




# A review on deep learning for recommender systems: challenges and remedies

Zeynep Batmaz<sup>1</sup> · Ali Yurekli<sup>1</sup> · Alper Bilge<sup>1</sup>  · Cihan Kaleli<sup>1</sup>

© Springer Nature B.V. 2018

## Abstract

Recommender systems are effective tools of information filtering that are prevalent due to increasing access to the Internet, personalization trends, and changing habits of computer users. Although existing recommender systems are successful in producing decent recommendations, they still suffer from challenges such as accuracy, scalability, and cold-start. In the last few years, deep learning, the state-of-the-art machine learning technique utilized in many complex tasks, has been employed in recommender systems to improve the quality of recommendations. In this study, we provide a comprehensive review of deep learning-based recommendation approaches to enlighten and guide newbie researchers interested in the subject. We analyze compiled studies within four dimensions which are deep learning models utilized in recommender systems, remedies for the challenges of recommender systems, awareness and prevalence over recommendation domains, and the purposive properties. We also provide a comprehensive quantitative assessment of publications in the field and conclude by discussing gained insights and possible future work on the subject.

**Keywords** Recommender systems · Deep learning · Survey · Accuracy · Scalability · Sparsity

## 1 Introduction

Recent advances in information technologies and prevalence of online services have provided people with the ability to access a massive amount of information quickly. Today, an ordinary user can instantly access descriptions, advertisements, comments, and reviews about almost

---

✉ Alper Bilge  
[abilge@anadolu.edu.tr](mailto:abilge@anadolu.edu.tr)

Zeynep Batmaz  
[zozdemir@anadolu.edu.tr](mailto:zozdemir@anadolu.edu.tr)

Ali Yurekli  
[aliyurekli@anadolu.edu.tr](mailto:aliyurekli@anadolu.edu.tr)

Cihan Kaleli  
[ckaleli@anadolu.edu.tr](mailto:ckaleli@anadolu.edu.tr)

<sup>1</sup> Computer Engineering Department, Anadolu University, 26470 Eskisehir, Turkey

all kinds of products and services. Although accessing information is a valuable ability, people confront a colossal amount of data sources which confuses them to find useful and appropriate content and results in the information overload problem.

Recommender systems are information filtering tools that deal with such problem by providing users with the conceivably exciting content in a personalized manner (Schafer et al. 2001). Currently, many online vendors equip their systems with recommendation engines, and most of the Internet users take advantage of such services in their daily activities such as reading books, listening to music, and shopping. In a typical recommender system, the term item refers to the product or service of which the system recommends to its users. Producing a list of recommended items for the user or predicting how much the user will like a particular item requires a recommender system to either analyze past preferences of like-minded users or benefit from the descriptive information about the items. These two options form the two major approaches in recommender systems, i.e., collaborative filtering (CF) and content-based recommendation (Bobadilla et al. 2013), respectively. There are also hybrid approaches that combine benefits of these two approaches (Burke 2002).

In recent years, artificial neural networks have begun to attract significant attention due to the increasing computational power and big data storage facilities. The researchers successfully build and train deep architectural models (Hinton et al. 2006; Hinton and Salakhutdinov 2006; Bengio 2009) which promotes deep learning as an emerging field of computer science. Currently, many state-of-the-art techniques in image processing, object recognition, natural language processing, and speech recognition utilize deep neural networks as a primary tool. Promising capabilities of deep learning techniques also encourage researchers to employ deep architectures in recommendation tasks, as well (Salakhutdinov et al. 2007; Gunawardana and Meek 2008; Truyen et al. 2009).

In this study, we intensely review applications of deep learning techniques applied in recommender systems field to enlighten and guide researchers interested in the subject. We present the current literature of the research field and reveal a perspectival synopsis of the subject in four distinct strategic directions. Contributions of the review can be listed as follows:

- (i) We present a systematic classification and detailed analysis of deep learning-based recommender systems.
- (ii) We focus on challenges of recommender systems and categorize existing literature based on proposed remedies.
- (iii) We survey on the domain awareness of existing deep learning-based recommender systems.
- (iv) We discuss the state-of-the-art and provide insights by identifying thought-provoking yet under-researched study directions.

The remainder of the article is structured as follows: Sect. 2 reviews the literature in the field briefly, and Sect. 3 provides necessary background information about recommender systems and major deep learning techniques. Section 4 reveals a perspectival synopsis of applied deep learning methodologies within the context of recommender systems. Section 5 presents a quantitative assessment of the comprehensive literature and Sect. 6 presents our insights and discussions on the subject and propose future research directions. Finally, we conclude the study in Sect. 7.

## 2 Related work

The success of deep learning practices has significantly affected research directions in recommender systems, as in many other computer science fields. Initially, Salakhutdinov et al. (2007) presents a way to use a deep hierarchical model for CF on a movie recommendation task. Since this cornerstone study, there have been several attempts to apply deep models into recommender systems research. By utilizing the effectiveness of deep learning at extracting hidden features and relationships, the researchers have proposed alternative solutions to recommendation challenges including accuracy, sparsity, and cold-start problem. Sedhain et al. (2015) achieve higher accuracy by predicting missing ratings of a user-item matrix with the help of autoencoders, and Devooght and Bersini (2017) utilize neural networks to improve short-term prediction accuracy by converting CF into a sequence prediction problem. Wang et al. (2015b) propose a deep model using CF in order to deal with sparsity issue by learning great representations. Furthermore, deep models have been utilized to deal with scalability concern as these models are quite useful in dimensionality reduction and feature extraction. Elkahky et al. (2015) propose a solution for scalability by using deep neural networks to obtain low dimensional features from high dimensional ones, and Louppe (2010) utilizes deep learning for dimensionality reduction to deal with large datasets.

The current popularity of deep architectures brings the need to review and analyze existing studies about deep learning in recommender systems research. A comprehensive analysis may help and guide the researchers who are willing to work on the field. Despite this urgent need, to the extent of our knowledge, only four studies are surveying the subject. Zheng (2016) surveys and critiques the state-of-the-art deep recommendation systems. However, this survey study contains an insufficient number of publications, which results in a very limited perspective over the whole concept. Betru et al. (2017) explain traditional recommender systems and deep learning approaches. This survey is also inadequate regarding its scope since it only analyzes three publications. Liu and Wu (2017) analyze deep learning-based recommendation approaches and come up with a classification framework which categorizes the procedures by input and output aspects. The authors explain the research in limited directions. However, our proposed work provides guidelines to understand more precisely the usage of deep learning-based techniques in recommender systems.

Recently, Zhang et al. (2017a) have published a comprehensive survey on deep learning-based recommender systems. Although the number of reviewed papers in (Zhang et al. 2017a) and this study is very close, the classification approaches demonstrate specific differences. While Zhang et al. (2017a) focus only on a structural classification of publications and propose a two-aspect scheme (neural network models and integration models), we provide a four-dimensional categorization (neural network models, offered remedies, applied domains, and purposive properties). Furthermore, instead of diving into implementation details when examining the publications, we prefer to constitute a general understanding of the subject and lead the way for researchers willing to work on deep learning for recommender systems. Our work allows scholars interested in this topic to understand main effects of utilizing deep learning techniques in recommender systems. This review study focuses on understanding the motivation of using each deep learning-based method in recommender systems. Moreover, it aims to project insights on provided deep learning-based solutions to current challenges of recommender systems.

### 3 Background

While a recommender system can be defined as a particular type of information filtering system, deep learning is a growing trend in machine learning. Before examining how these two fields come together, it is necessary to go over the basics of both subjects. In this background section, we briefly describe the fundamentals, major types, and the primary challenges of recommender systems. Then, we introduce deep learning concept by explaining the factors that promote it as an emerging field of computer science. Finally, we illustrate the deep learning models that have been widely applied in machine learning.

#### 3.1 Recommender systems

While the widespread use of the Internet and increasing data storage capabilities make it easy to access large volumes of data, it becomes harder to find relevant, engaging, and useful content for daily computer users due to information overload.

Over the last few decades, there has been a significant amount of research on computer applications that can discover tailored appropriate content. Recommender systems are one of those applications that can filter information in a personalized manner (Schafer et al. 2001).

Recommender systems produce suggestions and recommendations to assist their users in many decision-making processes. With the help of the recommender systems, users are more likely to access appropriate products and services such as movies, books, music, food, hotels, and restaurants.

In a typical recommender system, the recommendation problem is twofold, i.e., (i) estimating a prediction for an individual item or (ii) ranking items by prediction (Sarwar et al. 2001). While the former process is triggered by the user and focuses on precisely predicting how much the user will like the item in question, the latter process is provided by the recommendation engine itself and offers an ordered top- $N$  list of items that the user might like. Based on the recommendation approach, the recommender systems are classified into three major categories (Adomavicius and Tuzhilin 2005):

*CF recommender systems* produce recommendations to its users based on inclinations of other users with similar tastes.

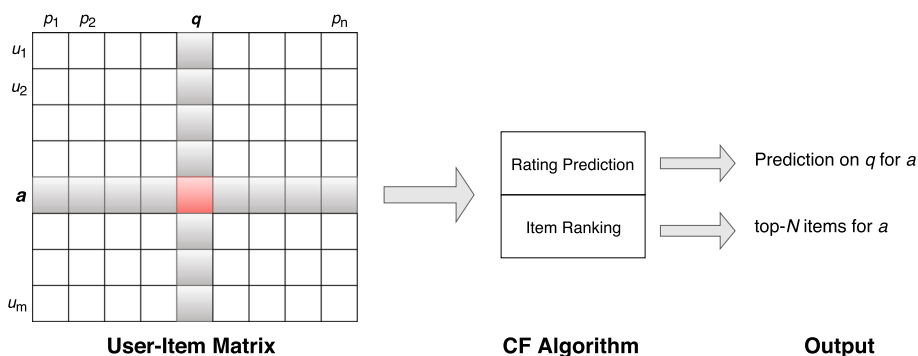
*Content-based recommender systems* generate recommendations based on similarities of new items to those that the user liked in the past by exploiting the descriptive characteristics of items.

*Hybrid recommender systems* utilize multiple approaches together, and they overcome disadvantages of certain approaches by exploiting compensations of the other.

Beside these common recommender systems, there are some specific recommendation techniques, as well. Specifically, *context-aware* recommender systems incorporate contextual information of users into the recommendation process (Verbert et al. 2010), *tag-aware* recommender systems integrate product tags to standard CF algorithms (Tso-Sutter et al. 2008), *trust-based* recommender systems take the trust relationship among users into account (Bedi et al. 2007), and *group-based* recommender systems focus on personalizing recommendations at the group of users level (McCarthy et al. 2006).

##### 3.1.1 Collaborative filtering recommender systems

CF is the most prominent approach in recommender systems which makes the assumption that people who agree on their tastes in the past would agree in the future, as well (Sarwar



**Fig. 1** An overview of the CF process

et al. 2001). In such systems, preferences of like-minded neighbor users form the basis of all produced recommendations rather than individual features of items.

The primary actor of a CF system is the active user ( $a$ ) who seeks for a rating prediction or ranking of items. By utilizing past preferences as an indicator for determining correlation among users, a CF recommender yield referrals to  $a$  relying on tastes of compatible users.

Typically, a CF system contains a list of  $m$  users  $U = \{u_1, u_2, \dots, u_m\}$  and  $n$  items  $P = \{p_1, p_2, \dots, p_n\}$ . The system constructs an  $m \times n$  user-item matrix that contains the user ratings for items, where each entry  $r_{i,j}$  denotes the rating given by user  $u_i$  for item  $p_j$ . In need of a referral for the  $a$  on the target item  $q$ , the CF algorithm either predicts a rating for  $q$  or recommends a list of most likable top- $N$  items for  $a$ . An overview of the general CF process is illustrated in Fig. 1.

CF algorithms follow two main methodologies approaching the recommendation generation problem:

*Memory-based algorithms* utilize the entire user-item matrix to identify similar entities. After locating the nearest neighbors, past ratings of these entities are employed for recommendation purposes (Breese et al. 1998).

Memory-based algorithms can be user-based, item-based, or hybridized. While past preferences of nearest neighbors to  $a$  are employed in user-based CF, the ratings of similar items to  $q$  are used in item-based approach (Aggarwal 2016).

*Model-based algorithms* aim to build an offline model by applying machine learning and data mining techniques. Building and training such model allows estimating predictions for online CF tasks. Model-based CF algorithms include Bayesian models, clustering models, decision trees, and singular value decomposition models (Su and Khoshgoftaar 2009).

### 3.1.2 Content-based recommender systems

Content-based recommender systems produce recommendations based on the descriptive attributes of items and the profiles of users (Van Meteren and Van Someren 2000). In content-based filtering, the main purpose is to recommend items that are similar to those that a user liked in the past. For instance, if a user likes a website that contains keywords such as “stack”, “queue”, and “sorting”, a content-based recommender system would suggest pages related with data structures and algorithms.

Content-based filtering is very efficient when recommending a freshly inserted item into the system. Although there exists no history of ratings for the new item, the algorithm can benefit from the descriptive information and recommend it to the relevant users. For instance, a new science fiction movie might be suggested to a user who has previously seen and liked movies “The Terminator” and “The Matrix”.

Although content-based recommender systems are effective at recommending new items, they cannot produce personalized predictions since there is not enough information about the profile of the user. Furthermore, the recommendations are limited in terms of diversity and novelty since the algorithms do not leverage the community knowledge from like-minded users (Lops et al. 2011).

### 3.1.3 Hybrid recommender systems

Both CF systems and content-based recommenders have idiosyncratic strengths and weaknesses. Hybrid recommender systems, on the other hand, combine CF and content-based methods to avoid limitations of each approach by exploiting the benefits of the other. A typical hybridization scenario would be employing content-based descriptive information of a new item without any user rating in a CF recommender system (Tran and Cohen 2000). Various hybridization techniques have been proposed which can be summarized as follows (Burke 2002):

*Weighted:* A single recommendation output is produced by combining scores of different recommendation approaches.

