Knowledge-Based Systems 249 (2022) 108954

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A survey for trust-aware recommender systems: A deep learning perspective

Manqing Dong[a], Feng Yuan[a], Lina Yao[a,*], Xianzhi Wang[b], Xiwei Xu[c], Liming Zhu[c]

[a] University of New South Wales, Sydney, Australia
[b] University of Technology Sydney, Sydney, Australia
[c] Data61, CSIRO, Sydney, Australia

ARTICLE INFO

Article history:
Received 14 December 2021
Received in revised form 7 April 2022
Accepted 28 April 2022
Available online 8 May 2022

Keywords:
Recommender systems
Deep learning
Systematic survey

ABSTRACT

A significant remaining challenge for existing recommender systems is that users may not trust recommender systems for either inaccurate recommendation or lack of explanation. Thus, it becomes critical to embrace a trustworthy recommender system. This survey provides a systematic summary of three categories of trust issues in recommender systems: social-aware recommender systems, which leverage users' social trust relationships; robust recommender systems, which filter untruthful information, noises and enhance attack resistance; and explainable recommender systems, which provide explanations of the recommended items. We focus on the work based on deep learning techniques, which is an emerging area in the recommendation research.

1. Introduction

Recommender systems provide information, products, or services to meet users' personalized tastes and preferences. They have been applied in various domains, such as e-commerce applications [1]. Despite the success of recommender systems, a significant remaining challenge is that users may not trust the recommender systems for either inaccurate recommendation or lack of explanation. For example, a user may not trust a stranger's taste even though they have similar historical records; another example is that the system may recommend an item that is intentionally highly rated by malicious users. All these make trustworthy recommender systems urgent and important.

In this survey, we define three aspects of trust (see Fig. 1) that contribute to trustworthy recommender systems: social-awareness, robustness, and explainability.

Social-awareness. Social-aware recommender systems refer to recommend items with using information from the user social relationships. Recent studies suggest user's rating has a positive correlation with the average of the social neighbors for both trust-alike relationships or trust relationships [2]. On the one hand, based on the phenomenon that users' tastes are often influenced by their friends [3], leveraging the trust relationship has great potential to provide a trustworthy recommender system to the user and improve the recommendation quality. On the other hand, adding social information alleviates the cold start problem of traditional recommender systems.

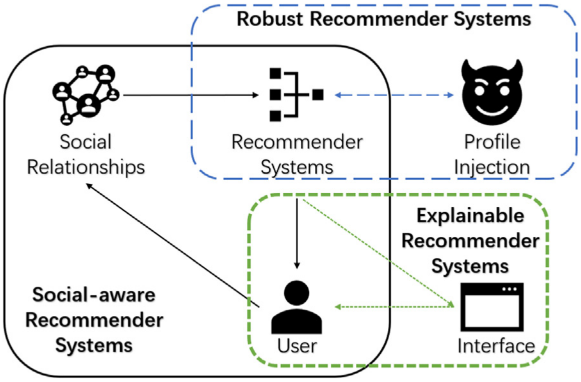

Robust Recommender Systems
Social Relationships
Recommender Systems
Profile Injection
Explainable Recommender Systems
Social-aware Recommender Systems
User
Interface

Fig. 1. Three scopes of trust issues in recommender systems.

Robustness. The recommender systems face a series of intended or unintended noises that challenge the robustness of the recommender systems. While most people rely on online product reviews to make purchase decisions [4], some reviews are untrustworthy due to attacks from spammers and bots. Moreover, recommender systems face noises caused by system errors or manual operations. Thus, there are two directions of research in improving the robustness of recommender systems. The first direction is to filter out noisy or malicious feedback from the data before executing the recommendation algorithm. Research in this direction aims to find patterns in users ratings to identify malicious content [5–7]. The second direction aims to develop

[*] Corresponding author.
E-mail address: lina.yao@unsw.edu.au (L. Yao).

https://doi.org/10.1016/j.knosys.2022.108954
0950-7051/© 2022 Elsevier B.V. All rights reserved.

Table 1
The difference between our survey and other related surveys.

Survey	Scopes		
	Social-awareness	Robustness	Explainability
Yang et al. (2014) [10]	✓		
Gunes et al. (2014) [11]		✓	
Gupta et al. (2015) [12]	✓	✓	
Zhang et al. (2020) [13]			✓
Si et al. (2020) [8]		✓	
Our survey	✓	✓	✓
Survey	**Methods**		
	Content-based	Collaborative filtering	Deep learning
Yang et al. (2014) [10]		✓	
Gunes et al. (2014) [11]	✓	✓	
Gupta et al. (2015) [12]	✓	✓	
Zhang et al. (2020) [13]	✓	✓	✓
Si et al. (2020) [8]	✓	✓	
Our survey	✓	✓	✓

noise attack-resistant algorithms [8,9] to improve the denoising capability of the recommender systems.

Explainability. Users tend to trust a recommendation when provided the appropriate information to understand the recommendation process and results. Explainable recommender systems [13] not only provide users with personalized recommendations but also generate descriptions of the reasons why the items are recommended. Therefore, explainability improves both the trustworthiness and transparency of recommendation results. In general, explainable recommender systems can be classified according to two orthogonal criteria: information source and methodology. Existing work utilizes a variety of contents, such as features of items/users [14], textual reviews [15], product images [16], and social connection [17]. Various approaches have been used to generate explanations, e.g., matrix factorization [18], graph-based models [19], topic models [20], deep learning [21], and association rule mining [22]. Besides, much work exists that combines different information sources and methods.

In this survey, we focus on the scope of deep learning models and give a thorough analysis of the recent work. Deep learning techniques have exploded during recent years. A recent survey on deep learning-based recommender systems [1,23] points out neural architectures have two advantages: being end-to-end differentiable and the ability to provide suitable inductive biases. One significant contribution of deep learning approaches is about representation learning. For example, many studies use deep learning to learn compact representations from auxiliary data such as content, tag, images, or social graph relationships and then use the compact representations for further recommendation [3,24].

Until now, considerable work has been conducted on summarizing deep learning-based techniques for trust-aware recommendation, including social-aware recommender systems [10], robust recommender systems [11,25,26], and explainable recommendation [13]. However, to the best of our knowledge, there is a lack of systemic survey on current deep learning-based trust-aware recommendation methods. The main differences between our survey and other related surveys are listed in Table 1. This survey aims to review the trust issue in recommender systems from a deep-learning perspective to fill the gap.

We outline three aspects of trust, i.e., social-awareness, robustness, and explainability, in Sections 2 to 4. For each aspect, we present the literature review and summarize the related deep learning-based techniques. We summarize several challenges of current trust-aware recommendation techniques and provide insights in Section 5. The following sections assume readers have a basic understanding of deep learning techniques and concepts of recommendation techniques.

Table 2
A Summary for social-aware recommender systems.

Method	Ensemble	Co-factorization	Others
Autoencoder	[3,28–30]	[31]	[31,32]
RNN	–	[33,34]	–
GNN	[27,34–36]	[37]	[38]
GM	[39,40]	–	–
Hybrid methods	[41–45]	–	[24,46,47]
Others	[20,48–50]	–	[51–54]

2. Social-aware recommender systems

2.1. Overview

Social relations have proven helpful in boosting recommendation performance [27]. The related work leverages social links between users to complement the sparse rating data and considers both users' ratings and the preferences of trusted neighbors to improve user preference prediction.

2.1.1. Traditional methods

Traditional classification for social-aware recommendation techniques includes memory-based methods and model-based methods. The **memory-based** methods deduce ratings of a targeted user via trust propagation based on ratings of its friends [36]. For example, Jamali and Ester [55] combine TrustWalker [56] with neighborhood collaborative filtering; they first run random walks on a trust network and then perform a probabilistic item selection strategy to generate recommendations. Similarly, Zhang et al. [57] extract reliable social information from user feedback and use top-k identified friends to infer user preferences. Matrix factorization is probably the most widely used technique for **model-based** social-aware recommendation. Wen et al. [53] learn vector representations of social relations via node2vec [58] and then combine them with user rating history to form user representation and conduct matrix factorization. Guo et al. [2] use an SVD++ [59] based methods with considering the user preference and friend's influence in forming the user representation.

2.1.2. Classification for the social-aware techniques

Here, we organize the techniques related to social-aware recommendations into the following categories, according to the way of combining social information: **ensemble methods**, which assume extrinsic (which will be affected by social relations) and intrinsic preferences of users and conduct recommendations based on the two types of preferences; **co-factorization methods**, which assume users share the same preferences in rating and

Table 3
The roles of different deep learning approaches for social-aware recommendation.

