eCommerce*, Montreal, QC, Canada, 2015, pp. 150–158.

[93] J. Wang *et al.*, "IRGAN: A minimax game for unifying generative and discriminative information retrieval models," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, 2017, pp. 515–524.

[94] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Diego, CA, USA, 2011, pp. 448–456.

[95] H. Ma, D. Zhou, M. R. Lyu, I. King, and C. Liu, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, Hong Kong, 2011, pp. 287–296.

[96] Q. Liu, S. Wu, Z. Li, L. Wang, and D. Wang, "Context-aware sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining*, Barcelona, Spain, 2016, pp. 1053–1058.

[97] M. Unger, A. Bar, L. Rokach, and B. Shapira, "Towards latent context-aware recommendation systems," *Knowl.-Based Syst.*, vol. 104, pp. 165–178, Jul. 2016.

[98] Z. Xu, C. Chen, Y. Miao, X. Meng, and T. Lukasiewicz, "Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Indianapolis, IN, USA, 2016, pp. 1921–1924.

[99] S. Okura, Y. Tagami, A. Tajima, and S. Ono, "Embedding-based news recommendation for millions of users," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, 2017, pp. 1933–1942.

[100] S. Zhang, L. Yao, Y. Tay, and A. Sun. (2017). "Deep learning based recommender system: A survey and new perspectives." [Online]. Available: https://arxiv.org/abs/1707.07435

[101] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Las Vegas, NV, USA, 2008, pp. 650–658.

[102] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 2052–2057.

[103] L. Hu, J. Cao, L. Cao, Z. Gu, C. Zhu, and G. Xu, "Personalized recommendation via cross-domain triadic factorization," in *Proc. 22nd Int. Conf. World Wide Web*, Rio de Janeiro, Brazil, 2013, pp. 595–606.

[104] W. X. Zhao, J. Wang, J.-R. Wen, E. Y. Chang, X. Li, and Y. He, "Mining product adopter information from online reviews for improving product recommendation," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 3, pp. 29–36, 2016.

[105] J. Chen, H. Zhang, L. Nie, W. Liu, T.-S. Chua, and X. He, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, 2017, pp. 335–344.

[106] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, 2016, pp. 2782–2788.

[107] Q. Zhang, J. Wang, Y. Gong, and H. Huang, "Hashtag recommendation for multimodal microblog using co-attention network," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, 2017, pp. 3420–3426.

[108] Y. Li, T. Liu, L. Zhang, and J. Jiang, "Hashtag recommendation with topical attention-based LSTM," in *Proc. 26th Int. Conf. Comput. Linguistics*, Osaka, Japan, 2016, pp. 970–980.

[109] H. Huang, Q. Zhang, X. Huang, and Y. Gong, "Hashtag recommendation using end-to-end memory networks with hierarchical attention," in *Proc. 26th Int. Conf. Comput. Linguistics*, Osaka, Japan, 2016, pp. 943–952.

[110] X. Wang *et al.*, "Dynamic attention deep model for article recommendation by learning human editors' demonstration," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, 2017, pp. 2051–2059.

[111] Y. Fang and T. Ebesu, "Neural citation network for context-aware citation recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, 2017, pp. 1093–1096.

RUIHUI MU received the M.S. degree in computer technology from the Huazhong University of Science and Technology and the Ph.D. degree from Hohai University. He is currently an Associate Professor with Henan Xinxiang University. His research interests include deep learning, neural networks, and personalized recommender systems.