*Switching:* Recommendation outputs are selectively produced by either algorithm depending on the current situation.

*Mixed:* Recommendation outputs of both approaches are shown at the same time.

*Cascade:* Recommendation outputs produced by an approach are refined by the other approach.

*Feature combination:* Features from both approaches are combined and utilized in a single algorithm.

*Feature augmentation:* Recommendation output of an approach is utilized as the input of the other approach.

### 3.1.4 Challenges of recommender systems

Even though recommender systems provide efficient ways to deal with the information overload problem, they also come up with many different challenges (Su and Khoshgoftaar 2009; Bobadilla et al. 2013). In this section, we briefly describe the major issues in recommender systems including accuracy, sparsity, cold-start, and scalability.

One of the critical requirements of a recommender system is to bring thrilling and relevant items to the users. The trust level of the users to the system is directly related to the quality of recommendations. If the users are not provided with favorable products and services, the recommendation engine might be considered inadequate regarding customer satisfaction, which makes it evident for users to look for alternative systems. Therefore, a recommender system must satisfy an appropriate level of prediction accuracy to improve preferability and effectiveness. Accuracy as being the most discussed challenge of recommender systems is commonly investigated in three means as the accuracy of rating predictions, usage predictions, and ranking of items (Shani and Gunawardana 2011).

CF systems rely on the rating history of the items given by the users of the system. Sparsity appears as a major problem especially for CF since the users only rate a small fraction of the available items, which makes it challenging to generate predictions (Su and Khoshgoftaar 2009; Bobadilla et al. 2013). When working on a sparse dataset, a CF algorithm may fail to take advantage of beneficial relationships among users and items. Data sparsity leads to another severe challenge referred to as the cold-start problem. Producing predictions for a new user having very few ratings is not possible due to insufficient data to profile them. Likewise, presenting recently added items as recommendations to users is also not achievable due to the lack of ratings for those items. However, unlike CF techniques, newly added users and items can be managed in content-based recommender systems by utilizing their content information.

Most of the recommender systems are deployed in a responsive environment. In a typical recommendation scenario, a user is provided with a set of recommended items according to her preferences while she is navigating through a web page. In order to carry out such scenario efficiently, the recommendations should be provided in a reasonable amount of time, which requires a highly-scalable system (Linden et al. 2003). With the growth of the number of users and/or items in the system, many algorithms tend to slow down or require more computational resources (Shani and Gunawardana 2011). Thus, scalability turns into a significant challenge that should be managed efficiently.

### 3.2 Deep learning

Deep learning is a field of machine learning that is based on learning several layers of representations, typically by using artificial neural networks. Through the layer hierarchy of a deep learning model, the higher-level concepts are defined from the lower-level concepts (Deng and Yu 2014).

Since Hinton et al. (2006) introduced an efficient way of training deep models and Bengio (2009) showed the capabilities of deep architectures in complicated artificial intelligence tasks, deep learning has become an emerging topic in computer science. Currently, deep learning approaches produce the state-of-the-art solutions to many problems in computer vision, natural language processing, and speech recognition (Deng and Yu 2014).

Although neural networks and the science behind deep models have been around for more than 50 years, the power of deep learning techniques has started revealing during the last decade. The main factors that promote deep learning as the state-of-the-art machine learning technique can be listed as follows:

*Big data:* A deep learning model learns better representations as it is provided with more amount of data.

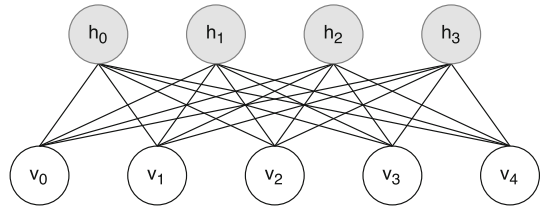
*Computational power:* Graphical processing units (GPU) meet the processing power required for complex computations in deep learning models.

Throughout this section, we briefly introduce the deep learning models that have been widely utilized in recommendation tasks.

#### 3.2.1 Restricted Boltzmann machines

A restricted Boltzmann machine (RBM) is a particular type of a Boltzmann machine, which has two layers of units. As illustrated in Fig. 2, the first layer consists of visible units, and the second layer includes hidden units. In this restricted architecture, there are no connections between the units in a layer (Salakhutdinov and Hinton 2009).

**Fig. 2** A restricted Boltzmann machine



The visible units in the model correspond to the components of observation, and the hidden units represent the dependencies between the components of the observations. For instance, in case of the famous handwritten digit recognition problem (Cireşan et al. 2010), a visible unit becomes a pixel of a digital image, and a hidden unit represents a dependency between pixels in the image.

### 3.2.2 Deep belief networks

A deep belief network (DBN) is a multi-layer learning architecture that uses a stack of RBMs to extract a deep hierarchical representation of the training data. In such design, the hidden layer of each sub-network serves as the visible layer for the upcoming sub-network (Hinton 2009).

When learning through a DBN, firstly the RBM in the bottom layer is trained by inputting the original data into the visible units. Then, the parameters are fixed up, and the hidden units of the RBM are used as the input into the RBM in the second layer. The learning process continues until reaching the top of the stacked sub-networks, and finally, a suitable model is obtained to extract features from the input. Since the learning process is unsupervised, it is common to add a new network of supervised learning to the end of the DBN to use it in a supervised learning task such as classification or regression.

### 3.2.3 Autoencoders

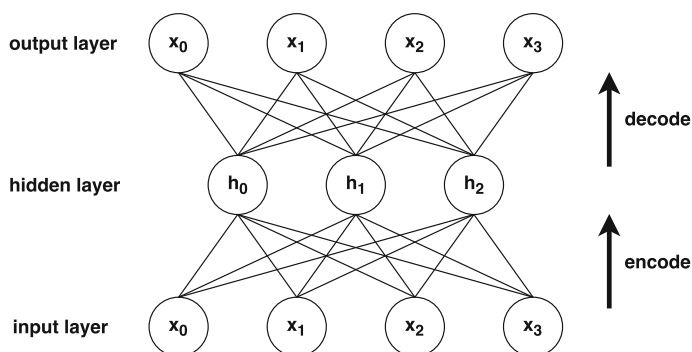
An autoencoder is a type of feedforward neural network, which is trained to encode the input into some representation, such that the input can be reconstructed from such representation (Hinton and Salakhutdinov 2006). Typically, an autoencoder consists of three layers, namely, the input layer, the hidden layer, and the output layer. The number of neurons in the input layer is equal to the number of neurons in the output layer.

An autoencoder reconstructs the input layer at the output layer by using the representation obtained in the hidden layer. During the learning process, the network uses two mappings, which are referred to as encoder and decoder. While the encoder maps the data from the input layer to the hidden layer, the decoder maps the encoded data from the hidden layer to the output layer. An illustration of an autoencoder is given in Fig. 3.

Reconstruction strategy in autoencoders may fail to extract useful features. The resulting model may result in uninteresting solutions, or it may provide a direct copy of the original input. In order to avoid such kind of problems, a denoising factor is used on the original data.

A denoising autoencoder (DAE) is a variant of an autoencoder that is trained to reconstruct the original input from the corrupted form. Denoising factor makes autoencoders more stable and robust since they can deal with data corruptions (Vincent et al. 2010).





**Fig. 3** An autoencoder

Similar to the way in combining RBMs to build deep belief networks, the autoencoders can be stacked to create deep architectures. A stacked denoising autoencoder (SDAE) is composed of multiple DAEs one on top of each other.

### 3.2.4 Recurrent neural networks

A recurrent neural network (RNN) is a class of artificial neural networks that make use of sequential information (Donahue et al. 2015). An RNN is specialized to process a sequence of values  $x^{(0)}, x^{(1)}, \dots, x^{(t)}$ . The same task is performed on every element of a sequence, while the output depends on the previous computations. In other words, RNNs have internal memory that captures information about previous calculations.

Despite the fact that RNNs are designed to deal with long-term dependencies, vanilla RNNs tend to suffer from vanishing or exploding gradient (Hochreiter and Schmidhuber 1997). When backpropagation trains the network through time, the gradient is passed back through many time steps, and it tends to vanish or explode. The popular solutions to this problem are Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures.

### 3.2.5 Convolutional neural networks

A convolutional neural network (CNN) is a type of feed-forward neural network which applies convolution operation in place of general matrix multiplication in at least one of its layers. CNNs have been successfully applied in many difficult tasks like image and object recognition, audio processing, and self-driving cars.

A typical CNN consists of three components that transform the input volume into an output volume, namely, convolutional layers, pooling layers, and fully connected layers. These layers are stacked to form convolutional network architectures as illustrated in Fig. 4.

In a typical image classification task using a CNN, the layers of the network carry out following operations.

1. *Convolution*: As being the core operation, convolutions aim to extract features from the input. Feature maps are obtained by applying convolution filters with a set of mathematical operations.
2. *Nonlinearity*: In order to introduce nonlinearities into the model, an additional operation, usually ReLU (Rectified Linear Unit), is used after every convolution operation.

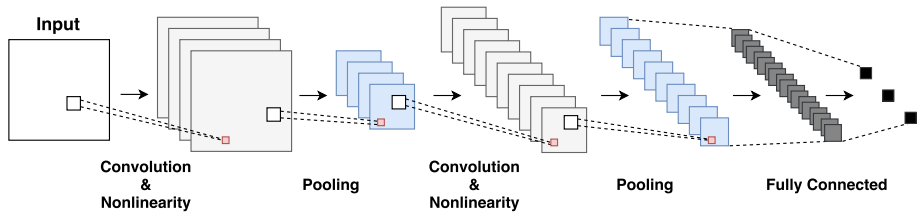


Fig. 4 A convolutional neural network

3. *Pooling (Subsampling)*: Pooling reduces the dimensionality of the feature maps to decrease processing time.
4. *Classification*: The output from the convolutional and pooling layers represents high-level features of the input. These features can be used within the fully connected layers for classification (Zhang and Wallace 2015).

## 4 Perspectival synopsis of deep learning within recommender systems

The application of deep learning techniques to the recommendation domain is a favorite and trending topic. Deep learning is beneficial in analyzing data from multiple sources and discovering hidden features. Since deep learning techniques' ability of data processing is on the rise due to the advances in big data facilities and supercomputers, the researchers have already started to benefit from deep learning techniques in recommender systems. They have utilized deep learning techniques to produce practical solutions to the challenges of recommender systems such as scalability and sparsity. Moreover, they have used deep learning for producing recommendations, dimensionality reduction, feature extraction from different data sources and integrating them into the recommendation systems. Deep learning techniques are utilized in recommender systems to model either user-item preference matrix or content/side information, and sometimes both of them. Table 1 demonstrates publications which use deep learning techniques in data modeling practices.

### 4.1 Deep learning techniques for recommendation

In this section, we analyze how and for what purpose deep learning methods are utilized in recommender systems. The techniques that are described throughout this section are **RBMs**, **DBNs**, **autoencoders**, **RNNs**, and **CNNs**. Furthermore, some less conventional methods are also analyzed under other techniques subsection.

#### 4.1.1 Restricted Boltzmann machines for recommendation

**RBMs are particular types of Boltzmann machines, and they have two types of layers, which are visible softmax layer and hidden layer.** In an RBM, there is no intra-layer communication. RBMs are used to extract latent features of user preferences or item ratings in recommendation domain (Salakhutdinov et al. 2007; Deng et al. 2017). RBMs are also utilized for jointly modeling of both correlations between a user's voted items and the correlation between the users who voted a particular item to improve the accuracy of a recommendation system (Georgiev and Nakov 2013). RBMs are also used in group-based recommender systems to

**Table 1** Purpose of using deep learning in publications regarding data modeling

User-item preference matrix	Content/side information	Both
Salakhutdinov et al. (2007)	Oord et al. (2013)	Wang et al. (2015b)
Truyen et al. (2009)	Wang and Wang (2014)	Li et al. (2015)
Georgiev and Nakov (2013)	Elkahky et al. (2015)	Jia et al. (2015)
Sedhain et al. (2015)	Shin et al. (2015)	Lei et al. (2016)
Strub and Mary (2015)	Zhou et al. (2016)	Covington et al. (2016)
Du et al. (2016)	Shen et al. (2016)	Strub et al. (2016)
Zheng et al. (2016b)	Zhang et al. (2016b)	Gunawardana and Meek (2008)
Wu et al. (2016b)	Vuurens et al. (2016)	Devooght and Bersini (2017)
Baalen (2016)	Zanotti et al. (2016)	Dai et al. (2017)
Deng et al. (2017)	Unger et al. (2016)	Ying et al. (2016)
Zuo et al. (2016)	Wei et al. (2017)	Ko et al. (2016)
Ouyang et al. (2014)	Wang et al. (2015a)	Zhang et al. (2016a)
Hu et al. (2014)	Huang et al. (2015)	Paradarami et al. (2017)
Zheng et al. (2016a)	Kyo-Joong et al. (2014)	Yang et al. (2017)
Du et al. (2017)	Tang et al. (2015)	Lian et al. (2017)
Barbieri et al. (2017)	Suglia et al. (2017)	Li and She (2017)
Cao et al. (2017)	Oramas et al. (2017)	Kumar et al. (2017)
Suzuki and Ozaki (2017)	Obadić et al. (2017)	Soh et al. (2017)
Xu et al. (2017b)	Dominguez et al. (2017)	Ebesu and Fang (2017)
Hsieh et al. (2017)	Wu et al. (2017b)	Wu et al. (2017a)
Zhuang et al. (2017)	Jaradat (2017)	Zhao et al. (2018)
Zhang et al. (2018)	Kim et al. (2017)	Yin et al. (2017)
	Shankar et al. (2017)	Pana et al. (2017)
	He et al. (2017)	Zhang et al. (2017c)
	Wang et al. (2017)	Wang et al. (2016)
	Xu et al. (2017a)	Volkovs et al. (2017)
	Rassweiler Filho et al. (2017)	
	Vall et al. (2017)	
	Nguyen et al. (2017)	
	Hassan (2017)	
	Seo et al. (2017b)	
	Nedelec et al. (2017)	
	Peska and Trojanova (2017)	
	Deldjoo et al. (2017)	
	Zhang et al. (2017b)	
	Kim et al. (2016)	

model group preferences by jointly modeling collective features and group profiles (Hu et al. 2014).