Methods	Functions	Pros	Cons
NN	Representation learning, Predicting	Simple, Keep all information	Need more parameter space, Unable to capture structural patterns
Autoencoder	Representation learning (encoder), Predicting (decoder)	Learn dense representation	May take high computation cost (if predicting by decoder)
RNN	Sequential preference/information modeling	Capturing sequential patterns	Lose information in long terms
CNN	Representation learning (for structured data)	Learning structured data (e.g. images), Save parameter space	Lose information in large pooling size
GNN	Social graph representation learning	Capturing graph information	May take high computation cost
Generative models	Representation learning, Predicting	Latent distribution learning	Rely on statistical assumptions
Attention mechanism	Automatic weighting, Visualization	Efficient, Explainability	May lose sequential information

social space [60]; **regularization methods**, which minimize the distances of latent features between trusted users and maximize the latent features' distances between distrusted users to reflect social proximity. We summarize the techniques in Table 2; in particular, we classify regularization methods as 'Others', which involves only one piece of work [31]. We will introduce the works according to the used deep learning techniques. Table 3 shows the basic descriptions of the methods.

2.2. Autoencoder-based methods

Autoencoder is a type of artificial neural network for learning compressed representations (encodings) for a set of high-dimensional data [30]. It can help recommendations with either learning latent factors of users and items (encoder) or reconstructing user's preferences (decoder). The learned latent representations from the encoder are normally further reconstructed by the decoder or cooperated with other methods for prediction [30]. So we classify the works by decoder-centered methods and encoder-centered methods.

Decoder-centered method learns from dense representation and reconstructs user rating patterns as predictions for recommendation. A key issue is to form the dense representation from the user's intrinsic rating patterns and the extrinsic social relationships. One way is to learn an **ensemble representation** of the two types of information for prediction. For example, Pan et al. [3] balance the contributions of the two representations learned from social relationships and rating history via a weighted layer; then, they use a correlative regularization to exchange information [3] and the unified latent representation to predict user ratings and trust relationships. Wang et al. [28], instead, directly concatenate the two latent representations for recommendation. Another idea is to assume that a user's social representation **share the same representation** with the user's rating representation. For example, Nisha et al. [31] generate a list of trusted users and minimize the distance between their social representation to learn users' social representations—they first use an autoencoder to encode patterns in users' ratings and patterns in item's history ratings; then, they decode the learned representations for recommendation; they also use a regularization item to control the distance between the user's social representation and the user's rating representation when training the autoencoder.

Encoder-centered methods. Another idea for utilizing autoencoder for social-aware recommendation is to combine the learned latent representation with other methods. For example, Rafailidis et al. [32] learn user latent representations from the social relationships via deep autoencoders and then use the middle-layer units of the network as the latent factors. Liu et al. [43] use stacked denoising autoencoder (SDAE) [61] to learn the social information. The input of SDAE is from K friends of a given user, and each friend is represented by a vector. By aggregating the information from all friends, a condense vector is used for representing the user. Such a representation vector is further combined with other methods for recommendation. Similarly, Wu et al. [29] use autoencoder to extract the compact representation of the social network and predict ratings by aggregating user information and item information via several fully connected neural network layers.

2.3. RNN-based methods

Recurrent Neural Network (RNN) has shown its power in dealing with sequential data, e.g., textual [62] and time-series data [63]. In recommendation problems, RNN-based methods generally target in dynamic user behaviors [64], preferences [65, 66] or the side information [67]. Such methods majorly use RNN to capture the sequential information and then learn a temporal or concrete representation for further uses. For the social-aware recommendation problem, RNN can extract users' temporal preferences and temporal biases of friends. Sun et al. [33] propose a recurrent-network-based model with attention for temporal recommendation. The method includes a static part that captures the consistent user preference and a dynamic part that captures the dynamic user preference. For the static part, the static social attention module is used to select the static social relationships for each user and to aggregate these social relationships to enrich the user's representation vector. For the dynamic part, an LSTM module is implemented for capturing the complex temporal latent representation of users, i.e., consider the social influences into the temporal preference modeling. Each part will predict a user preference score, and the final rating prediction is the sum of the scores from the two parts.

2.4. GNN-based methods

Graph neural network (GNN) has shown the effectiveness in learning on graphical data by the power of integrating node information and topological structures. As such, for social-aware recommendation problems, GNN has a great potential for mining the social graph structures and user–item graph. The key is to utilize the GNN to learn the latent factors of users and items [27]. For example, Fan et al. [27] learn the user latent factor and item latent factor via GNNs first, and then concatenate the two latent factors for the final rating prediction. For user modeling, the latent representation of the user is the concatenation of item aggregation and social aggregation. The process of item aggregation is aggregating the user rating histories toward different items (the representation for each item is the combination of item-vector and the item rating) with attention algorithms. The process of social aggregation is aggregating the user's friends rating histories (for each friend, the representation is the reconstruction of the user–item vector) with attention algorithms. For item modeling, the item's latent factor is the aggregation of other users' historical ratings toward the target item. Instead of static modeling of social relationships, the inference by friends may also change along with time. In this regard, Song et al. [34] consider a dynamic situation that social relationships dynamically influence users' interests. They propose a session-based social recommendation algorithm, which models dynamic interests and dynamic social influences. The model captures the user's current preferences by an RNN module, which models the user's historical actions, such as clicks. For modeling the friends' interests, the authors considered both short-term and long-term preferences—short-term preference is modeled by RNN with capturing the current session's preference, and long-term preferences capture the average interests. Then, each friend is represented by a concatenation of short-term and long-term preferences. To learn social-aware user representations, the authors use a graph attention algorithm (which is learned by the similarity between the target user and the friends) to leverage the importance of the social relationships and then aggregate them with different weights. Then, they combine the social-aware user representation with dynamic user interest. The probability distribution of recommending items is the softmax of the similarities between item embeddings and user hybrid representation. As one type of graph neural network, **Graph Convolutional Network (GCN)** generates the node embedding in view of message passing or information diffusion [37], which can encode the graph structure information as low-dimensional representations. Specifically, the embedding for each node is the aggregation of the information from the neighborhoods, and the neighbors' embedding is further learned from the neighbors of the neighbors, and so on [36]. For example, in [38], the authors represent the node in the graph with a two-layer graph convolutional neural network. Specifically, the embedding for a node (i.e., an item) is the aggregation of feature information (e.g., visual, textual features) from the node's local graph neighborhood. Each aggregation module learns how to aggregate information from a small graph of neighborhoods. These embeddings are then used for recommender system candidate generation via nearest neighbor lookup or as features in machine learning systems for ranking the candidates.

2.5. Generative models

Generative models generally describe models of joint distribution with data and labels. It will be taken to produce new samples with mechanisms sampling from the real data. The two most common types of generative models are generative adversarial nets (GANs) [68] and Variational Autoencoders (VAEs) [69].

2.5.1. GAN-based methods

Generative Adversarial Network (GAN) includes a generator and a discriminator to conduct adversarial learning. It has shown the effectiveness across various domains, given the ability to learn data probabilistic distribution and generate new samples. The generator attacks the discriminator by generating new samples with similar distribution as real samples. The discriminator is distinguishing the source of the samples, i.e., whether the sample is coming from real cases or generated cases. A min–max game is played between the two processors, which can promote both of the two models. This is known as adversarial training. When it comes to the recommendation problem, generative models are generally applied for (i) predicting missing values and (ii) enhancing the representation of items and users [70]. For example, Wang et al. [70] use a generative model to generate simulated user preference distribution of real data. For the social-aware recommendation problem, the critical issues are learning information from trust relationships and the way of combining such trust information with rating history. Fan et al. [39] design two adversarial learning modules for enhancing the user representations in the user–item rating part and the social part. In detail, for each part, a discriminator is designed for distinguishing the real instances and the generated samples, and a generator is designed for modeling the actual conditional distribution for a given user. For adaptively enhancing the representations in two parts, they utilize a bidirectional mapping between the two parts, wherein each iteration, the user's social representation will be updated by a nonlinear mapping operation from rating pattern representation, and then updated by the domain adversarial trainer; likewise, the user's rating pattern representation will be updated with the trained social representation, and then be following with the domain-specific training.

2.5.2. VAE-based methods

Similar to autoencoder-based methods, variational autoencoders can predict missing values or learn comprehensive representations. For the former task, the VAE predicts item ratings/scores by inferring the latent factors [71]. It consists of three parts: a bag-of-items vector i_u by user u is provided as input to the decoder; a latent user vector z_u is sampled from a Gaussian distribution with parameters specified by the encoder; and a new bag-of-items vector $\tilde{i}_u$. For the latter, the VAE learns representation for users or items. For example, Xiao et al. [40] consider three types of information for recommendation: user-trust relationship, user–item rating history, and item content information. For representing the users, each user is represented by the aggregation of trusted users. For representing the items, the content information for items is considered, where a variational autoencoder model is used to learn the latent patterns for content information. Such latent patterns are used to represent the items. Then, they consider using the traditional matrix factorization method for predicting the user–item ratings.

2.6. Hybrid methods

Several hybrid models are proposed to bridge the advantages of the above models for enhancing recommendation performance. The hybrid models may adopt hybrid algorithms or multiple types of input for recommendation.

According to the research targets and characteristics of features, multiple algorithms may apply. For example, RNN based methods are widely used for dealing with session-based recommendation problems; autoencoder is good at extracting a condense representation of the input. Then for solving a session-based trust-aware recommendation problem, such two methods can be applied together. Liu et al. [43] consider that user's preferences are changing over time. They use a stacked denoising autoencoder to learn the user representation, which is the aggregation of the friends' representations, at each time step. Then, such representation is used as the input of the LSTM module for predicting a user's current preference. A stacked LSTM for predicting the user's whole time preferences.

For recommendation problems such as image recommendation and movie recommendation, the auxiliary information will be considered. Wu et al. [41] target at an image recommendation problem. For the image information, they use CNNs for learning the embedded representation. Besides the images, they consider three auxiliary information: user upload history, social influence, and creator admiration. They use an attention model to leverage such three aspects. The user's preference is represented by the aggregation of all the information. And the rating prediction is based on the product of item embedding and the user's preference vector. Zhao et al. [46] design a heterogeneous social-aware movie recommender system by exploiting multi-modal movie contents (i.e., images and corresponding descriptions), users' social relations, and users' preference feedback. The model aims to recommend top-K movies to a user by calculating each movie's score based on the representations for the user and the movie. For learning a sharing item representation with both movie images and descriptions, they use a multi-modal learning approach: a deep convolutional neural network for the images and a deep recurrent neural network for the descriptions. The representations for users, which is the aggregation of friend relationships, are learned via DeepWalk [72]. Monti et al. [24] propose two different ways of predicting the user–item rating matrix. The first one is Recurrent Multi-Graph CNN (RMGCNN) architecture, which operates on the user–item matrix and operates simultaneously on the rows and columns. Both of the users and items are learned via Multi-Graph CNNs, where their relationships model the users, and the items are modeled with the images. The whole rating matrix X is learned by the RNN model step by step until providing stable predictions. For the second method, which is called Separable Recurrent MGCNN (sRMGCNN), operates separately on the rows and columns of the matrix. Gao et al. [44] consider a video recommendation problem. They propose a dynamic RNN to capture users' dynamic preferences by considering the video information, user interest, and user social relationships. The videos' semantic embedding includes visual features and textual features learned by pre-trained deep models. User interest modeling is based on the user view history, which is learned by topic modeling.

2.7. Others

2.7.1. Attention-based methods

Informally, a neural attention mechanism enables the neural network focus on a subset of its input (or features), i.e. assigns different weights to the input. For example, for the machine translation problem, it allows the machine translator to look over all the information the original sentence holds, then generate the proper word according to current word it works on and the context [73]. Now the attention mechanism is popular in many other areas, such as object recognition and image caption [74]. There is limited work only emphasizes on discussing the effectiveness of applying an attention mechanism into recommender systems; but on the contrast, as an enhancing module, the attention can work well when incorporated with other models for recommendation [1].

For the social-aware recommender system, the attention mechanism could be used for balancing friend influences. For example, Chen et al. [48] consider the problem that the influence of users' friends should be different and dynamic. For different items, the user may infer different friends' preferences. They propose a hierarchical attention module for the recommendation. First, each friend's representation is built on the user embedding and friend embedding, and is learned by attention mechanism. Then, for different friends, the authors also applied an attention mechanism for balancing friends' influences to get a final user representation. This user representation is then multiplied with item representation as a score for ranking. Rafailidis and Weiss [49] propose a similar structure that considers a subset of friends and uses an attention mechanism for social collaborative filtering.

2.7.2. Combining with matrix factorization

Traditional matrix factorization methods is predicting the ratings/scores by multiplying the latent representations of users h_u and items h_i, i.e. $\hat{r}_{ui} = h_u^\top \cdot h_i$. The difference between deep matrix factorization methods and traditional ones is that the latent representation learning is implemented with deep learning techniques. A common way for social-aware recommendation is to combine the social influences into the user representation. Fan et al. [51] learn the social embeddings via node2vec [58] and then use a multi-layer perceptron for learning the embeddings. Each user is represented by the social embeddings and initialized latent factors. Such user representation is further be used in probabilistic matrix factorization [75]. Bao et al. [42] use an attentive way for learning social influences. They first use autoencoder to learn compact representation of neighbors, where each neighbor is represented by h_v. Then the social influences for user u is modeled by $h_T = \Sigma_{v \in V} \alpha_v \cdot h_v$; α is the attention value. Then the user latent representation is given by $\hat{r}_{ui} = (\beta h_u + (1 - \beta) h_T)^\top \cdot h_i$, where β is a self-defined hyper-parameter. Wang et al. [52] argue that the matrix factorization method could be represented with a shallow neural network model. They consider a cross-domain recommendation problem [76], where they are trying to recommend a top-K items list from information-domain to users in social-domain. Specifically, they use a deep collaborative filtering model to predict the user preference $\hat{s}_{ui}$, which is calculated by the latent user representation h_u and latent item representation h_i. Both latent representations are learned from the initial embedding vector and attribute vectors (i.e., the hashtags). Then, the prediction of user–item interaction is further enhanced by integrating the social relationships: the intuition that users with strong connections are more likely to share similar tastes on items. The processing is minimizing the user latent representation gap between strongly connected users.

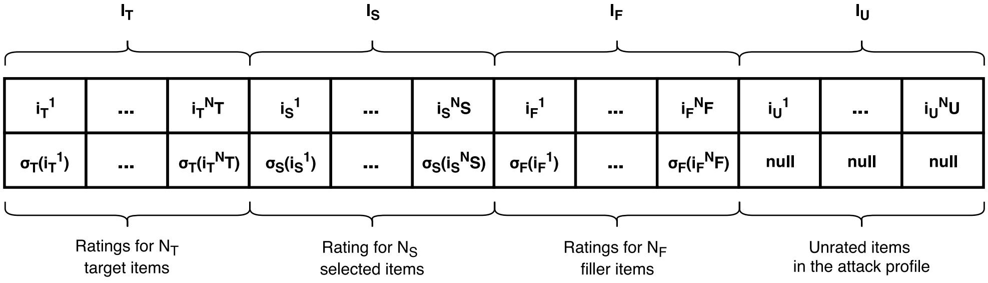
2.7.3. Others

Xiao et al. [50] propose a SVD++ [59] based model for recommendation. The inputs for the network are user's representation h_u, item's representation h_i and user's social representation h_t. They incorporate the social relationships by considering both its latent representation and the interaction with items. The following layer is represented as the concatenation of the above information, i.e. $[h_u, h_i, h_t, f(h_u, h_i), f(h_i, h_t)]$, where f is representing several neural networks. The prediction is made after a few fully connected layers.

2.8. Summary

To summarize, deep learning based social-aware recommendation algorithms have shown their effectiveness in different tasks. Different to traditional methods, deep learning based techniques need less manual extracted features and have the advances in grasping complex latent feature interactions. It is a trend to incorporate traditional recommendation methods with deep learning methods, e.g., graph neural network, which can leverage both advantages. Although effective, current deep-learning-based social-aware recommendation algorithms have the following challenges.

- The quality and the number of social links. For example, in most recommender systems, it is hard to get explicit and reliable links since few users indicate their social relationships.
- Most of the current works model the trust relationships with shallow models and ignore the high-order interactions among each users' friends. A user can take all the opinions of his friends into account and then come out of his thinking rather than linearly combine them.
- The assumption that user shares similar tastes with friends may mislead the recommendation. For example, a user can connect with people who have different shopping preferences.