Truyen et al. (2009) utilize Boltzmann machines to extract both the relation between a rated item and its rating and the correlations between rated items thanks to the connections between the hidden layer and softmax layer, and the connections between softmax

layer units, respectively. Boltzmann machines are utilized in recommender systems to model pairwise-correlation between items or users. Moreover, Gunawardana and Meek (2008) employ Boltzmann machines not only for modeling correlation between users and items but also for integrating content information. They tie the machines' parameters with content information.

RBMs are used primarily for providing a low-rank representation of user preferences. On the other hand, Boltzmann machines are used for integrating correlations between the user or item pairs, and neighborhood formation within visible layers. Combining both user-user and item-item correlations via RBMs is possible by generating a hybrid model in which hidden layers are connected to two visible layers (one for items and one for users) (Georgiev and Nakov 2013). However, using Boltzmann machines to model the pairwise user or item correlations and neighborhood formation performs with higher accuracy (Georgiev and Nakov 2013). Since RBMs have more straightforward parametrization and are more scalable than Boltzmann machines, they might be preferable when the pairwise user and item correlations are considered (Georgiev and Nakov 2013). RBMs allow handling large datasets, as well. Additionally, both RBMs and Boltzmann machines enable integrating auxiliary information from different data sources.

#### 4.1.2 Deep belief networks for recommendation

One of the applications of DBNs on recommender systems domain focuses on extracting hidden and useful features from **audio content for content-based and hybrid music recommendation** (Wang and Wang 2014). **DBNs are also utilized in recommender systems based on text data** (Kyo-Joong et al. 2014; Zhao et al. 2015). **Furthermore, DBNs are applied in content-based recommender systems as a classifier to analyze user preferences, especially on textual data** (Kyo-Joong et al. 2014). **Semantic representation of words is provided by utilizing DBNs** (Zhao et al. 2015). **Additionally, DBNs are used for extracting high-level features from low-level features on user preferences** (Hu et al. 2014). **These studies reveal that DBNs are used mostly for extracting features and classification tasks, especially on textual and audio data.**

#### 4.1.3 Autoencoders for recommendation

A simple autoencoder compresses given data with the encoder part and reconstructs data from its compressed version through the decoder. Autoencoders try to reconstruct the initial data through a dimensionality reduction operation. Such type of deep models is used in recommender systems to **learn a non-linear representation of user-item matrix and reconstruct it by determining missing values** (Ouyang et al. 2014; Sedhain et al. 2015). Autoencoders are also used for dimensionality reduction and extracting more latent features by using the output values of the encoder parts (Deng et al. 2017; Zuo et al. 2016; Unger et al. 2016). Moreover, sparse coding is applied to autoencoders to learn more effective features (Zuo et al. 2016).

**DAEs are special forms of autoencoders where input data is corrupted to prevent becoming an identity network. SDAEs are simply many autoencoders that are stacked on top of each other.** These autoencoders provide the ability to extract more hidden features. **DAEs are used in recommender systems to predict missing values from corrupted data** (Wu et al. 2016b), **and SDAEs help recommender systems to find out a denser form of the input matrix** (Strub and Mary 2015). Moreover, they are also helpful for integrating auxiliary information to a recommender system by allowing data from multiple data sources (Wang et al. 2015a, b; Li

et al. 2015; Strub et al. 2016; Ying et al. 2016; Wei et al. 2017). Wang et al. (2015b) utilize Bayesian SDAE whereas Li et al. (2015) utilize marginalized DAE to integrate auxiliary information into their recommender systems. Wang et al. (2015a) also propose relational SDAE by generating a probabilistic form of SDAEs to integrate auxiliary data to ratings. Since marginalized DAE is more scalable and faster than DAE, it becomes an attractive deep learning tool in recommender systems. In (Wang et al. 2015b), auxiliary data is integrated only at the input level. Unlike (Wang et al. 2015b), auxiliary information is integrated to every layer of SDAE in (Strub et al. 2016). In (Strub et al. 2016; Wei et al. 2017), features are extracted from auxiliary data with autoencoders and tightly coupled with CF methods as probabilistic matrix factorization and timeSVD++ for (Strub et al. 2016) and (Wei et al. 2017), respectively to solve sparsity problem. Ying et al. (2016) extract latent features of side information with SDAEs and integrate them into the Bayesian framework of the pair-wise ranking model.

The studies in this field state that, autoencoders provide more accurate recommendations compared to RBMs. One of the reasons for such situation is that RBMs produce predictions by maximizing log likelihood whereas autoencoders by minimizing Root Mean Square Error (RMSE) which is one of the most commonly used accuracy metrics for recommender systems. Moreover, training phase of autoencoders is comparatively faster than RBMs due to the used methods such as gradient-based backpropagation for autoencoders and contrastive divergence for RBMs (Sedhain et al. 2015). Stacked autoencoders provide more accurate predictions than non-stacked forms since stacking autoencoders allows learning deeply more hidden features (Li et al. 2015). Autoencoders are used for many purposes in recommender systems such as feature extracting, dimensionality reduction, and producing predictions. Autoencoders are utilized in recommender systems, especially for handling with sparsity and scalability problems.

#### 4.1.4 Recurrent neural networks for recommendation

RNNs are specialized for processing a sequence of information. In an e-commerce system, a user's current browsing history affects her purchase behaviors. However, most of the typical recommender systems create user preferences at the beginning of a session, which results in overlooking the current history and the order of sequences of user actions. RNNs are utilized in recommender systems to integrate current viewing web page history and order of the views to provide more accurate recommendations (Wu et al. 2016a; Hidasi et al. 2016a; Tan et al. 2016). Wu et al. (2016a) also merge results of RNN with the outcome of a feed-forward neural network to consider user-item correlations in producing predictions. Ko et al. (2016) utilize RNNs to represent temporal, contextual aspects of user behaviors to improve more accurate recommendations by combining these representations with latent factors of user preferences.

RNNs are also used for non-linearly representing influence between users' and items' latent features and coevolution of them over time (Dai et al. 2017). Devooght and Bersini (2017) utilize RNNs in order to integrate the evolution of user tastes into recommendation process by considering the issue as a sequence prediction problem.

It is possible to obtain several findings when the studies about deep learning on recommender systems are analyzed. RNNs have positive effects on coverage of recommendations and short-term predictions (predicting the next consumable product) compared to conventional nearest-neighbor and matrix factorization-based approaches. Such success originates from accounting the evolution of users tastes and coevolution between user and item latent features by RNNs (Devooght and Bersini 2017; Dai et al. 2017). Moreover, RNNs are good

choices especially for session-based recommender systems and integrating users' implicit behaviors to their preferences.

#### 4.1.5 Convolutional neural networks for recommendation

A CNN uses convolution with at least one of its layers, and such type of neural networks are used for particular tasks such as image recognition and object classification. Recommender systems also benefit from CNNs. Oord et al. (2013) utilize CNNs to extract latent factors from audio data when the factors cannot be obtained from the feedbacks of users. Shen et al. (2016) use CNNs to extract latent factors from text data. Zhou et al. (2016) extract visual features with the purpose of generating visual interest profiles of users for the recommendation. Lei et al. (2016) utilize CNNs to extract latent features of images with the purpose of mapping the features and user preferences into the same latent space. Semantic meanings of textual information extracted with CNNs are also utilized in recommender systems especially for context-aware recommender systems to provide more qualified recommendations (Wu et al. 2017b). As a result, CNNs are mainly used for extracting latent factors and features from data, especially from images and text, for recommendation purposes.

#### 4.1.6 Other techniques

Neural autoregressive distribution estimation (NADE) is an alternative form of an RBM, which provides tractable conditional distributions for discrete variables (Larochelle and Murray 2011). Zheng et al. (2016b) apply NADE on user-item preferences for CF purposes. Authors extract conditional hidden representations of user-item preferences to produce predictions. Compared to RBM-based CF (Salakhutdinov et al. 2007), NADE provides more accurate recommendations, and it is more efficiently optimizable (Zheng et al. 2016b). Du et al. (2016) utilize NADE in CF by modeling ratings across both all users and items simultaneously.

Feedforward neural networks are used in recommender systems for classification purposes and producing predictions (Wakita et al. 2016; Cheng et al. 2016; Zhang et al. 2018). Zhang et al. (2018) utilize multi-layer neural networks in order to non-linearly model interactions between users and items from implicit feedback for personalized ranking of items.

### 4.2 Remedies for challenges of recommender systems

One of the directions of utilizing deep learning methods in recommender systems is producing effective solutions to the challenges of recommender systems. In this section, we analyze deep learning-based studies regarding the provided solutions to the difficulties of recommender systems.

#### 4.2.1 Solutions for improving accuracy

One of the main purposes of employing deep learning techniques in recommender systems is to improve the accuracy of produced predictions. Since deep learning techniques are successful in extracting hidden features, researchers utilize them to extract latent factors. Salakhutdinov et al. (2007) demonstrate that combining RBM models with Singular Value Decomposition (SVD) provide more accurate predictions than Netflix recommendation system. Sedhain et al. (2015) propose AutoRec which uses autoencoders as a predictor

for missing ratings. Experimental results imply that AutoRec outperforms biased matrix factorization, RBM-based CF (Salakhutdinov et al. 2007), and local low-rank matrix factorization (LLORMA) regarding accuracy on MovieLens and Netflix datasets. Zheng et al. (2016b) apply NADE to CF process. In order to improve the accuracy performance of the proposed algorithm, they generate the model based on items by sharing parameters between different ratings of the same item and apply their model into a deeper form. Their experiments present that the proposed method precedes the state-of-the-art algorithms such as LLORMA, AutoRec (Sedhain et al. 2015), RBM-based CF (Salakhutdinov et al. 2007), and several matrix factorization-based approaches on MovieLens and Netflix datasets. Wu et al. (2016b) propose Collaborative Denoising Autoencoder (CDAE) for top- $N$  recommendation by reconstructing the dense form of user preferences. They demonstrate the effects of CDAE's main components (mapping function, loss function, and corruption level) on the accuracy performance. They generate four variants of their model by defining mapping functions of hidden and output layers as sigmoid and identity, loss functions as square and logistic, corruption levels as 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0. They perform experiments to reveal the best variant on MovieLens, Netflix, and Yelp datasets. Empirical results show that the best variant depends on the dataset and non-linear functions improve the recommendations. Moreover, according to their results, CDAE outperforms the state-of-the-art top- $N$  recommendation methods regarding accuracy. Unger et al. (2016) utilize autoencoders and principal component analysis (PCA) to represent latent context features from data collected from sensors to improve the accuracy of context-aware recommender systems. They also combine specific contextual features with latent context features learned by PCA or autoencoders and compare them with explicit context model and matrix factorization model regarding accuracy. Their experimental results confirm that utilizing only latent contextual features using autoencoders has better performance, especially when positive and negative feedbacks are provided. Deng et al. (2017) employ deep autoencoder to optimally initialize user- and item-latent features in the initialization phase of matrix factorization to improve accuracy in trust-aware recommender systems. Their results show that the proposed method precedes the state-of-the-art trust-aware recommendation algorithms.

The efficiency of extracting hidden features makes deep learning approach highly preferable. In recommender systems, deep learning is commonly used to obtain features of users and items, generate a joint model of either user- and item-based approaches or auxiliary information with preference information. Truyen et al. (2009) integrate latent features of user preferences with correlations of intra-users and intra-items with Boltzmann machines. They compare user-based Boltzmann machines (a graphical model of each user's latent aspects and ratings), item-based Boltzmann machines (a graphical model of item ratings), joint modeling of both user- and item-centric processes and joint modeling of both user- and item-correlations with SVD algorithm. According to their experimental results, they manage to outperform SVD especially for the joint model with correlation on MovieLens dataset. Du et al. (2016) propose a neural autoregressive model for CF, and they aim to improve accuracy by generating autoregressive structure in domains of both users and items. Their experimental results show that the proposed algorithm transcends the state-of-the-art methods including some matrix factorization-based methods and neural network-based methods such as AutoRec (Sedhain et al. 2015) and CF-NADE (Zheng et al. 2016b) on MovieLens and Netflix datasets regarding accuracy. Wang et al. (2015a) propose a hybrid tag-aware recommendation system which integrates auxiliary information with SDAEs to improve accuracy. Moreover, they propose a probabilistic SDAE to learn the relation between items, and then combine layered representational learning and relational learning called relational SDAE. Their results show that the proposed relational SDAE outperforms the state-of-the-art tag-



aware recommendation methods on CiteULike and MovieLens datasets. Hu et al. (2014) utilize deep learning techniques in a group-based recommendation to improve accuracy. They utilize deep learning techniques in both extracting collective features from group members' features and combining collective features with group preferences. Their experimental results demonstrate that deep learning provides the proposed recommendation algorithm to have better performance compared to the state-of-the-art methods.

Deep learning techniques are useful for monitoring evolution of user tastes in time which helps to improve the quality of recommendations. Dai et al. (2017) utilize RNNs to non-linearly model co-evolutionary nature of user-item interactions in user behavior prediction to improve accuracy. According to their results, regarding accuracy, the proposed method outperforms the state-of-the-art methods modeling user-item interactions such as LowRankHawkes, Coevolving, PoissonTensor, and timeSVD++ on datasets of IPTV, Reddit, and Yelp. Devooght and Bersini (2017) utilize RNNs to improve short-term prediction accuracy and item coverage in CF by converting the recommendation process into a sequence prediction problem. Utilizing RNNs, they consider not only the preferences of users but also the order of their preferences. Their results confirm that the proposed method outperforms the state-of-the-art top- $N$  recommendation methods regarding short-term prediction accuracy. Wu et al. (2016a) utilize deep RNNs to provide real-time recommendations on current user browsing patterns. They link their model with feedforward neural networks to simulate CF technique by considering user purchase history. Wang et al. (2016) propose a collaborative recurrent autoencoder to generate a hybrid recommender system to improve recommendation accuracy by jointly modeling generation of sequences and implicit relationships between users and items.

According to studies, the main motivation behind the capacity of deep learning to improve recommendation accuracy comes from its power on extracting hidden features and jointly combining information from varying sources. Table 2 compiles RMSE values from various studies that utilize only user-item matrix as the recommendation source. The experiments performed on MovieLens 1M dataset show that CF-NADE (Zheng et al. 2016b), AutoRec (Sedhain et al. 2015), and CF-UserItem-CoAutoregressive (Du et al. 2016) are very successful models compared to the state-of-the-art algorithms. As shown in Table 2, item-based algorithms are more accurate than user-based ones. The reason for such improvement is that the number of average ratings per item is much greater than the number of average ratings per user. Moreover, the researchers show that designing a recommendation system jointly based on both user and item features improves accuracy. Furthermore, converting utilized deep learning technique into deeper form makes the neural network to learn more hidden representations. In this way, the produced predictions become more accurate. As it can be followed in Table 2, U-CF-NADE with single layer provide less precise recommendations compared to U-CF-NADE with two layers. Accordingly, the same relation applies to I-CF-NADE with one layer and two layers. Moreover, Sedhain et al. (2015) experimented with a deep I-AutoRec with three hidden layers and according to the obtained results, deep and single layer implementations of I-AutoRec output 0.827 and 0.831 RMSE values, respectively.

#### 4.2.2 Solutions for sparsity and cold-start problems

One of the solutions for CF-based recommender systems to overcome sparsity problem is to transform the high-dimensional and sparse user-item matrix into a lower-dimensional and denser set using deep learning techniques (Georgiev and Nakov 2013; Strub and Mary 2015; Strub et al. 2016; Unger et al. 2016; Yang et al. 2017; Shu et al. 2018; Du et al.



**Table 2** RMSE values of models on MovieLens 1M dataset (Du et al. 2016)

Model	RMSE
PMF	0.883
U-RBM	0.881
U-AutoRec	0.874
LLORMA-GLOBAL	0.865
I-RBM	0.854
U-CF-NADE-SingleLayer	0.850
U-CF-NADE-TwoLayers	0.845
BiasMF	0.845
NNMF	0.843
LLORMA-LOCAL	0.833
I-AutoRec	0.831
I-CF-NADE-SingleLayer	0.830
I-CF-NADE-TwoLayers	0.829
CF-UserItem-CoAutoregressive	0.823

2017). For instance, Unger et al. (2016) utilize autoencoders to deal with sparsity in context-aware recommender systems caused by integrating high context features where authors derive low-dimensional latent representations of context features extracted from sensors with autoencoders.

Consequences of data sparsity problem within recommender systems are often relieved by utilizing extra side information (Singh and Gordon 2010; Ma et al. 2011). However, latent factors might not be useful due to the sparsity preference matrix and side information. Thus, deep learning techniques are used to extract high-level representations of users and items based on preference matrix and side information which then incorporated with matrix factorization to deal with sparsity and cold-start problems (Wang et al. 2015a, b; Li et al. 2015; Oord et al. 2013; Wang and Wang 2014; Xu et al. 2017a; Zhang et al. 2017b; Kim et al. 2017). Wang et al. (2015b) propose collaborative deep learning as a tightly-coupled method to deal with sparsity problem by learning high-level representations through Bayesian SDAE. Li et al. (2015) integrate matrix factorization-based CF and deeply learned features via marginalized DAE, and Oord et al. (2013) utilize CNN to extract high-level features from music audio signals to deal with the cold-start problem in CF-based approaches. Furthermore, Wang and Wang (2014) propose a content-based model and a hybrid model based on DBNs for handling warm-start and cold-start stages without relying on CF. The proposed hybrid method employs probabilistic matrix factorization together with the features learned via DBNs. Wang et al. (2015a) propose relational probabilistic SDAE to extract features from content information and relation between items aiming to handle sparsity problem by integrating the extracted features to matrix factorization. Wei et al. (2017) integrate timeSVD++ and features extracted from content information with SDAEs to deal with cold-start and sparsity problems. Shin et al. (2015) also utilize side information to deal with sparsity and cold-start problems for blog recommendation. They integrate extracted features from text and images with *word2vec* and CNNs, respectively into their proposed boosted inductive matrix completion method. Shen et al. (2016) integrate latent factors extracted with CNNs into matrix factorization using the latent factor model to deal with sparsity problem. Ying et al. (2016) extracts deep features

from content information with SDAE and integrate them into the Bayesian framework of pair-wise ranking model aiming to handle sparsity problem.

Content information is directly integrated to recommendation process to deal with the sparsity and cold-start problems in some studies (Strub et al. 2016; Gunawardana and Meek 2008; Kim et al. 2017; Ebesu and Fang 2017; Zhang et al. 2016a; Oramas et al. 2017; Bellini et al. 2017; Li and She 2017). Strub et al. (2016) integrate side information into each layer of the DAE. Gunawardana and Meek (2008) tie Boltzmann machines to content information of items to handle the cold-start problem. Kim et al. (2017) utilize CNN to extract features from images aiming to alleviate lack of tags in tag-aware recommender systems. Also, Ebesu and Fang (2017) propose to learn item representations from both implicit feedbacks of users and textual content information of items using deep neural networks. However, they do not consider context and order of texts. Another alternative for dealing with the cold-start problem is producing predictions based on users' current browser activities instead of historical activities. Tan et al. (2016) propose to use RNNs to generate predictions based on current user activities. To handle cold-start issues for a user in the current session, Ruocco et al. (2017) utilize RNN to integrate recent sessions of the user. Furthermore, Tuan and Phuong (2017) utilize 3D-CNNs to deal with sparsity problem in session-based recommender systems by integrating content information of items.

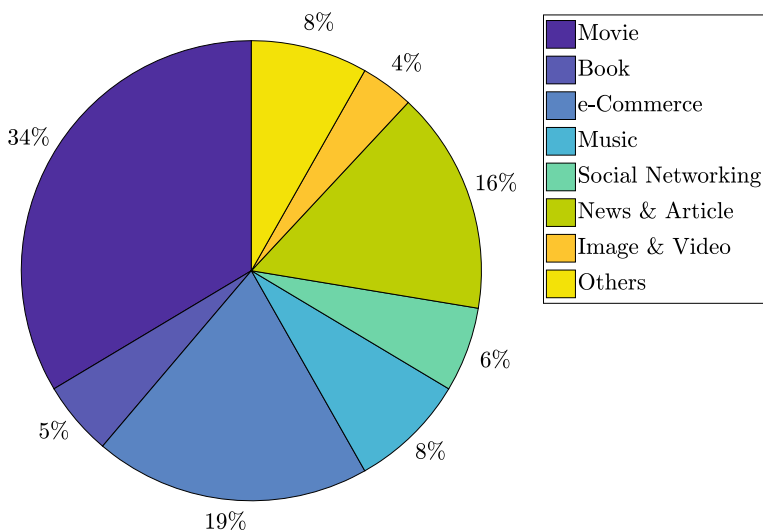
Chatzis et al. (2017) utilize Bayesian inference techniques with RNN to deal with the sparsity problem in session-based recommender systems. In music recommendation domain, Vall et al. (2017) propose a classifier based on neural networks for producing recommendations on sparse datasets. Volkovs et al. (2017) propose to apply dropout during network training to condition cold-start situation.

As it can be followed, the sparsity and cold-start problems are mostly handled by integrating extracted features from heterogeneous data sources into recommendation process by utilizing the power of deep learning techniques on feature engineering. Among various deep learning techniques, autoencoders, CNNs, and DBNs are the most frequently applied ones for such purpose.

#### 4.2.3 Solutions for scalability problem

Aiming to cope with scalability problem, researchers utilize deep learning techniques to extract low-dimensional latent factors of high-dimensional user preferences and item ratings (Salakhutdinov et al. 2007; Truyen et al. 2009; Georgiev and Nakov 2013; Elkahky et al. 2015). Truyen et al. (2009) utilize Boltzmann machines to extract latent features of items and users to produce predictions over large-scale datasets. Elkahky et al. (2015) set up multi-view deep neural networks to map high dimensional features into lower dimensional features to deal with scalability. Moreover, they apply some dimensionality reduction techniques such as selecting top- $k$  most relevant features, grouping similar features into the same cluster with the  $k$ -means algorithm, and local sensitive hashing to user features before training the network to scale up their system. Some researchers utilize RBM for dimensionality reduction purposes to handle large-scale datasets. Louppe (2010) proposes some methods such as parallel computing with shared memory, distributed computing, and method ensembles for modifying RBM-based CF to improve scalability. The author claims that parallel computing is more efficient than the other ones regarding the quality of recommendations.

Modifying parts of deep learning models to improve scalability is a preferred approach in recommender systems. Du et al. (2016) propose neural co-autoregressive model along both users and items where authors develop stochastic optimization algorithm to the scale-up learning process of their autoregressive model.



**Fig. 5** Distribution of recommendation domains

### 4.3 Awareness and prevalence over recommendation domains

The awareness and prevalence of deep learning-based models onto recommendation domains are critical for the researchers willing to work on the subject. In this section, we categorize existing publications according to their target recommendation domains of experimental studies. We contextualize closely related domains together and obtain eight main groups. **Figure 5** illustrates the distribution of publications experimenting on these recommendation domains. Note that a publication might be assigned to more than one category due to the variety of experiments carried on different recommendation domains.

#### 4.3.1 Movie recommendation

The movie recommendation domain is the basis of recommender systems research since there are many publicly available movie preference datasets of different volumes. Furthermore, the tabular structure of these datasets is well-suited for CF tasks.

The pioneer work in this field (Salakhutdinov et al. 2007) shows the potential of using RBMs for producing recommendations on Netflix dataset. The success of the study encourages many researchers to work with Boltzmann machines for recommendation purposes on the movie domain (Gunawardana and Meek 2008; Truyen et al. 2009; Louppe 2010). Ouyang et al. (2014) and Sedhain et al. (2015) introduce autoencoders to CF tasks, and they evaluate their models on MovieLens datasets. Similarly, Wang et al. (2015a) propose integrating auxiliary information with SDAEs and demonstrate empirical results obtained on MovieLens datasets. The autoregressive models proposed by Zheng et al. (2016b) and Du et al. (2016) are also evaluated within the movie domain.

### 4.3.2 Book recommendation

The book recommendation domain is closely related to movie recommendation domain since both of these end products have similar characteristics such as consuming period and content features. Furthermore, the movie industry considerably benefits from **written materials such as novels and comics**.

Shen et al. (2016) propose a method that can provide personalized book recommendations to students in an e-learning environment. With a similar purpose, Shu et al. (2018) also investigate improving recommendations of an e-learning system. Besides these studies, Li et al. (2015), Zhang et al. (2016a), Suglia et al. (2017), Hsieh et al. (2017) evaluate their models on both movie and book domains.

### 4.3.3 E-commerce

E-commerce is a famous domain of deep learning applications in recommender systems. In our domain analysis, the term “e-commerce” involves various items including hotels, restaurants, clothes, and many other commercial products.

Tang et al. (2015) propose a neural network model to predict restaurant review ratings. Wu et al. (2016a) and Hidasi et al. (2016a) analyze user sessions of e-commerce websites in a recurrent manner. In fashion recommendation, Wakita et al. (2016) utilize deep learning to discover favorite brands of users to improve the accuracy of recommended clothes, and Jaradat (2017) suggest utilizing the social connections between users in a cross-domain approach. On a large product shopping dataset containing item descriptions as image, text, and category, Nedelec et al. (2017) propose a new product representation architecture.

### 4.3.4 Music recommendation

With the recent advances, most people prefer to consume music digitally through online music services such as Spotify<sup>1</sup> and SoundCloud.<sup>2</sup> The evolution in digital music consumption has promoted music recommendation as a popular research field (Oord et al. 2013). Furthermore, the music domain is highly relevant for deep learning algorithms since the audio signals contain hidden patterns that can be transformed into useful information.

In music recommendation domain analysis, we classify the publications according to the utilization of data types and obtain three main categories, which are audio signals, content-based information, and ratings and usage data. While some researchers utilize deep learning to extract latent factors from audio signals (Oord et al. 2013; Wang and Wang 2014; Oramas et al. 2017), some others use deep models to make music recommendations from user ratings and usage data (Ko et al. 2016; Zuo et al. 2016; Kamehkhosh et al. 2017). Publications related to music recommendation domain are illustrated in Table 3, in which a checked cell indicates that corresponding publication utilizes corresponding data type.

### 4.3.5 Social networking recommendation

A social networking platform allows **users to stay in touch with their friends and meet new people with similar tastes**. According to the definition of social networking, one can

<sup>1</sup> <https://www.spotify.com/>.

<sup>2</sup> <https://soundcloud.com/>.