I_T
I_S
I_F
I_U
Ratings for N_T target items
Rating for N_S selected items
Ratings for N_F filler items
Unrated items in the attack profile

i_T^1	...	$i_T^{N_T}$	i_S^1	...	$i_S^{N_S}$	i_F^1	...	$i_F^{N_F}$	i_U^1	...	$i_U^{N_U}$
$\sigma_T(i_T^1)$	...	$\sigma_T(i_T^{N_T})$	$\sigma_S(i_S^1)$	...	$\sigma_S(i_S^{N_S})$	$\sigma_F(i_F^1)$	...	$\sigma_F(i_F^{N_F})$	null	null	null

Fig. 2. The general components of an attacker profile.

- Most of the existing approaches ignore that users have different knowledge in different domains.

3. Robust recommender systems

3.1. Overview

The recommender systems promote the efficiency and benefits for both customers and merchants. Although effective, the recommendation schemes are vulnerable to shilling attacks or noise. For example, merchants may hire a group of spammers to insert their profiles and fake ratings into the systems, which will affect the performance of the recommendation [11] and also the customer's trust on recommender systems. Detecting such attacks and designing a robust recommender system is very important. Generally, researches in this field include shilling attack detection techniques and robust recommender systems.

3.1.1. Attack types

The shilling attacks can be classified by attacker types, the intent for the attack, the knowledge of the attack, etc. For example, according to the intention, the attacks can be categorized into push attack, nuke attack, and random attack [8]. The first intends to increase the popularity of the items while the second intends to decrease the popularity. According to the knowledge-cost, the attacks diverge into high-knowledge attacks (i.e., the attackers get some knowledge about other normal users) and low-knowledge attacks.

The attack profile consists of the history rated items and generally includes four parts: the target items, selected items, filler items, and unrated items. The target items I_T, which will be either "push" or "nuke" ratings, are rated with a rating function σ_T. The selected items I_S are rated by the attacker with particular intentions, e.g., the group attacks. The filler items I_F include randomly chosen items to make the profile look normal and harder to detect. Also, We denote the unrated items by I_U [8]. Different parts may have different generative functions for getting the ratings (see Fig. 2).

3.1.2. Research fields

The two directions to reduce the effects of shilling attacks on recommender systems include shilling attack detection techniques and robust algorithms. The former is first detecting the attacks, filtering the attack profiles, and constructing the recommender systems. The latter refers to construct attack-resistant recommender systems, i.e., robust recommendation methods [8].

Shilling Attack Detection Algorithms are mainly discussing the way of detecting malicious user files. According to the research targets, the shilling attack detection algorithms can be used for detecting point (personal) attacks or collaborative (group) attacks. The point attacks may represent an irregularity or deviation that happens randomly and may have no particular interpretation. Also, according to the nature of input data, we may utilize sequential (e.g., textual information and time series) or non-sequential data (e.g., images, user profiles). Techniques used for shilling attack detection can be roughly classified into statistical methods, supervised classification methods, semi-supervised methods, and unsupervised clustering methods. The *statistical methods* are focusing on detecting the outlier items. For example, statistical testing is used for identifying the differences between the sample distribution. Zou et al. [5] introduce a probabilistic inference network and the Belief Propagation (BP) algorithm [77] to perform inference efficiently. For the *supervised classification methods*, most of the work conduct feature engineering first and then implement different algorithms. Features such as rating deviation, the similarity with top neighbors are considered. For example, Yang et al. [6] propose three new features, i.e., the filler size with maximum, minimum, and average ratings, on filler or selected items to identify the attack profiles. The features are analyzed with statistical tests and classified by a variant of AdaBoost. *Unsupervised clustering approach* is normally clustering the users into groups and then eliminate suspicious users. For example, Bhaumik et al. [7] apply k-means clustering on user-profiles and identify the small clusters as attacker groups.

Robust Recommender Systems are focusing on developing attack-resistant systems, which is trying to reduce the influence of shilling attacks. Current robust recommendation algorithms mainly lie in two aspects: constructing robust algorithms or considering trust relationships. We have discussed the latter one in the previous section. Thus, here, we majorly introduce the former one: robust algorithms for recommendation. Some work adopts matrix factorization for the methods. For example, Alonso et al. [78] use a matrix-factorization-based method to identify the shilling attacks. According to the observation that fake ratings occur during a short interval of time, they assume that malicious profiles will affect the reliability of the model in an anomalous way. They use two matrix factorization models to obtain real prediction errors and the estimated prediction errors; the second error is used for evaluating the prediction reliability. Zhang et al. [79] incorporate the L1-norm into the loss function to improve the robustness. They claim that the squared error function is sensitive to large residuals. Yu et al. [80] design a robust matrix factorization model with kernel mapping and kernel distance. He et al. [81] consider to improve the robustness of the recommender system by adding an adversarial module to the training.

3.2. Deep learning-based shilling attack detection algorithms

For the shilling attack detection problem, the key point is to evaluate users' suspiciousness, posts, and reviews. One advantage of using deep learning to detect the shilling attacks is its ability to capture complex structures in the data. Also, there is a need for large-scale detection techniques as the volume of data increases in the real world cases. Unlike traditional methods, deep learning-based approaches need less manually defined features and thus solve the problem in an end-to-end trainable way [82].

Some use convolutional neural networks for mining the local features. Convolutional neural networks work as feature extractors; they can learn from local representations and map them into higher or lower dimensional representations for further uses [83]. Many studies mine textual information for detecting the suspicious ratings/reviews [84]. For example, Zhang et al. [85] propose a deep model to identify the review spam. They assume the fraudulent users lack real experience, while normal users have real experience; then, the textual information should indicate different patterns between the fraudulent users and normal users. Li et al. [86] use word-to-vector [87] to represent the textual features, and use CNN to learn the semantic representation. Some work also considers the user's behaviors [88]. Wang et al. [89] consider the cold-start problem for new-coming users. They use CNN to learn the embedding from both textual and behavioral information.

Some use RNN-based methods for targeting the sequence input. Recurrent Neural Network (RNN), which has the function of memory, has shown its efficiency for processing sequential information. The following work, such as Gated Recurrent Unit (GRU) [90] and Long Short-Term Memory (LSTM) [91] are further designed for solving the gradient vanishing problems. As for detecting malicious ratings or reviews, the RNN based methods are used for learning sequential patterns such as texts and time series. For example, Ren et al. [92] use CNN to learn from words and use a bidirectional-GRU for learning the sentences. The learned representation is further be used for prediction. Similarly, Wang et al. [93] use LSTM to learn from texts.

Some consider hybrid methods or multiple inputs. For example, Wang et al. [94] propose a hybrid method that learns from both the review content and product information. They use a tensor factorization algorithm to learn the latent representations from reviews and products. Then, the learned representations are further combined with deep learning-based classifiers. Dong et al. [95] use autoencoder to extract latent representations from textual information and user behavioral patterns. Aghakhani et al. [96] improve model performance by adding adversarial noises.

3.3. Deep learning for robustness of recommender systems

3.3.1. Introducing noises to recommender systems

An intuition way for enhancing the robustness of recommender systems is adding noises in the recommendation training process. By doing so, the model is forced to learn robust parameters to improve denoising capability.

Randomly generalized noises. Some may add human-made noise into the input. For example, the model may be combined with additional bias terms; the inputs are corrupted before feeding in the models. By doing so, the model is forced to learn the most informative and robust parameters to improve the recommender systems' robustness. One example is the denoising auto-encoder (DAE) [9], which corrupts the inputs with adding noises. Wu et al. [97] propose the collaborative denoising autoencoder (CDAE) that has similar ideas of DAE. They first corrupt the inputs, i.e., the ratings, with Gaussian noises, feed the inputs into neural nets, and get a dense representation of such corrupted inputs. The decoder of the model tries to recover the original values of the compact representation and predicts the ratings for recommendation. Strub et al. [98] also corrupt inputs by stacked denoising autoencoders [61]. Besides, they incorporate the side information, such as the user profile (age and gender) and the movie category, to enhance the model's robustness. Wang et al. [99] propose a collaborative recurrent autoencoder that integrates RNNs and denoising autoencoders for recommendation. They design a robust recurrent network to process the item textual information and overcome the shortage of using mancrafted features. In detail, the recurrent network is designed in the autoencoder way, i.e., the layers in the encoder and decoder are recurrent networks. The recurrent autoencoder can learn both the sequential information and the dense representation of inputs. The learned dense representation is regarded as the item representation and further combined with user representation for rating prediction. Furthermore, in the case of over-fitting, they design denoising and a beta-pooling approach.