**Table 3** Publications in music domain based on utilized data types

Publication	Audio Signals	Content-based Info	Ratings and Usage
Oord et al. (2013)	✓		✓
Wang and Wang (2014)	✓	✓	✓
Ko et al. (2016)			✓
Zuo et al. (2016)		✓	✓
Kamehkhosh et al. (2017)			✓
Oramas et al. (2017)	✓	✓	✓
Barbieri et al. (2017)			✓
Ruocco et al. (2017)			✓
Xu et al. (2017b)		✓	

**Table 4** Publications of news, articles, and similar textual content recommendation

News	Articles and Blogs	Tags and citations
Kyo-Joong et al. (2014)	Shin et al. (2015)	Huang et al. (2015)
Baalen (2016)	Wang et al. (2015a)	Ebesu and Fang (2017)
Dai et al. (2017)	Wang et al. (2015b)	Nguyen et al. (2017)
Kumar et al. (2017)	Strub and Mary (2015)	
Ruocco et al. (2017)	Ying et al. (2016)	
Cao et al. (2017)	Li and She (2017)	
	Wang et al. (2017)	
	Hassan (2017)	

easily deduce that recommendation plays an important role in social networking services. User profiles, friends, followers, likes, comments, and tags constitute the terminology of recommendation in this domain.

Ko et al. (2016) apply a deep recurrent model to a location-based social network by using the check-in logs. Zuo et al. (2016) and Xu et al. (2017b) experiment with social bookmarking data in tag-aware recommender systems. Unger et al. (2016) and Yang et al. (2017) explore the contextual factors in **point-of-interest (POI) recommendation**.

#### 4.3.6 News and article recommendation

News and articles are usually large collections that are especially suitable for content-based recommendation. Besides news and articles, there are some other recommendable textual contents like blogs, tags, research papers, and citations.

While Kyo-Joong et al. (2014) classify the keywords of news and articles to capture user preferences, Ruocco et al. (2017) track user sessions to improve news recommendation quality. Wang et al. (2017) study dynamic deep models for editor or article recommendation task. Shin et al. (2015) propose a novel matrix factorization method for blog recommendation in Tumblr. Through a tag recommender system, Nguyen et al. (2017) use images to find appropriate tags related to each image. Mentioning some of the exciting works about news, articles, and similar content, the complete set of publications is presented in Table 4.

#### 4.3.7 Image and video recommendation

Images and videos are favorable recommendation items since they play an important role in entertainment, knowledge acquisition, education, and social networking. Lei et al. (2016) extract visual information from images for personalized image recommendation. Dominguez et al. (2017) present a model that is capable of predicting the next paintings that the users will probably purchase. Jia et al. (2015) emphasize the popularity of mobile devices and propose a video recommendation technique that utilizes textual descriptions of Android applications. Covington et al. (2016) split the video recommendation process of YouTube into two stages, which are candidate generation and ranking, and propose separate deep models for both stages.

#### 4.3.8 Other domains

Besides the domains described above, there exist substantial studies on health care, accommodation, advertising, software, and social security, which apply deep learning into recommender systems research. Zhao et al. (2015) propose a retrieval model for similar cases recommendation in the Internet inquiries for patients. Zhou et al. (2016) extract visual features from images to utilize visual user interest profiles in a hotel booking system. Another image-oriented recommendation method, proposed by Peska and Trojanova (2017), deals with photo lineups in the eyewitness identification process. Moreover, Zhang et al. (2016b) propose a deep architecture to predict user responses to online advertisements. In the software domain, Paisarnsrisomsuk (2015) propose a model to recommend moves of the Go game, Soh et al. (2017) provide users with adaptive user interfaces, and Bai et al. (2017) develop a framework to recommend long-tail web services.

The final observation of domain analysis reveals that multiple domains are brought together in the cross-domain recommendation. In a Microsoft research, Elkahky et al. (2015) combine company's different data sources related to search queries, application downloads, news browsing, and movie/TV views. Lian et al. (2017) propose a deep model that analyzes user feedback from movies, books, music, and activities.

### 4.4 Specialized recommender systems and deep learning

Specialized recommender systems allow representing relationships between users and items more precisely. In this section, we analyze such specialized recommender systems regarding deep learning-based techniques. Table 5 represents distributions of deep learning-based publications within specialized recommender systems where 'Others' category holds trust-aware and group-based recommender systems due to limited number of publications.

#### 4.4.1 Dynamic recommender systems

Personal preferences of users change over time. For example, while people love cartoons as children, they often do not prefer them as adults. Also, young people mostly enjoy sports news. However, they tend to prefer political news as they get older. Thus, the evolution of user preferences over time is an essential factor to be studied by recommender systems to be able to provide more appropriate recommendations. **Dynamic recommender systems deal with such temporal changes of user preferences.**

**Table 5** Publications on specialized recommender systems

Recommender system type	Publications	
Dynamic	Ko et al. (2016)	Wu et al. (2017a)
	Devooght and Bersini (2017)	Wei et al. (2017)
	Dai et al. (2017)	Wang et al. (2017)
	Kumar et al. (2017)	
Context-aware	Unger et al. (2016)	Seo et al. (2017a)
	Kim et al. (2017)	Paradarami et al. (2017)
	Wu et al. (2017b)	Yang et al. (2017)
Tag-aware	Zuo et al. (2016)	Shin et al. (2015)
	Xu et al. (2017b)	Zanotti et al. (2016)
	Strub et al. (2016)	Vall et al. (2017)
Session-based	Tan et al. (2016)	Ruocco et al. (2017)
	Hidasi et al. (2016a)	Chatzis et al. (2017)
	Wu et al. (2016a)	Hidasi et al. (2016b)
	Quadana et al. (2017)	
Cross-domain	Elkahky et al. (2015)	Jaradat (2017)
	Lian et al. (2017)	Zhao et al. (2018)
Others	Deng et al. (2017)	Yin et al. (2017)
	Pana et al. (2017)	Hu et al. (2014)

RNNs are utilized in recommender systems to consider such changes of user preferences over time (Ko et al. 2016; Devooght and Bersini 2017; Dai et al. 2017; Kumar et al. 2017; Wu et al. 2017a). Ko et al. (2016) propose a collaborative sequence model which represents contextual states of users in dynamic recommender systems. Authors utilize GRUs since they are more practical than LSTMs where they dynamically utilize orderings of events instead of the absolute time the events occur. Dai et al. (2017) use basic RNNs for nonlinearly representing item- and user-features and coevolution of these features at absolute times. Moreover, they integrate contextual information as interactive features. Devooght and Bersini (2017) utilize LSTMs to deal with changes in the interests of a user by interpreting CF as a sequence prediction problem. Kumar et al. (2017) utilize bidirectional LSTMs to distinguish between static and dynamic interests of users. Wu et al. (2017a) utilize an LSTM autoregressive model for non-linearly and non-parametrically demonstrating dynamics of both users and movies. Besides RNNs, researchers also utilize other deep learning techniques to capture dynamics of users. Wei et al. (2017) utilize SDAEs to extract features of items and combine these features with the timeSVD++ algorithm to alleviate sparsity problem in dynamic recommender systems. Wang et al. (2017) utilize a dynamic attention network architecture to reveal behavioral changes of editors in the middle ward and contextual changes concerning days of the week.

As heavily noticed, RNNs are extremely helpful in capturing evolving preferences over time where such kind of networks is directly utilized for producing dynamic recommendations. However, this type of recommender systems still needs significant improvements regarding accuracy. Other deep learning techniques such as LSTMs and especially GRUs seem to be promising in capturing changing preferences over short periods of time.

#### 4.4.2 Context-aware recommender systems

Context-aware recommender systems integrate contextual information such as time, location, and social status into recommendation process to improve quality of recommendations. For example, recommendations of clothes should be produced considering the season of the year, or hotels should be recommended in the context of business, pleasure, or both. Deep learning-based techniques are utilized in context-aware recommender systems to provide recommendations for the awareness of particular context.

Primarily, deep learning techniques are used for modeling various contexts. Unger et al. (2016) propose a latent context-aware recommendation system where they utilize autoencoders and PCA to extract latent contexts from raw data. After extracting them, these explicit contexts are integrated into matrix factorization process to produce recommendations. Kim et al. (2017) propose utilizing CNNs to capture contextual information from textual descriptions of items. Wu et al. (2017b) utilize dual CNNs for modeling textual information on both user and item sides by considering word order and surrounding words as contexts for more precisely representing users and items. Seo et al. (2017a) utilize CNNs with dual local and global attention layers to extract complex features capturing contexts from an aggregated user and item review texts. They represent complex features by two separate CNNs. Furthermore, Kim et al. (2016) utilize CNNs in order to extract latent contextual information from documents. Spatial dynamics of user preferences can be characterized with deep learning techniques. Yin et al. (2017) utilize DBNs to more precisely represent POIs from heterogeneous data sources for spatial-aware POI recommendations.

Also, deep learning approaches are used for producing predictions based on integrated context and preference information. Paradarami et al. (2017) utilize a gradient descent-based artificial neural network to generate recommendations based on context and collaborative features. Authors define collaborative features like ratings and votes of reviews which represent context and sentiment behind user behaviors. Yang et al. (2017) propose preference and context embedding through feedforward neural networks as a bridge between CF and semi-supervised learning where they produce predictions for users over the POIs.

As intensely noted, deep learning-based approaches are used in context-aware recommender systems with two purposes, i.e., (i) context modeling and (ii) producing predictions by both capturing preferences of users over items and contexts. Regarding context modeling, textual information is mostly utilized for capturing contextual information. Other contextual information such as social status and time of consumption should be modeled by deep learning techniques to provide more accurate recommendations. Regarding the accuracy of such recommenders, user- and item-latent factors capturing contextual information still needs to be extracted more precisely. Moreover, jointly representing the user and item features using only one neural network still is a need to improve performance. Studying impacts of other deep learning techniques on modeling and integrating context-awareness is an open research direction.

#### 4.4.3 Tag-aware recommender systems

In some e-commerce systems, users annotate products referred to as tags. These tags are sometimes utilized by recommender systems as side information to alleviate sparsity issues where they help profiling users, discovering hidden representations of users, and reveal semantic relationships among items. Tag-aware recommender systems are specialized in producing recommendations based on these tag information along with other conventional inputs.



Deep learning-based approaches are utilized in tag-aware recommender systems to help discover more hidden user representations and correlate items semantically based on tags. Zuo et al. (2016) utilize SDAE to extract out high-level features of users based on tags. Xu et al. (2017b) represent the user and item deep semantic models in two different autoencoders on user-tag and item-tag matrix. Authors optimize the deep model by training with hybrid learning for improving accuracy and utilize resulting user- and item-semantic models to produce predictions. Similarly, Strub et al. (2016) utilize SDAE for integrating tags as side information and non-linearly represent items and users. Authors utilize one network for integrating side information with the ratings. They integrate side information into each layer of the autoencoder and tag information is integrated as a matrix during implementation. Shallow neural network models are modeled for providing vector space representations of items and users by utilizing content and tag information (Shin et al. 2015; Zanotti et al. 2016; Vall et al. 2017).

Consequently, deep learning-based techniques, especially autoencoders, are used for extracting latent features of users and items over tags or integrating tags as side information to deal with sparsity. Other deep learning-based techniques can be applied to improve accuracy and reduce training time of the networks. Moreover, deep learning techniques are helpful in combining multiple sources of information. Thus, applying deep hybrid models on tag-aware recommender systems to integrate other side information, along with tags to deal with sparsity warrants future work.

#### 4.4.4 Session-based recommender systems

CF-based recommender systems rely on historical preference data of users to produce recommendations. However, lack of such preference data cripples the CF process resulting in the cold-start problem. As an alternative, session-based recommender systems rely on the recent behavior of users within the current session, which helps to handle cold-start users (Tan et al. 2016).

RNNs are used in session-based recommender systems to estimate the next event in a session for a user. Hidasi et al. (2016a) utilize a GRU-based RNN to predict the next event in a session. Although authors fit the proposed network into recommender systems by introducing ranking loss function, sampling the output and parallel mini-batches, they utilize only session clicks. Alternatively, Tan et al. (2016) process each sequence separately instead of parallel-mini batches. To improve accuracy, authors apply data augmentation to deal with noisy clicks. They utilize a threshold value to deal with temporal changes in user behaviors over time and propose a novel item embedding model to reduce time and space constraints of the RNN-based model. Wu et al. (2016a) also utilize RNN to predict the next event in session-based recommender systems. They restrict the number of history states to improve the model training time and combine the RNN and feedforward neural network models to integrate user-item interactions. Ruocco et al. (2017) utilize an additional RNN to alleviate the problem of producing recommendations at the start of a session. Moreover, Chatzis et al. (2017) propose an inferential framework over session-based recommender systems based on RNN to alleviate the sparsity issues. Quadrana et al. (2017) propose a session-aware recommendation approach by incorporating past session information of users in their current sessions by utilizing RNNs.

Deep learning-based approaches are used in session-based recommender systems to deal with both session clicks and content information. Hidasi et al. (2016b) utilize RNNs to integrate content information of clicked items into session-based recommender systems.

However, they use a separate RNN architecture for each type of content information. Alternatively, Tuan and Phuong (2017) utilize deep 3D-CNN to combine sequential pattern of session clicks with item content features. The content information used in the model contains *id*, *name*, and *category* of products, and more content information, such as timestamp can be utilized, as well.

Among other deep learning techniques, RNNs are effectively used in session-based recommender systems to integrate session clicks and content information into recommendation process. However, existing studies should be improved regarding processing time, accuracy, and novelty. In addition to session click information, other content information such as item descriptions and user comments would be integrated into session-based recommender systems via deep learning-based approaches to improve accuracy without jeopardizing processing time.

#### 4.4.5 Cross-domain recommender systems

Cross-domain recommender systems aim to deal with sparsity issues by combining feature of users and items from different domains (Lian et al. 2017).