Adversarial noises. In recent work, some attempt to add adversarial noise to the model to improve the robustness. For example, He et al. [81] consider to solve the problem by adversarial training based on Bayesian Personalized Ranking. In detail, they corrupt the model parameters with adversarial noises; the adversarial personalized ranking is made by minimizing the training loss and maximizing the adversarial loss, i.e., identifying the worst case of the corruption. The model is optimized via stochastic gradient descent. Similarly, Yuan et al. [100] propose an adversarial training framework for recommendation. The model is designed based on the collaborative denoising autoencoder. Unlike the traditional CDAE method that corrupts the inputs, they insert a noise mixing layer into the autoencoders. The addressed adversarial training strategy includes a training step to obtain optimal parameters, and a re-training step to minimize the training loss while maximizing the adversarial noise loss. Wang et al. [101] consider a session-based recommendation problem and design a memory network for storing the long term and short term user preferences. They use generative adversarial nets to generate negative samples to improve the model parameter inference. Wang et al. [70] propose a generative adversarial model for recommendation. Like the generative adversarial network, the model includes two modules: the generative model simulates the real user profiles by capturing the patterns from the raw datasets. In contrast, the discriminative model tries to identify such generated samples from the real ones. In such a case, the generator does similar work as malicious users. The discriminator determines the malicious content; thus, they promote both performance and improve the robustness of the recommender system.

3.3.2. Other methods

The attention mechanism can filter the uninformative information from the input and prevent the side effects of noise. Many works consider mining informative patterns from user history records. Jhamb et al. [102] examine from user preferences that they propose Attentive Contextual Denoising Autoencoder and use attention mechanism for encoding contextual attributes of user preferences. Zhou et al. [103] consider the heterogeneous user behaviors; they use a self-attention algorithm to predict user preferences by aggregating the different contributions of the user behaviors. Loyala et al. [104] study the user transitions in different sessions by an RNN-based method and use the attention module to learn the more expressive portions of the sequences. Ying et al. [105] also use attention mechanism for a dynamic situation of user preferences. Two attention layers are utilized to learn users long-term preferences and learn from both long-term and short-term preferences separately. Liu et al. [106] provide similar ideas for incorporating long-term and short-term preferences. Some works consider from other patterns. Seo et al. [107] mine the potential benefits from textual features. They build vector representations of user and item using attention-based CNNs, where the attention mechanism is used for extracting keywords before the CNN modules; such vector representations will be further used to predict the ratings. Tay et al. [108] focus on the user–item relationships and use an attention module that can visualize the model and enhance the model performance by capturing the significant patterns. Chen et al. [109] propose two attention modules that one for selecting informative components of multimedia items and one for scoring the item preferences.

Besides, incorporating auxiliary knowledge from other domains, such as social relationships (we have discussed in Section 2), can also help improve the robustness of the recommendation.

3.4. Summary

Shilling attack detection has been the traditional research field to tackle the robustness of recommender systems, by filtering or removing malicious profiles. Some other work designs robust machine learning methods to neutralize the impact of the malicious profile, e.g., adding human-made noise to improve the robustness. Both ways show the improvement in the recommendation performance. However, there are still several challenges in this area.

- It is hard to unify such two methods in a trainable end-to-end model for leveraging both capabilities. And most of the deep learning methods are sensitive to the data resource and the cross-domain conditions.
- Most of the current work does not consider the dynamic conditions, i.e., anomalous behavior may change over time.
- The anomalies are rare entities in real life. Thus, it is challenging to obtain labels.

4. Explainable recommender systems

4.1. Overview

Explainable recommender systems enable trust-aware recommendation from a different perspective [13]. Explainable recommender systems differ from other systems in offering reasons why the systems provide users with such recommendations and giving system designers guidance to improve the recommendation results. It not only improves the effectiveness and user satisfaction of recommendation systems but also enables the systems to generate trustworthy recommendations. Recently, a host of explainable recommendation approaches have been proposed, including but not limited to matrix factorization, deep learning, association rule mining, topic modeling, and knowledge-graph models. Despite such variety, these methods can be divided into two groups in general. ***Post-hoc*** [110,111] methods do not modify the recommendation algorithm itself, but attempt to explain the results, such as "this item is the most popular" and "people like the same item as you do also bought". These methods often cannot explain the recommendation mechanism, and the diversity of explanations is limited. On the contrary, ***Embedded methods*** [109,112,113] design explanation-oriented recommendation models so that the recommendation process itself can automatically generate explanations which are normally selected from the side information, e.g., texts or images. In this section, we narrow down our focus on only deep learning models, which belongs to the family of **embedded explanation methods**. Deep learning has recently become very successful in recommendation tasks [1]. Like other embedded methods, we find that most recent work based on deep learning leverages text or image information, e.g., user reviews, product photos, and movie posters for explanation generation. Most of them are proposed to explain a specific recommendation model, but, recently, some work addresses the recommendation explainability from a model-agnostic perspective [114,115]. For instance, Wang et al. [114] employ reinforcement learning to explain any recommendation model. Therefore, in this section, we classify the previous work into five categories: 1. traditional explanation based on collaborative filtering; 2. explanation using textual sentences; 3. explanation via visual contents; 4. explanation via temporal dynamics; 5. other deep learning-based explainable recommendation models (see Table 4).

4.2. Explanation on collaborative filtering

In the early days of recommendation explanation research, collaborative filtering (CF) is the fundamental method for personalized recommendations. CF leverages the users' implicit or explicit feedback from which explanations can sometimes be generated in a very straightforward way. For example, a user-based CF system decides whether to provide a certain user with an item according to the ratings from his/her neighbors, which can be considered a form of explanation. Similarly, in item-based CF, explanations are generated for a target user based on whether the rating given to an item is similar to the other already-given ratings. However, this is not the case for deep learning-based CF. With several deep learning-based recommendation models being proposed, the state-of-the-art performance in various recommendation tasks, such as rating prediction, top-N recommendation, and sequential recommendation, has been dominated by deep approaches. Still, most of them cannot explain their recommendation results because what the deep neural networks have learned usually is hard to interpret. Abdollahi and Nasraoui [116] focuses on the interpretability of Restricted Boltzmann Machines (RBM) based CF recommendations without relying on any auxiliary data, such as item content or user attributes. This work follows a similar idea of explainable user-based CF. It introduces the concept of "explainability score". The explainability score of a item for a user is calculated from the rating distribution of the user's neighbors, which ranges from zero to one and indicates the consistence of the neighbors' ratings. For example, a higher score means most of the neighbors have same preference for an item, and a zero score means there is no rating record from the user's neighbors—and thus is not explainable for the user. Then, the authors employ the conditional RBM model with an additional visible layer with the same number of hidden units as the number of items. The output value of each hidden unit in this layer is limited within 0 and 1, representing the above explainability score. In this way, the conditional RBM model tends to recommend items that are explainable. In essence, this approach provides explanations via user-based neighborhoods.

4.3. Explanation on textual data

In recommender systems, textual contents are a common and major source of auxiliary information such as user reviews and product descriptions. Numerous deep learning techniques have been adopted to exploit textual data, such as CNN [21,112, 118], RNN [16,125,126], and attention mechanism [109,117,122]. Among the above methods, CNNs are often employed for deep feature extraction from text and are combined with an attention mechanism to generate explanations [21,112]. At the same time, RNNs are often used for textual explanation generation [126,127]. Here, we review these deep learning-based explainable recommendation models that exploit textual side information and analyze their strengths and weaknesses.

The first group of methods combines **CNN** with an attention mechanism to analyze the textual data. Seo et al. [21] aggregates all the review texts given by a user and an item respectively to form two sets of representations from which the abstract features of different users and items are learned via convolutional neural networks with dual attention mechanism (i.e., global and local attention). The predicted ratings are then

Table 4
Summary for deep learning-based explainable recommender systems.