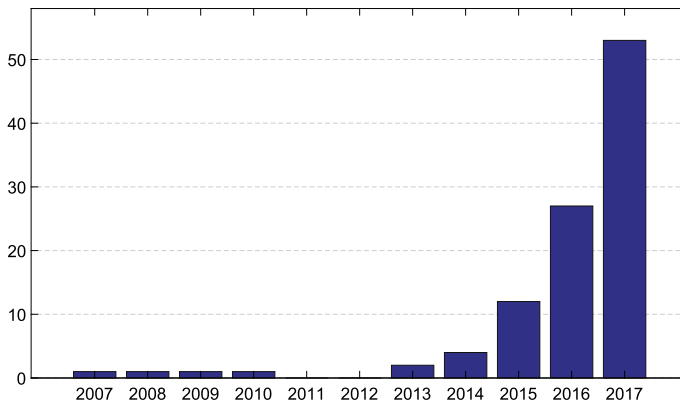
Elkahky et al. (2015) modify the deep-structured semantic model to map user and item features extracted from different domains into the same semantic space for content-based recommender systems. Lian et al. (2017) utilize multi-view neural networks for different domains for hybrid recommender systems. Jaradat (2017) utilize CNNs both extracting features from images and transferring knowledge between domains, and Zhao et al. (2018) utilize RNN and CNN together to represent movies in terms of their textual synopsis and poster images.

Also, hybrid deep learning techniques can be used to represent users and items more precisely. Besides multi-view neural networks, deep neural networks can be used for cross-domain recommender systems.

#### 4.4.6 Other techniques

Users naturally trust their friends more than strangers. Trustworthiness of user ratings is essential for providing more dependable recommendations. Trust-aware recommender systems consider the trustworthiness of users' ratings while producing predictions. Therefore, Deng et al. (2017) utilize autoencoders to improve original item- and user-latent factors of matrix factorization by applying non-linear dimensionality reduction for trust-aware recommender system. In order to compute trust degrees, they utilize similarities and friendship between an active user and the remaining users. They propose *n*-Trust-Clique community structure to identify communities considering trust relationship in social networks. Pana et al. (2017) utilize DAEs to extract user preferences from both ratings and trust information. Their network model utilizes two autoencoders for each type of data and ties them through a weighted layer. Time sensitivity can be integrated into trust-aware recommender systems using deep learning-based approaches to provide more accurate recommendations. Other deep learning-based techniques such as NADE can be used to extract trust degrees between users and their friends and integrating trust and rating information into a single recommendation model.

Rather than recommending an item to a single user, recommending it to a group of users considering group preferences is essential in daily life. In that direction, Hu et al. (2014) utilize collective DBNs to extract high-level features of a group considering each member's



**Fig. 6** Number of publications over years

features. Then, they utilize a dual-wing RBM to produce group preferences from individual and collective features. Sparsity and cold-start problems can be handled by integrating side information to group-based recommendation model with deep learning techniques. Moreover, other deep learning techniques can be used to represent common features.

## 5 Quantitative assessment of comprehensive literature

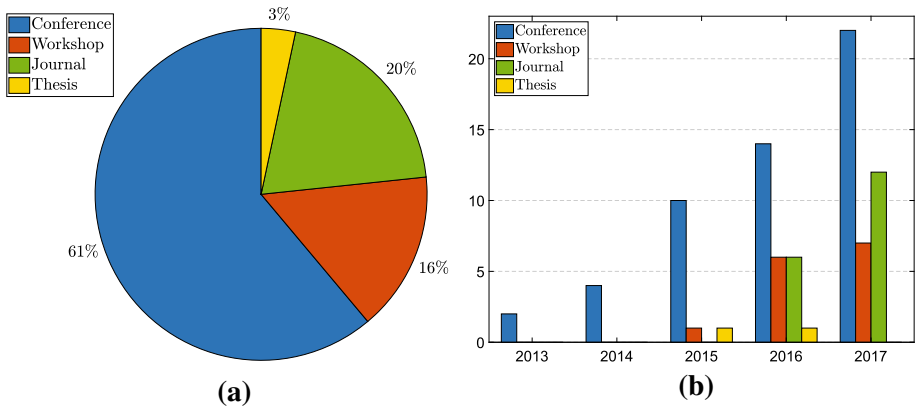
In this section, we provide a quantitative assessment of the comprehensive literature that analyzes publication with respect to their publishing time, type of publication, and utilized datasets.

Firstly, we present the number of publications over the years in Fig. 6. Since the first remarkable work that was conducted by Salakhutdinov et al. (2007), there is an increasing trend in applying deep learning techniques to recommender systems. Since then, the number of studies proposed in the field has been on the rise. As a vital sign of the emergence of the field, note that 27 research papers are published in 2016, whereas there are more than 50 studies in 2017 covered by this survey.

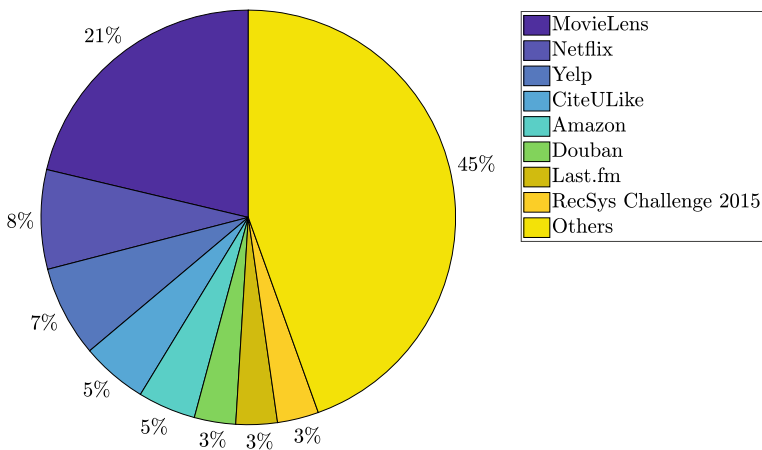
The increase in the number of studies bringing deep learning and recommender systems together can be related to the overall popularity and effectiveness of deep learning within computer science. Regarding recommender systems field, deep models are very successful at learning from different sources and extracting hidden features among recommendation participants. By considering the advances in big data processing capabilities and interpreting the current trend in applying deep models to recommender systems, one can assert that the collaboration of the two fields will continue to gain popularity in the near future, as well.

Secondly, we examine the studies according to their publishing types. Although there are some electronic preprints on arXiv<sup>3</sup>, we exclude those studies from the analysis and obtain four major publication types, namely, conference papers, journal papers, workshop papers, and theses and dissertations. Figure 7 illustrates the overall distribution and temporal change of publication types over the years. Since more than half of the studies are published in conferences, the dominant publishing type among the papers is such type. However, workshop and journal papers are also on the rise as it can be followed by the trend over the last 5 years.

<sup>3</sup> <https://arxiv.org/>.



**Fig. 7** Distribution of publications by their types. **a** Overall distribution and **b** temporal change



**Fig. 8** Distribution of datasets

The third assessment covers the datasets used in the studies. We investigate the experiments in each paper and present the distribution of utilized datasets as a pie chart in Fig. 8.

Although some of the researchers use custom and private datasets, most of the experiments are carried on public and common datasets. Among all, 44% of the datasets are categorized as “Others”, and such category contains the datasets that are used in only one of the studies. Keeping this critical information in mind, one can observe that MovieLens, Netflix, and Yelp datasets are the most commonly preferred ones in the experiments as 21% of the studies use MovieLens, 8% use Netflix, and 7% use Yelp datasets. The properties of MovieLens and Netflix datasets are also presented in Table 6.

## 6 Insights and discussions

As we discussed in detail how deep learning techniques prevail into many fields in computer science, these methods also demonstrate efficient and proficient performance in recommender

**Table 6** Properties of commonly used datasets

Name	Domain	Users	Items	Range	Total Ratings	Density (%)
MovieLens 100 K	Movie	943	1682	5-star	100,000	6.30
MovieLens 1 M	Movie	6040	3952	5-star	1,000,209	4.19
MovieLens 10 M	Movie	71,567	10,681	5-star	10,000,054	1.31
Netflix	Movie	408,189	17,770	5-star	100,480,507	1.39

systems field. Improving computing capabilities, collection and efficient processing of big data, integration of multiple data sources are prominent causes of the contemporary dominance of deep learning applications.

In this section, we discuss our collection of findings and inferences and present the reader with insights based on the overall analysis on deep learning-based recommender systems throughout the study.

- (i) Deep learning techniques are not specialized onto a unique recommendation method; they are utilized in all kinds of recommendation methodologies with different purposes. In content-based filtering, these techniques are mostly used for extracting features to generate content-based user/item profiles from heterogeneous data sources. However, in CF, they are often utilized as a model-based approach to extracting latent factors on user-item matrix. In hybrid recommender systems, deep learning methods are utilized for extracting features from auxiliary information and integrating them into the recommendation process.
- (ii) Among all deep learning techniques, RBMs, autoencoders, and NADE are the most used ones in latent factor analysis. Although existing studies show that NADE precedes RBM- and autoencoder-based recommender systems regarding accuracy, autoencoders are more popular in recommender systems research. Comparatively, autoencoders are more popular in recommender systems field due to their straightforward structure, suitability for feature engineering, dimensionality reduction, and missing value estimation capabilities.
- (iii) RNNs are mostly used for session-based recommendations to improve the accuracy by integrating current user history to their preferences. Moreover, RNNs are preferred in recommender systems to take evolving tastes of users over time into account.
- (iv) CNNs and DBNs are mostly used for feature engineering from the text, audio, and image inputs. The extracted features are used in either content-based filtering techniques or as side information in CF.
- (v) Deep learning techniques are utilized to deal with the sparsity and cold-start problems of recommender systems by extracting features from side information and integrating them into user-item preferences. Moreover, deep learning-based approaches are used to reduce dimensions of high-level and sparse features into low-level and denser features.
- (vi) Existing literature demonstrates that deep learning-based approaches provide more accurate recommendations than traditional recommendation algorithms such as matrix factorization- and nearest neighbor-based approaches. The main reason for such situation is that deep learning algorithms offer non-linear representations of user preferences which enable discovering unexpected or incomprehensible behavior.
- (vii) Deep learning-based approaches are utilized in context-aware recommender systems to model contextual data or capturing both user preferences and contexts.

- (viii) Deep learning techniques are utilized in tag-aware recommender systems to integrate tags as side information or to model user-item preferences based on tags.
- (ix) Deep learning-based approaches are also used in recommender systems with the purpose of handling large-scale datasets by reducing dimensionality. In addition to dimensionality reduction, existing deep learning techniques can be adopted into a more scalable form to deal with large-scale data.
- (x) Existing studies conclude that deeper forms of deep learning models yield more accurate recommendations compared to shallow ones. Moreover, applying deep learning methods along with both user-based and item-based approaches provides recommendations with higher accuracy, as well.
- (xi) Since most of the deep learning-based recommendation approaches focus on improving accuracy of either rating prediction or item ranking prediction, the evaluation metrics typically used in recommender systems for measuring statistical accuracy are utilized in deep learning-based techniques, as well. Typical measures for evaluating deep learning-based recommendation methods are mean-squared error (MSE), mean absolute error (MAE), and RMSE for prediction accuracy; precision, recall, F1-measure, and receiver operating characteristic (ROC) curve for classification accuracy; and normalized discounted cumulative gain (nDCG) for ranking recommended items lists.
- (xii) Mostly utilized evaluation methods for experimental scrutinization of deep learning-based approaches are similar with machine learning field and include k-fold cross validation and randomly splitting input data into train and test sets with reservation of a validation set to avoid over-fitting.

Additionally, we would like to direct the reader to open research subjects that warrant future works on deep learning in recommender systems area. We list noticed inadequacies that possible researchers might focus.

- (i) Modifying deep learning models to make them more scalable is still a challenge in recommender systems. Moreover, there is still a need for solutions on optimizing exponential growth of parameters when the size of data grows.
- (ii) In order to improve accuracy and handling sparsity issues in recommender systems, additional information needs to be integrated into user-item preferences. For e-commerce, decent recommendations can be produced by utilizing both contextual data, tags, and evolution of user preferences in addition to the user-item preference matrix. Thus, deep learning approaches can be used in a more composite manner to extract useful information from many different data sources and integrate them into the recommendation generation processes.
- (iii) Deep learning-based approaches are utilized in pattern recognition for extracting features from audio signals, videos, images, and texts. Unfortunately, there is a limited number of studies which utilize abilities of deep learning techniques for extracting low-level features, especially in the image, video, and music (song) recommendations.
- (iv) The evolution and co-evolution of users and items are significant temporal dynamics in recommender systems. Thus, deep sequential networks can be used to improve dynamic recommender systems.
- (v) Since deep learning-based approaches provide more qualified recommendations due to their non-linear representation of data abilities; they can also be utilized for improving other success criteria of recommender systems such as serendipity, novelty, diversity, and coverage of produced recommendations.
- (vi) Specialized recommender systems allow alleviating the sparsity problem and providing more accurate recommendations. However, there are a limited number of studies

in this area, especially for cross-domain, spatial-aware, group-based, and trust-aware recommender systems.

- (vii) Deep learning techniques are applied on numerical preference data in recommender systems. Since deep learning is originally utilized in classification problems, using such models might be used for binary preferences-based recommender systems.
- (viii) Deep learning-based approaches achieve high accuracy in produced recommendations when applied on non-privately collected preferences. Utilizing deep learning techniques on privately collected data to provide a balance between conflicting goals of accuracy and privacy levels in recommender systems is a challenge. Furthermore, utilizing such techniques on either centrally stored or distributed private collections is another challenge.
- (ix) Deep learning is successful in extracting features. However, it has not been utilized in identifying nearest neighbors of active users in memory-based CF algorithms which can improve accuracy further. Deep learning techniques to identify neighbors of an active user to improve accuracy is a challenge.

## 7 Conclusion

Deep learning has become more and more popular throughout all subfields of computer science, such as natural language processing, image and video processing, computer vision, and data mining, which is a remarkable phenomenon since there has not been such a common approach to be used in solving different kinds of computing problems before. With such aspect of deep learning techniques, they are not only highly capable of remedying complex problems in many fields, but they also form a shared vocabulary and common ground for these research fields. Deep learning methods even help these subfields to collaborate with each other where it was a bit problematical in the past due to the diversity and complexity of utilized techniques.