Method	Data type			
	Ratings only	Textual reviews	Images	Temporal data
RBM	[116]	–	–	–
Attentive CNN	–	[21,112,117,118]	[109,113,119]	[120,121]
Attentive RNN	–	[122]	[120,123–127]	–
GAN	–	–	[128,129]	–
Memory networks	–	–	–	[130]
Others	–	[114,131]	[16,41,132]	–

generated from the learned features similar to that of matrix factorization. In the meantime, the dual attention networks enable word focusing in the review texts. Different from the architecture in [21], Chen et al. [112] proposed a neural attentional regression model with review-level explanations (NARRE) which employs the DeepCoNN network to process the reviews [133]. Although NARRE only uses a single attention layer on the output of the DeepCoNN network, it is not only capable of generating highly accurate prediction ratings. It can also choose useful reviews that offer a form of review-level explanation to the target users. The above two methods use review texts as inputs only. In other words, they ignore user–item interactions and fail to model the users' rating behaviors completely. Wu et al. [117] proposed a context-aware user–item representation learning model (CARL) to overcome such shortcomings. CARL fuses two different networks, one for review feature extraction and the other for user–item interaction feature extraction. To process reviews, CARL employs an attentive CNN neural network, and to model user–item rating interactions, it adopts a matrix factorization-like approach to learn user/item latent representations. The final predicted ratings are fused by a dynamic weighting scheme between the outputs of the two networks. Unlike static methods that apply attention only on textual data, Chen et al. [118] combine a gated recurrent unit (GRU)-based network (which models user dynamic ratings) with a sentence-level CNN (which profiles an item by its reviews) to build a dynamic explainable recommender (DER). DER applies attention to the mixture of a user's time-varying preference at a certain time and the sentence-level features of the item reviews, merging sentence embeddings under "user-aware" attention weights. Therefore, DER can provide explanations in a dynamic, personalized manner. Despite the model differences, in all the above approaches, the recommendation explanations are produced in the form of a group of words with high attention weights to help the user understand the recommendations.

The second group of methods exploits **RNN** [134], a very effective family of deep neural networks for natural language processing. Some approaches introduce attention mechanisms to RNNs for similar reasons as the above attentive CNN-based models to select highly relevant words from the review texts as explanations [122]. Most existing models take advantage of the generative ability of RNNs to produce user/item review explanations [16,120,123–127]. Cong et al. [122] proposed a hierarchical attention-based network(HANN), which generates explanations by considering the contribution of reviews to the overall ratings at two levels (namely word level and review level). HANN is similar to Seo et al. [21] in that HANN replaces CNN by GRU-RNN. HANN also splits the textual data into user reviews and item reviews that are further fed into two separate GRU-based deep neural networks, known as the user net and the item net. Dual attention is adopted, one at word level for intra-review attention and the other at review level for inter-review attention. Both nets are fused by fully-connected layers to predict the ratings. The explanations are generated using the attention scores at both levels. The darker the pink color, the higher the attention score reaches the review level. Word level attention scores are similarly denoted by the green color. In this way, HANN extracts useful words from the reviews to form the explanation and, meanwhile, globally distinguishes the effectiveness of reviews on the final predicted rating scores.

A series of RNN-based explainable recommender systems are proposed to leverage the powerful text generation ability of RNNs. Costa et al. [123] designed a character-level generative concatenative network based on LSTM cells [91], where the ground-truth ratings serve as auxiliary information and are concatenated into the input layer. Therefore, the model can generate reviews following the directions pointed by the rating scores. By adjusting the hyper-parameters, the model can provide very natural explanations for human readers. Instead of generating reviews, Li et al. [120] proposed a multi-task learning model, i.e., the neural rating and tips generation network (NRT). NRT takes ratings and reviews as context and produces abstract tips. User–item rating pairs are first used to learn the user/item latent factors via multi-layer perceptrons (MLP) that form the rating regression network. These latent factors are then fed into a standard MLP-based review text generation network whose output layer, together with the predict ratings, serves as the context of a GRU-RNN based tip generation network. The multi-task objective is then composed of the rating regression loss, the review generation loss, and the tip generation loss. The generated tips are both concise and vivid enough to predict users' possible experiences and feelings. Another multi-task recommendation model is proposed by Lu et al. [124]. The authors utilize the adversarial sequence to sequence learning techniques. The reviews are first encoded into latent feature vectors using a bidirectional GRU-RNN network and decoded by a single GRU-RNN network to create a review autoencoder structure. This autoencoder is adversarially trained with a CNN-based review discriminator to identify if a piece of given review is written by user i on item j. Unlike NRT, which considers ratings as the context of the explanation generation process, The model feeds the latent textual features into a matrix-factorization-based rating prediction algorithm. Both models are jointly trained using the alternating least squares (ALS) technique [135] to perform rating prediction and explanation generation. Furthermore, the RNN-generated reviews provide explanations and act as inputs for the recommender system. To test whether the generated reviews are more effective in recommendation than human-written reviews, using DeepCoNN [133] as the recommender, Ouyang et al. [125] compare human-written reviews with synthetic reviews that are produced at both character and word levels by popular review generation models. Results show that synthetic reviews can carry more consistent information appropriate to the demands of a recommender system than human-written reviews, justifying the feasibility and rationality of using generated reviews to explain recommendations. Instead of using reviews or tips as inputs, Zhao et al. [126] feed user/item side information (e.g., user/item tags, item title, user gender, etc.) into a recurrent attention generation network to produce reasons for the explainable recommendation in conversation applications. Similarly, Suzuki et al. [127] adopt an MLP network to encode multicriteria evaluation ratings (e.g., overall rating, location rating, service quality rating, price rating, etc.) into a latent vector. An attentive LSTM-RNN network then decodes the vector into reviews to generate personalized explanations for the predicted ratings.

4.4. Explanation on visual data

Compared with textual data, visual contents often contain more information that can be exploited for recommendation explanations. Most previous image-based recommendation approaches transform images into latent representation vectors to be incorporated into recommendation algorithms [136–139]. However, such approaches are hardly useful in ex planing why a particular item is recommended. Recently, some initial steps have been taken toward the visual explainability of recommendations via exploiting the power of deep learning. Most existing recommendation models using visual data adopt CNN as the building block, given its popularity and success in processing visual data. Other deep learning techniques for explainable recommendations include the attention mechanism [109,113,119], and generative adversarial network (GAN) [128,129]. Apart from generating explanations from the visual data itself, some approaches regard images as auxiliary information to help explain the recommendations [16,41,132]. Unlike textual data based explainable recommendation models, all the above approaches explain the recommendation results straightforwardly.

Similar to those models that leverage attention mechanism for textual data, the first type of visually-explainable recommender systems applies **attentive deep neural networks** to select a group of "physical regions" [109,113] or "semantic regions" [119] from the images as the explanations. The earliest attempt we can find is from Chen et al. [109], in which work the attentive, collaborative filtering (ACF) model is proposed via hierarchical attention at both component level and item level. ACF combines the latent factor model with an attentive neural network that processes the features from items to provide top-N recommendations using implicit feedback. The item features are extracted by a CNN-based deep network, ResNet-152 [140], from images or video frames. After processed by the dual attention network, these features are merged with users' latent factors via element-wise addition to reflect the users' detailed preferences. Bayesian Personalized Ranking (BPR) [141] is adopted as the last step to generate the final recommendations. The explanations are given by the attention weights where the higher the weights, the more probable the user will like the entire images or the regions of images. Chen et al. [113] exploits both item images and user's textual reviews whose features are extracted via VGG-19 and GRU-RNN, respectively. The VGG-19 produced image features are split into several regions, which are then passed through an attention layer for explanation and merged with the item latent factors to represent the items. The user and item latent factors are combined with the item representation vectors to serve as the GRU-RNN network's inputs for review generation. This model is named review-enhanced, visually explainable collaborative filtering (Re-VECF). Unlike ACF that aims at user representations, Re-VECF focuses on item representations and uses a single attention layer to adopt element-wise multiplication to merge the image features and item latent factors. While the above two approaches exert attention on "physical regions" of the images, Hou et al. [119] propose Semantic Attribute Explainable Recommender System (SAERS) to understand users' semantic preferences via integration of the Fine-grained Preferences Attention (FPA) mechanism and the Semantic Extraction Network (SEN) for fashion recommendation. SAERS converts each attribute extracted from a particular region of the clothing images into one dimension in a semantic attribute visual space. SEN consists of the CNN-based ResNet-50 [140] network for semantic attribute classification and the Gradient-weighted Attribute Activation Maps (Grad-AAM) [142] for location and extraction of attribute representations in a weakly-supervised manner. The authors then adopt FPA to align users' latent factors with the semantic attribute visual space. Each user's latent factor is concatenated with one transferred semantic attribute representation vector, upon which the attention mechanism is applied to learn the user's preferences over different semantic attributes. Finally, BPR is used for recommendation, where the ratings are predicted via the inner product of user and item latent factors. The learned attention weights are visualized, and the weights indicate how much the user prefers a particular attribute. For instance, when a dress is recommended to user C, the model explains that this dress has a V-shaped neckline, which is reasonable because, according to user C's purchase history, she has bought three V neckline dresses before. Therefore, the recommendations are visually explained, improving the trust of the system.