This survey study aims to review existing complete literature on deep learning-based recommender system approaches to help new researchers build a comprehensive understanding of the field. Mainly, we classify current literature in four primary aspects which we believe help the reader constitute a holistic comprehension. Firstly, we investigate the literature based on varying models of deep learning and categorize studies according to utilized deep models which aim to understand how deep learning techniques are fit to the recommendation generation problem. Secondly, we examine the literature with the perspective of how deep learning methods remedy current challenges of recommender systems research and how much they are successful at achieving such goal. Moreover, we aim to direct the reader to contemplate on how to provide deep learning-based solutions to existing problems and what warrants future work in the field. Then, we evaluate the application domains where deep learning-based recommender systems focus on and provide a categorization of the major domains. Finally, we analyze the relation between deep learning techniques and purposive properties of recommender systems and categorize the publications into the corresponding recommender system type. We also provide a general classification of publications where we present a quantitative assessment of the literature and discuss gained insights on the subject.

As long as the personalization trend remains popular, the recommender systems research will play an essential role in information filtering. Although the application of deep learning into recommender systems field promises significant and encouraging results, challenges such as the accuracy and scalability are still open for improvements and warrant future work.



## References

- Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 17(6):734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Aggarwal CC (ed) (2016) An introduction to recommender systems. In: *Recommender systems*, 1st edn. Springer, Cham, pp 1–28
- Baalen MV (2016) Deep matrix factorization for recommendation. Master's thesis, University of Amsterdam
- Bai B, Fan Y, Tan W, Zhang J (2017) Dltsr: a deep learning framework for recommendation of long-tail web services. *IEEE Trans Serv Comput*. <https://doi.org/10.1109/TSC.2017.2681666>
- Barbieri J, Alvim LGM, Braidã F, Zimbrão G (2017) Autoencoders and recommender systems: cofils approach. *Expert Syst Appl* 89:81–90. <https://doi.org/10.1016/j.eswa.2017.07.030>
- Bedi P, Kaur H, Marwaha S (2007) Trust based recommender system for semantic web. In: *Proceedings of the 20th international joint conference on artificial intelligence*, Hyderabad, India, vol 7, pp 2677–2682
- Bellini V, Anelli VW, Di Noia T, Di Sciascio E (2017) Auto-encoding user ratings via knowledge graphs in recommendation scenarios. In: *Proceedings of the 2nd workshop on deep learning for recommender systems*, Como, Italy, pp 60–66
- Bengio Y (2009) Learning deep architectures for ai. *Found Trends Mach Learn* 2(1):1–127. <https://doi.org/10.1561/22000000006>
- Betru BT, Onana CA, Batchakui B (2017) Deep learning methods on recommender system: a survey of state-of-the-art. *Int J Comput Appl* 162(10):17–22. <https://doi.org/10.5120/ijca2017913361>
- Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowl Based Syst* 46:109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>
- Breese JS, Heckerman D, Kadie CM (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the 14th conference on uncertainty in artificial intelligence*, Madison, Wisconsin, USA, pp 43–52
- Burke R (2002) Hybrid recommender systems: survey and experiments. *User Model User Adapt Interact* 12(4):331–370. <https://doi.org/10.1023/A:1021240730564>
- Cao S, Yang N, Liu Z (2017) Online news recommender based on stacked auto-encoder. In: *Proceedings of the 16th IEEE/ACIS international conference on computer and information science*, Wuhan, China, pp 721–726
- Chatzis SP, Christodoulou P, Andreou AS (2017) Recurrent latent variable networks for session-based recommendation. In: *Proceedings of the 2nd workshop on deep learning for recommender systems*, Como, Italy, pp 38–45
- Cheng HT, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, Anderson G, Corrado G, Chai W, Isipir M, Anil R, Haque Z, Hong L, Jain V, Liu X, Shah H (2016) Wide & deep learning for recommender systems. In: *Proceedings of the 1st workshop on deep learning for recommender systems*, Boston, MA, USA, pp 7–10
- Cireşan DC, Meier U, Gambardella LM, Schmidhuber J (2010) Deep, big, simple neural nets for handwritten digit recognition. *Neural Comput* 22(12):3207–3220. [https://doi.org/10.1162/NECO\\_a\\_00052](https://doi.org/10.1162/NECO_a_00052)
- Covington P, Adams J, Sargin E (2016) Deep neural networks for youtube recommendations. In: *Proceedings of the 10th ACM conference on recommender systems*, Boston, MA, USA, pp 191–198
- Dai H, Wang Y, Trivedi R, Song L (2017) Deep coevolutionary network: embedding user and item features for recommendation. [arXiv:1609.03675](https://arxiv.org/abs/1609.03675)
- Deldjoo Y, Quadana M, Elahi M, Cremonesi P (2017) Using mise-en-scène visual features based on mpeg-7 and deep learning for movie recommendation. [arXiv:1704.06109](https://arxiv.org/abs/1704.06109)
- Deng L, Yu D (2014) Deep learning: methods and applications. *Found Trends Signal Process* 7(3–4):197–387. <https://doi.org/10.1561/20000000039>
- Deng S, Huang L, Xu G, Wu X, Wu Z (2017) On deep learning for trust-aware recommendations in social networks. *IEEE Trans Neural Netw Learn Syst* 28(5):1164–1177. <https://doi.org/10.1109/TNNLS.2016.2514368>
- Devooght R, Bersini H (2017) Long and short-term recommendations with recurrent neural networks. In: *Proceedings of the 25th conference on user modeling, adaptation and personalization*, Bratislava, Slovakia, pp 13–21
- Dominguez V, Messina P, Parra D, Mery D, Trattner C, Soto A (2017) Comparing neural and attractiveness-based visual features for artwork recommendation. In: *Proceedings of the 2nd workshop on deep learning for recommender systems*, Como, Italy, pp 55–59
- Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: *Proceedings of the 28th IEEE conference on computer vision and pattern recognition*, Boston, MA, USA, pp 2625–2634



- Du C, Li C, Zheng Y, Zhu J, Liu C, Zhou H, Zhang B (2016) Collaborative filtering with user-item co-autoregressive models. [arxiv:1612.07146](https://arxiv.org/abs/1612.07146)
- Du Yp, Yao Cq, Huo Sh, Liu Jx (2017) A new item-based deep network structure using a restricted Boltzmann machine for collaborative filtering. *Front Inf Technol Electron Eng* 18(5):658–666. <https://doi.org/10.1631/FITEE.1601732>
- Ebesu T, Fang Y (2017) Neural semantic personalized ranking for item cold-start recommendation. *Inf Retr J* 20(2):109–131. <https://doi.org/10.1007/s10791-017-9295-9>
- Elkahky AM, Song Y, He X (2015) A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: *Proceedings of the 24th international conference on world wide web*, Florence, Italy, pp 278–288
- Georgiev K, Nakov P (2013) A non-iid framework for collaborative filtering with restricted Boltzmann machines. In: *Proceedings of the 30th international conference on machine learning*, pp III–1148–III–1156
- Gunawardana A, Meek C (2008) Tied Boltzmann machines for cold start recommendations. In: *Proceedings of the 2nd ACM conference on recommender systems*, Lausanne, Switzerland, pp 19–26
- Hassan HAM (2017) Personalized research paper recommendation using deep learning. In: *Proceedings of the 25th conference on user modeling, adaptation and personalization*, pp 327–330
- He J, Zhuo HH, Law J (2017) Distributed-representation based hybrid recommender system with short item descriptions. [arxiv:1703.04854](https://arxiv.org/abs/1703.04854)
- Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2016a) Session-based recommendations with recurrent neural networks. In: *Proceedings of the 4th international conference on learning representations*, San Juan, Puerto Rico
- Hidasi B, Quadrana M, Karatzoglou A, Tikk D (2016b) Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: *Proceedings of the 10th ACM conference on recommender systems*, Boston, MA, USA, pp 241–248
- Hinton GE (2009) Deep belief networks. *Scholarpedia* 4(5):5947. <https://doi.org/10.4249/scholarpedia.5947>
- Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507. <https://doi.org/10.1126/science.1127647>
- Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hsieh CK, Yang L, Cui Y, Lin TY, Belongie S, Estrin D (2017) Collaborative metric learning. In: *Proceedings of the 26th international conference on world wide web*, pp 193–201
- Hu L, Cao J, Xu G, Cao L, Gu Z, Cao W (2014) Deep modeling of group preferences for group-based recommendation. In: *Proceedings of the 28th AAAI conference on artificial intelligence*, Québec City, Québec, Canada, pp 1861–1867
- Huang W, Wu Z, Liang C, Mitra P, Giles CL (2015) A neural probabilistic model for context based citation recommendation. In: *Proceedings of the 29th AAAI conference on artificial intelligence*, Austin, Texas, USA, pp 2404–2410
- Jaradat S (2017) Deep cross-domain fashion recommendation. In: *Proceedings of the 11th ACM conference on recommender systems*, Como, Italy, pp 407–410
- Jia X, Wang A, Li X, Xun G, Xu W, Zhang A (2015) Multi-modal learning for video recommendation based on mobile application usage. In: *Proceedings of IEEE international conference on big data*, Santa Clara, CA, USA, pp 837–842
- Kamehkhosh I, Jannach D, Ludewig M (2017) A comparison of frequent pattern techniques and a deep learning method for session-based recommendation. In: *Proceedings of the 1st workshop on temporal reasoning in recommender systems*, Como, Italy, pp 50–56
- Kim D, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document context-aware recommendation. In: *Proceedings of the 10th ACM conference on recommender systems*, Boston, MA, USA, pp 233–240
- Kim D, Park C, Oh J, Yu H (2017) Deep hybrid recommender systems via exploiting document context and statistics of items. *Inf Sci* 417:72–87. <https://doi.org/10.1016/j.ins.2017.06.026>
- Ko YJ, Maystre L, Grossglauser M (2016) Collaborative recurrent neural networks for dynamic recommender systems. In: *Proceedings of the 8th Asian conference on machine learning*, Hamilton, New Zealand, vol 63, pp 366–381
- Kumar V, Khattar D, Gupta S, Gupta M, Varma V (2017) Deep neural architecture for news recommendation. In: *Working notes of CLEF 2017 conference and labs of the evaluation forum*, Dublin, Ireland

- Kyo-Joong O, Won-Jo L, Chae-Gyun L, Choi HJ (2014) Personalized news recommendation using classified keywords to capture user preference. In: Proceedings of the 16th international conference on advanced communication technology, Phoenix Park, PyeongChang Korea(south), pp 1283–1287
- Larochelle H, Murray I (2011) The neural autoregressive distribution estimator. In: Proceedings of the 14th international conference on artificial intelligence and statistics, Fort Lauderdale, FL, USA, pp 29–37
- Lei C, Liu D, Li W, Zha ZJ, Li H (2016) Comparative deep learning of hybrid representations for image recommendations. In: Proceedings of the 29th IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, pp 2545–2553
- Li X, She J (2017) Collaborative variational autoencoder for recommender systems. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax, NS, Canada, pp 305–314
- Li S, Kawale J, Fu Y (2015) Deep collaborative filtering via marginalized denoising auto-encoder. In: Proceedings of the 24th ACM international conference on information and knowledge management, Melbourne, VIC, Australia, pp 811–820
- Lian J, Zhang F, Xie X, Sun G (2017) Cccfnet: a content-boosted collaborative filtering neural network for cross domain recommender systems. In: Proceedings of the 26th international conference on world wide web companion, Perth, Australia, pp 817–818
- Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput* 7(1):76–80. <https://doi.org/10.1109/MIC.2003.1167344>
- Liu J, Wu C (2017) Deep learning based recommendation: a survey. In: Proceedings of the 2017 international conference on information science and applications, Tel-Aviv, Israel, pp 451–458
- Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) *Recommender systems handbook*. Springer, Boston, pp 73–105
- Louppe G (2010) Collaborative filtering: Scalable approaches using restricted Boltzmann machines. Master's thesis, University of Liège
- Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the 4th ACM international conference on web search and web data mining, Hong Kong, China, pp 287–296
- McCarthy K, Salamó M, Coyle L, McGinty L, Smyth B, Nixon P (2006) Group recommender systems: a critiquing based approach. In: Proceedings of the 11th international conference on intelligent user interfaces, Sydney, Australia, pp 267–269
- Nedelec T, Smirnova E, Vasile F (2017) Specializing joint representations for the task of product recommendation. In: Proceedings of the 2nd workshop on deep learning for recommender systems, Como, Italy, pp 10–18
- Nguyen HTH, Wistuba M, Grabocka J, Drumond LR, Schmidt-Thieme L (2017) Personalized deep learning for tag recommendation. In: Proceedings of the Pacific-Asia conference on knowledge discovery and data mining, Jeju, South Korea, pp 186–197
- Obadić I, Madjarov G, Dimitrovski I, Gjorgjevikj D (2017) Addressing item-cold start problem in recommendation systems using model based approach and deep learning. In: Proceedings of the 9th international conference on ICT innovations, Skopje, Macedonia, pp 176–185
- Oord Avd, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. In: Proceedings of the 26th international conference on neural information processing systems, Lake Tahoe, NV, USA, pp 2643–2651
- Oramas S, Nieto O, Sordo M, Serra X (2017) A deep multimodal approach for cold-start music recommendation. [arxiv:1706.09739](https://arxiv.org/abs/1706.09739)
- Ouyang Y, Liu W, Rong W, Xiong Z (2014) Autoencoder-based collaborative filtering. In: Proceedings of the 21st international conference on neural information processing, Kuching, Malaysia
- Paisarnsrisomsuk S (2015) Uct-enhanced deep convolutional neural network for movie recommendation in go. Ph.D. thesis, Worcester Polytechnic Institute
- Pana Y, Hea F, Yua H (2017) Trust-aware collaborative denoising auto-encoder for top-n recommendation. [arxiv:1703.01760](https://arxiv.org/abs/1703.01760)
- Paradarami TK, Bastian ND, Wightman JL (2017) A hybrid recommender system using artificial neural networks. *Expert Syst Appl* 83:300–313. <https://doi.org/10.1016/j.eswa.2017.04.046>
- Peska L, Trojanova H (2017) Towards recommender systems for police photo lineup. In: Proceedings of the 2nd workshop on deep learning for recommender systems, Como, Italy, pp 19–23
- Quadrana M, Karatzoglou A, Hidasi B, Cremonesi P (2017) Personalizing session-based recommendations with hierarchical recurrent neural networks. In: Proceedings of the 11th ACM conference on recommender systems, Como, Italy, pp 130–137

- Rassweiler Filho RJ, Wehrmann J, Barros RC (2017) Leveraging deep visual features for content-based movie recommender systems. In: Proceedings of the 2017 international joint conference on neural networks, Anchorage, AK, USA, pp 604–611
- Ruocco M, Skrede OSL, Langseth H (2017) Inter-session modeling for session-based recommendation. In: Proceedings of the 2nd workshop on deep learning for recommender systems, Como, Italy, pp 24–31
- Salakhutdinov R, Hinton GE (2009) Deep Boltzmann machines. In: Proceedings of the 12th international conference on artificial intelligence and statistics, Clearwater Beach, Florida, USA, vol 1, pp 448–455
- Salakhutdinov R, Mnih A, Hinton GE (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on machine learning, Corvallis, Oregon, USA, pp 791–798
- Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on world wide web, Hong Kong, China, pp 285–295
- Schafer JB, Konstan JA, Riedl J (2001) E-commerce recommendation applications. *Data Min Knowl Discov* 5(1/2):115–153. <https://doi.org/10.1023/A:1009804230409>
- Sedhain S, Menon AK, Sanner S, Xie L (2015) Autorec: autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on world wide web, Florence, Italy, pp 111–112
- Seo S, Huang J, Yang H, Liu Y (2017a) Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of the 11th ACM conference on recommender systems, Como, Italy, pp 297–305
- Seo S, Huang J, Yang H, Liu Y (2017b) Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In: Proceedings of the 3rd international workshop on machine learning methods for recommender systems, Houston, TX, USA
- Shani G, Gunawardana A (2011) Evaluating recommendation systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) *Recommender systems handbook*. Springer, Boston, pp 257–297
- Shankar D, Narumanchi S, Ananya HA, Kompalli P, Chaudhury K (2017) Deep learning based large scale visual recommendation and search for e-commerce. [arxiv:1703.02344](https://arxiv.org/abs/1703.02344)
- Shen X, Yi B, Zhang Z, Shu J, Liu H (2016) Automatic recommendation technology for learning resources with convolutional neural network. In: Proceedings of the international symposium on educational technology, Beijing, China, pp 30–34
- Shin D, Cetintas S, Lee KC, Dhillon IS (2015) Tumblr blog recommendation with boosted inductive matrix completion. In: Proceedings of the 24th ACM international conference on information and knowledge management, Melbourne, VIC, Australia, pp 203–212
- Shu J, Shen X, Liu H, Yi B, Zhang Z (2018) A content-based recommendation algorithm for learning resources. *Multimed Syst* 24(2):163–173. <https://doi.org/10.1007/s00530-017-0539-8>
- Singh AP, Gordon GJ (2010) A Bayesian matrix factorization model for relational data. In: Proceedings of the 26th conference on uncertainty in artificial intelligence, Catalina Island, CA, USA, pp 556–563
- Soh H, Sanner S, White M, Jamieson G (2017) Deep sequential recommendation for personalized adaptive user interfaces. In: Proceedings of the 22nd international conference on intelligent user interfaces, pp 589–593
- Strub F, Mary J (2015) Collaborative filtering with stacked denoising autoencoders and sparse inputs. In: Proceedings of the NIPS workshop on machine learning for eCommerce, Montreal, Canada
- Strub F, Gaudel R, Mary J (2016) Hybrid recommender system based on autoencoders. In: Proceedings of the 1st workshop on deep learning for recommender systems, Boston, MA, USA, pp 11–16
- Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. *Adv Artif Intell* 2009:21,425:1–421,425:19. <https://doi.org/10.1155/2009/421425>
- Suglia A, Greco C, Musto C, de Gemmis M, Lops P, Semeraro G (2017) A deep architecture for content-based recommendations exploiting recurrent neural networks. In: Proceedings of the 25th conference on user modeling, adaptation and personalization, pp 202–211
- Suzuki Y, Ozaki T (2017) Stacked denoising autoencoder-based deep collaborative filtering using the change of similarity. In: Proceedings of the 31st international conference on advanced information networking and applications workshops, pp 498–502
- Tan YK, Xu X, Liu Y (2016) Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st workshop on deep learning for recommender systems, Boston, MA, USA, pp 17–22
- Tang D, Qin B, Liu T, Yang Y (2015) User modeling with neural network for review rating prediction. In: Proceedings of the 24th international joint conference on artificial intelligence, Buenos Aires, Argentina, pp 1340–1346
- Tran T, Cohen R (2000) Hybrid recommender systems for electronic commerce. In: Proceedings of the AAAI workshop on knowledge-based electronic markets, Austin, TX, USA, vol 4

- Truyen TT, Phung DQ, Venkatesh S (2009) Ordinal Boltzmann machines for collaborative filtering. In: Proceedings of the 25th conference on uncertainty in artificial intelligence, pp 548–556
- Tso-Sutter KHL, Marinho LB, Schmidt-Thieme L (2008) Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: Proceedings of the 2008 ACM symposium on applied computing, pp 1995–1999
- Tuan TX, Phuom TM (2017) 3D convolutional networks for session-based recommendation with content features. In: Proceedings of the 11th ACM conference on recommender systems, pp 138–146
- Unger M, Bar A, Shapira B, Rokach L (2016) Towards latent context-aware recommendation systems. *Knowl Based Syst* 104(C):165–178. <https://doi.org/10.1016/j.knosys.2016.04.020>
- Vall A, Eghbal-zadeh H, Dorfer M, Schedl M, Widmer G (2017) Music playlist continuation by learning from hand-curated examples and song features: alleviating the cold-start problem for rare and out-of-set songs. In: Proceedings of the 2nd workshop on deep learning for recommender systems, Como, Italy, pp 46–54
- Van Meteren R, Van Someren M (2000) Using content-based filtering for recommendation. In: Proceedings of the workshop on machine learning in the new information age, Barcelona, Spain, pp 47–56
- Verbert K, Duval E, Lindstaedt S, Gillet D (2010) Context-aware recommender systems. *J Univ Comput Sci* 16(16):2175–2178
- Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 11(Dec):3371–3408
- Volkovs M, Yu G, Poutanen T (2017) Dropoutnet: addressing cold start in recommender systems. In: Proceedings of the 31st annual conference on neural information processing systems, Long Beach, CA, USA, pp 4964–4973
- Vuurens JBP, Larson M, de Vries AP (2016) Exploring deep space: learning personalized ranking in a semantic space. In: Proceedings of the 1st workshop on deep learning for recommender systems, Boston, MA, USA, pp 23–28
- Wakita Y, Oku K, Kawagoe K (2016) Toward fashion-brand recommendation systems using deep-learning: preliminary analysis. *Int J Knowl Eng* 2(3):128–131. <https://doi.org/10.18178/ijke.2016.2.3.066>
- Wang X, Wang Y (2014) Improving content-based and hybrid music recommendation using deep learning. In: Proceedings of the 22nd ACM international conference on multimedia, Orlando, Florida, USA, pp 627–636
- Wang H, Shi X, Yeung DY (2015a) Relational stacked denoising autoencoder for tag recommendation. In: Proceedings of the 29th AAAI conference on artificial intelligence, Austin, TX, USA, pp 3052–3058
- Wang H, Wang N, Yeung DY (2015b) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, Sydney, NSW, Australia, pp 1235–1244
- Wang H, Xingjian S, Yeung DY (2016) Collaborative recurrent autoencoder: recommend while learning to fill in the blanks. In: Proceedings of the 30th annual conference on neural information processing systems, Barcelona, Spain, pp 415–423
- Wang X, Yu L, Ren K, Tao G, Zhang W, Yu Y, Wang J (2017) Dynamic attention deep model for article recommendation by learning human editors' demonstration. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax, NS, Canada, pp 2051–2059
- Wei J, He J, Chen K, Zhou Y, Tang Z (2017) Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst Appl* 69:29–39. <https://doi.org/10.1016/j.eswa.2016.09.040>
- Wu S, Ren W, Yu C, Chen G, Zhang D, Zhu J (2016a) Personal recommendation using deep recurrent neural networks in netease. In: Proceedings of the IEEE 32nd international conference on data engineering, Helsinki, Finland, pp 1218–1229
- Wu Y, DuBois C, Zheng AX, Ester M (2016b) Collaborative denoising auto-encoders for top-n recommender systems. In: Proceedings of the 9th ACM international conference on web search and data mining, San Francisco, CA, USA, pp 153–162
- Wu CY, Ahmed A, Beutel A, Smola AJ, Jing H (2017a) Recurrent recommender networks. In: Proceedings of the 10th ACM international conference on web search and data mining, Cambridge, United Kingdom, pp 495–503
- Wu H, Zhang Z, Yue K, Zhang B, Zhu R (2017b) Content embedding regularized matrix factorization for recommender systems. In: Proceedings of the 2017 IEEE international congress on big data, Boston, MA, USA, pp 209–215
- Xu Z, Chen C, Lukasiewicz T, Miao Y (2017a) Hybrid deep-semantic matrix factorization for tag-aware personalized recommendation. [arxiv:1708.03797](https://arxiv.org/abs/1708.03797)
- Xu Z, Lukasiewicz T, Chen C, Miao Y, Meng X (2017b) Tag-aware personalized recommendation using a hybrid deep model. In: Proceedings of the 26th international joint conference on artificial intelligence, Melbourne, Australia, pp 3196–3202

- Yang C, Bai L, Zhang C, Yuan Q, Han J (2017) Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax, NS, Canada, pp 1245–1254
- Yin H, Wang W, Wang H, Chen L, Zhou X (2017) Spatial-aware hierarchical collaborative deep learning for poi recommendation. *IEEE Trans Knowl Data Eng* 29(11):2537–2551. <https://doi.org/10.1109/TKDE.2017.2741484>
- Ying H, Chen L, Xiong Y, Wu J (2016) Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback. In: Proceedings of the 20th Pacific-Asia conference on knowledge discovery and data mining, Auckland, New Zealand, pp 555–567
- Zanotti G, Horvath M, Barbosa LN, Immedisetty VTKG, Gemmell J (2016) Infusing collaborative recommenders with distributed representations. In: Proceedings of the 1st workshop on deep learning for recommender systems, Boston, MA, USA, pp 35–42
- Zhang Y, Wallace B (2015) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. [arxiv:1510.03820](https://arxiv.org/abs/1510.03820)
- Zhang F, Yuan NJ, Lian D, Xie X, Ma WY (2016a) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, San Francisco, CA, USA, pp 353–362
- Zhang W, Du T, Wang J (2016b) Deep learning over multi-field categorical data. In: Proceedings of the 38th European conference on information retrieval, Padua, Italy, pp 45–57
- Zhang S, Yao L, Sun A (2017a) Deep learning based recommender system: a survey and new perspectives. [arXiv:1707.07435](https://arxiv.org/abs/1707.07435)
- Zhang S, Yao L, Xu X (2017b) Autosvd++: an efficient hybrid collaborative filtering model via contractive auto-encoders. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, Shinjuku, Tokyo, Japan, pp 957–960
- Zhang S, Yao L, Xu X, Wang S, Zhu L (2017c) Hybrid collaborative recommendation via semi-autoencoder. In: Liu D, Xie S, Li Y, Zhao D, El-Alfy EM (eds) Neural information processing. Springer International Publishing, pp 185–193
- Zhang S, Yao L, Sun A, Wang S, Long G, Dong M (2018) Neurec: on nonlinear transformation for personalized ranking. In: Proceedings of the 26th international joint conference on artificial intelligence, Stockholm, Sweden, pp 3669–3675
- Zhao Y, Wang J, Wang F (2015) Word embedding based retrieval model for similar cases recommendation. In: Proceedings of 2015 Chinese automation congress, Wuhan, China, pp 2268–2272
- Zhao Z, Yang Q, Lu H, Weninger T, Cai D, He X, Zhuang Y (2018) Social-aware movie recommendation via multimodal network learning. *IEEE Trans Multimed* 20(2):430–440. <https://doi.org/10.1109/TMM.2017.2740022>
- Zheng L (2016) A survey and critique of deep learning on recommender systems. Technical report, University of Illinois
- Zheng Y, Liu C, Tang B, Zhou H (2016a) Neural autoregressive collaborative filtering for implicit feedback. In: Proceedings of the 1st workshop on deep learning for recommender systems, Boston, MA, USA, pp 2–6
- Zheng Y, Tang B, Ding W, Zhou H (2016b) A neural autoregressive approach to collaborative filtering. In: Proceedings of the 33rd international conference on international conference on machine learning, New York, NY, USA, vol 48, pp 764–773
- Zhou J, Albatal R, Gurrin C (2016) Applying visual user interest profiles for recommendation and personalisation. In: Proceedings of the 22nd international conference on multimedia modeling, Miami, FL, USA, pp 361–366
- Zhuang F, Zhang Z, Qian M, Shi C, Xie X, He Q (2017) Representation learning via dual-autoencoder for recommendation. *Neural Netw* 90:83–89. <https://doi.org/10.1016/j.neunet.2017.03.009>
- Zuo Y, Zeng J, Gong M, Jiao L (2016) Tag-aware recommender systems based on deep neural networks. *Neurocomputing* 204(C):51–60. <https://doi.org/10.1016/j.neucom.2015.10.134>