The second type of recommendation approaches based on visual explanations exploits **GAN**. Similar to SAERS, Kang et al. [128] addresses the fashion recommendation by employing the Siamese CNNs [143] to extract "fashion-aware" image features to give the notations of "style". Although this is enough to provide explanations from a certain aspect, the authors further adopt conditional GANs [144] to generate images leveraging semantic inputs, where the product's top-level category is chosen as the condition. Thus, this approach can generate novel items that are likely to satisfy the users and are not in the training dataset. This way, the explanations for a series of recommended items can be summarized into such generated images. Kumar et al. [129] tackles the pairing problem in fashion recommendation via an enhanced conditional GAN model called c^{+}GAN. Given one piece of clothing image, the model recommends a set of items that best match the given clothing in a generative manner. c^{+}GAN modifies the generator with a classical mean squared error (MSE) loss and also a simplified perceptual loss using discrete cosine transform (DCT) coefficients of the generated as well as the target images. A simplified lensing technique [145] to the discriminator is applied to stabilize the generator training. Equipped with these techniques, c^{+}GAN can generate very meaningful fashion items as recommendation explanations.

Bharadhwaj [132] adopts the content-based similarity approach to recommend images. The authors modify the VGG-16 network by layer-wise relevance propagation [146], which enables relevance conservation at each layer in a pixel-wise manner. Given the purchase history of the target user (query items), the model can generate a list of other items that most resemble the query items. The explanations are given by the highlighted pixels that are the most informative for inferring which items go along well with the query item.

Other explainable recommender systems consider the visual features extracted via deep neural networks as the auxiliary information to generate non-visual explanations. Lin et al. [16] applied multi-task learning to recommendations and proposed the neural outfit recommendation (NOR) model. NOR recommends outfits to users with abstractive comments generated as explanations. To achieve both tasks, NOR adopts two neural networks, i.e., the outfit matching network and the comment generation network. Equipped with the mutual attention mechanism, the outfit matching network utilizes CNNs for visual feature extraction. These abstract visual features are further transformed into rating scores to predict the most matched outfit. To generate textual explanations from the aforementioned visual features, the authors exploit the cross-modality attention over the above CNN network and a GRU-RNN network, leading to the comment generation network. The recommendation explanations are given in the form of generated comments, mostly focused on the general opinions on the matched pair of outfits. Wu et al. [41] exploited the hierarchical attention mechanism in the field of image recommendation in social networks. The proposed model leverages heterogeneous data, e.g., users' rating behaviors, social network, upload behaviors, images. And from such complex relationships between users and images, the model represents these contextual factors as different sets of embeddings. A hierarchical attention network is then applied to attend differently to various embeddings. Images are represented using their content vectors, extracted by VGG-19, and their style vectors, generated by a CNN-based synthesis method [147]. However, this model does not generate explanations straightforwardly, but the recommendations can be intuitively interpreted by the learned attention weights of different aspects.

4.5. Explanation on temporal data

The sequential recommendation takes advantage of the temporal characteristics that exist in user dynamic behaviors to improve recommendation effectiveness [148]. The temporal aspect provides another dimension to generate explanations for the recommendations. A user's history is no longer a collection of unordered items, but a sequence of time-aware items. The order itself can provide certain explanations. For instance, if item i and j are complementary, a user bought item j at a certain time might be explained by the fact that this use had bought item i sometime earlier. With such observation, Li et al. [120] proposed the Neural Attentive Recommendation Machine (NARM) learn the user's primary intention in the current session. NARM employs GRU-RNN as the basic building block. NARM contains a global encoder that interprets the last hidden state in the RNN as the user's behavior feature. A local encoder that interprets all the hidden states in the current session as the user's primary purpose feature. The attention mechanism is applied to the local encoder to learn different weights for the hidden states so that the model can tell which past items contribute more to the future items. The two encoders are then combined as inputs of the decoder, predicting the recommendation possibility of each candidate item. In a particular session, the importance of items is reflected by the depth of the colors. This can give certain explanations on the next recommended items. To be specific, the users' decisions on the next clicked items are more influenced by those near the end of the session than those at the start, which is consistent with people's purchasing or browsing behavior we have noticed in reality. Tang et al. [121] proposed a Convolutional Sequence Embedding Recommendation Model (Caser) as another solution to the sequential pattern extraction problem. Caser embeds a set of recent items into a two-dimensional matrix whose dimensions represent the time and latent features. Two convolutional filters, one vertical and the other horizontal, are then applied onto this matrix to learn sequential patterns expressed as local features. The two filters capture patterns at different levels. Horizontal filters aim for union-level patterns via unifying the data into multiple sizes. In contrast, vertical filters aim for point-level sequential patterns by calculating the weighted sums using the previous items' latent representations. For clarity, the visualization of vertical filters reflects the importance of different past items. The horizontal filters can effectively extract Union-level sequential patterns. The recommended $\hat{R}_3$ (ground truth) is generated by the union of S_3, S_4 and S_5 due to the same genre they belong to. If any of S_3, S_4 and S_5 is masked in the horizontal filters, the ranking position of $\hat{R}_3$ is largely reduced. Chen et al. [130] took advantage of the memory mechanism for long-term memory and integrated collaborative filtering into a memory-augmented neural network (MANN). MANN stores and updates users' historical records explicitly and is also able to extract the intuitive patterns of how their previous decisions and behaviors influence users' future actions. MANN can capture two different types of sequential patterns. "One-to-one" behavior pattern generates sequences where the most recent action only influences the next action. "One-to-multiple" behavior pattern generates sequences where the same previous action influences a set of continuous behaviors. Both patterns can be widely observed in practice. For example, when browsing web pages, one may keep following the related links on each page and form a "one-to-one" pattern. When searching keywords through a search engine, one may browse multiple pages related to the same keywords, leading to a "one-to-multiple" pattern. The discovery of such patterns by MANN can explain why a particular user will buy a certain item in the future.

4.6. Other approaches

Wang et al. [114] proposed a reinforcement learning framework for explainable recommendations, which is quite universal because instead of integrating a certain explanation mechanism into a recommendation model, it has no restrictions on the details of the model to be explained. The authors consider users, items, side information, and a recommendation model to be explained as the environment. Two couple agents are employed, one for explanation generation and the other for explanation discrimination. At each state, the generator agent gives a piece of explanation, taking the user–item pairs as inputs. In contrast, the discriminator agent takes the generated explanation as input to predict rating scores. The agents' reward is calculated by measuring how similar the agent-predicted scores are to the recommendation model-predicted scores and the presentation quality (e.g., readability and consistency) of the generated explanation. By taking the textual sentences as interpretable components, the authors then adopt a personalized attention-based neural network as an instantiation of the proposed framework and show that it can well explain the recommendations via sentence-level explanations. Lin et al. [131] integrate the rating score prediction task and the explainable word generation task into a unified neural network. In this model, neural collaborative filtering (NCF) [149] is applied to the user-POI rating matrix to predict the rating scores. The reviews are transformed into syntax relations by utilizing the spaCy CNN dependency parsing model [150,151], which are further organized into pairs <opinion, aspect>. The learned user embeddings from NCF are then clustered based on cosine similarity. The textual explanations are then extracted from the pre-processed word pairs in the top-K users' reviews.

4.7. Summary

In this section, we reviewed deep learning-based explainable approaches for trust-aware recommendation. After introducing the general techniques for explainable recommendations, we focused on deep learning methods that leverage collaborative filtering, textual data, visual data, and temporal data. Generally, deep learning aims to explain the recommendation results from the mechanism of how the recommendation process works. Most models leverage textual reviews, item images, and temporal information of the user–item interactions to deal with limited user–item ratings. Considerable work combines attention mechanisms with deep neural networks (e.g., RNNs, CNNs, and memory neural networks) and generates explanations from auxiliary data (i.e., reviews, images, sequential patterns). Another category of models adopts generative methods (e.g., RNN and GAN), to provide novel textual/image explanations. Other models include traditional content-based similarity methods, hybrid methods that exploit texts and images, and reinforcement learning that controls the quality of explanations. Overall, deep learning has demonstrated as a promising approach to explainability for trustworthy recommendations.

5. Discussion

Though effective, current researches face challenges such as relying on sufficient labels, requiring manual tuning, and inflexibility for multi-tasks. We discuss potential solutions to some of the issues as follows.

5.1. Dynamic trust in recommender systems

In the real world cases, the trust information is changing over time. For example, trust friend relationships may change; we may have different groups of friends during different periods. Also, the preference for friendship may change over time [118]. Another example is for malicious users. We know that the malicious reviews or ratings may affect common users, but not all the review contents written by malicious reviewers are for sure the fake reviews [8]. Most work assumes that the reviews/ratings made by malicious users are fake samples; however, limited work considers solving the problem case by case because it is hard to get all the labels. Another challenge is that the dynamic system may be time-consuming for updating the whole system over time. Potential solutions include utilizing recent advances in session-based recommender systems and sequential recommender systems [152]. For example, the sequential models such as recurrent neural networks can store the trust-aware information and dynamically update itself with the changes. Memory neural networks [153,154] can also be applied to externally store the trust information.

5.2. Embedding for trust-aware recommender systems

Embedding methods, which include node embedding, sequential embedding, and graph embedding, are widely applied to recommender systems. For example, several graph embedding methods learn social relationship as a type of graph information. Such embedding methods include node2vec [58], Euclidean embedding [155], UniWalk (explainable) [156], deepWalk [72], and the recent model, graph neural networks [157]. Effective embedding has achieved significant improvement in recommender systems, including various application domains [158]. In trust-aware recommender systems, how to represent the trust issues and how to combine them with other information are two key points for constructing an effective trust-aware recommender system. The learned embedding vector is normally used for representing users/items or further combined with other representations. Some may construct the graph for items to model the user behaviors: a node in the graph represents each item, and an edge denotes each co-occurrence of items [158]. Some may represent the user by their trust relationships and then combine such representation with user rating behaviors for recommendation [36,37]. Most related work has the limitation of ignoring the inner interactions between different types of information. Intuitively, a user's decision is affected by many factors. Friendship, product reviews (especially malicious reviews), and product descriptions can all affect our decisions. Thus, the latent representation for the user should be better not simply the concatenation of representations in different domains but also a unified factor.

5.3. Meta-learning for trust-aware recommender systems

For the recommendation problem, there is no standard base model for dealing with different tasks. Like the aforementioned three trust-aware tasks, limited work combines different ideas, i.e., cover all the bases, for recommendation problems. Applying meta-learning or learning to learn may cover this limitation. Meta-learning, or learning to learn, is the science of observing the performance of different model configurations on various tasks, and then learning from the history observation, to guide the new tasks. This will improve the efficiency of the new task modeling and enable the model with automatic learning capabilities. This is an inspiring area since most current works are in hand-engineered way [159]. Most deep models only perform well on one task or a single dataset. This means we may cost a lot of manual efforts on designing the models instead of solving the problems. Thus, it is meaningful to design a module to let the machine learns itself by supporting the related information. Recent works [130,160,161] presented initial attempts in incorporating meta-learning approaches into the recommender systems for cold-start problems, while there lacks of discussion about few-shot learning in trust issues for improving the recommendation performance.

5.4. Blockchain for decentralized trust management

Current recommender systems are built upon the data from web users – which contain both normal users and malicious users – and thus become vulnerable to real-world frauds. For example, with the increasing number of fraudulent ratings or feedbacks, the truth for a recommender system will deviate from the genuine truth. A major reason for fraudulent behaviors is the easiness of getting publicly available information. With learning from normal users, fraudulent users can hide their intention and cheat the detection techniques, which will affect the robustness of recommender systems. One potential solution is leveraging the blockchain idea for trust management in recommender systems. Blockchain [162,163] is a shared ledger technology, and each participant shares a common view of truth. It uses a decentralized peer-to-peer network to manage the data, which can eliminate the potential risks of centrally stored data, and all validated activities are permanently recorded. Each participant can get access to their data, and even a system administrator cannot delete the records. This secures each transaction and thus eliminates the human error or fraud. Besides, most of the deep learning-based recommendation techniques are data-hungry and are centralized in computing. A recent idea is distributing the learning tasks of a deep learning method on blockchain, which can improve both efficiency and privacy [164].

6. Conclusion

In this survey, we investigate three aspects of trust in recommender systems: social-awareness, robustness, and explainability, with a focus on deep learning-based on recommender systems. We describe how deep learning methods work for the trust-aware recommendation in representation learning, predictive learning, and generative learning. We notice that the growing research in deep learning has improved the performance of recommender systems in various tasks. Meanwhile, current research still faces severe challenges in adapting to labeled data, reducing tuning efforts, and enhancing flexibility in handling multiple tasks. We hope this survey could give readers a comprehensive understanding of state-of-the-art studies in the deep-learning-based recommendation and inspire more insights and contributions to this vibrant research domain.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. (2019).

[2] G. Guo, J. Zhang, N. Yorke-Smith, TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in: AAAI, 2015.

[3] Y. Pan, F. He, H. Yu, Trust-aware collaborative denoising auto-encoder for top-n recommendation, 2017, arXiv preprint arXiv:1703.01760.

[4] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, R. Ghosh, Exploiting burstiness in reviews for review spammer detection, in: AAAI, 2013.

[5] J. Zou, F. Fekri, A belief propagation approach for detecting shilling attacks in collaborative filtering, in: CIKM, ACM, 2013.

[6] Z. Yang, L. Xu, Z. Cai, Z. Xu, Re-scale AdaBoost for attack detection in collaborative filtering recommender systems, Knowl.-Based Syst. (2016).

[7] R. Bhaumik, B. Mobasher, R. Burke, A clustering approach to unsupervised attack detection in collaborative recommender systems, in: ICDM, IEEE, 2011.

[8] M. Si, Q. Li, Shilling attacks against collaborative recommender systems: a review, Artif. Intell. Rev. (2020).

[9] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, F. Zhang, A hybrid collaborative filtering model with deep structure for recommender systems, in: AAAI, 2017.

[10] X. Yang, Y. Guo, Y. Liu, H. Steck, A survey of collaborative filtering based social recommender systems, Comput. Commun. (2014).

[11] I. Gunes, C. Kaleli, A. Bilge, H. Polat, Shilling attacks against recommender systems: a comprehensive survey, Artif. Intell. Rev. (2014).

[12] S. Gupta, S. Nagpal, Trust aware recommender systems: a survey on implicit trust generation techniques, Int. J. Comput. Sci. Inf. Technol. (2015).

[13] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, 2020, arXiv preprint arXiv:1804.11192.

[14] W.X. Zhao, S. Li, Y. He, L. Wang, J.-R. Wen, X. Li, Exploring demographic information in social media for product recommendation, Knowl. Inf. Syst. (2016).

[15] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: SIGIR, ACM, 2014.

[16] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, M. De Rijke, Explainable outfit recommendation with joint outfit matching and comment generation, TKDE (2019).

[17] B. Wang, M. Ester, J. Bu, D. Cai, Who also likes it? generating the most persuasive social explanations in recommender systems, in: AAAI, 2014.

[18] X. Chen, Z. Qin, Y. Zhang, T. Xu, Learning to rank features for recommendation over multiple categories, in: SIGIR, ACM, 2016.

[19] R. Heckel, M. Vlachos, T. Parnell, C. Dünner, Scalable and interpretable product recommendations via overlapping co-clustering, in: ICDE, IEEE, 2017.

[20] Z. Ren, S. Liang, P. Li, S. Wang, M. de Rijke, Social collaborative viewpoint regression with explainable recommendations, in: WSDM, ACM, 2017.

[21] S. Seo, J. Huang, H. Yang, Y. Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in: ACM Conference on Recommender Systems, ACM, 2017.

[22] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al., The YouTube video recommendation system, in: ACM Conference on Recommender Systems, 2010.

[23] J. Lu, Q. Zhang, G. Zhang, Recommender Systems: Advanced Developments, World Scientific, 2020.

[24] F. Monti, M. Bronstein, X. Bresson, Geometric matrix completion with recurrent multi-graph neural networks, in: NeurIPS, 2017.

[25] Q. Zhang, W. Liao, G. Zhang, B. Yuan, J. Lu, A deep dual adversarial network for cross-domain recommendation, IEEE Trans. Knowl. Data Eng. (2021).

[26] W. Liao, Q. Zhang, B. Yuan, G. Zhang, J. Lu, Heterogeneous multidomain recommender system through adversarial learning, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[27] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, 2019, arXiv preprint arXiv:1902.07243.

[28] M. Wang, Z. Wu, X. Sun, G. Feng, B. Zhang, Trust-aware collaborative filtering with a denoising autoencoder, Neural Process. Lett. (2019).

[29] L. Wu, P. Sun, R. Hong, Y. Ge, M. Wang, Collaborative neural social recommendation, IEEE Trans. Syst. Man Cybern.: Syst. (2018).

[30] S. Deng, L. Huang, G. Xu, X. Wu, Z. Wu, On deep learning for trust-aware recommendations in social networks, IEEE Trans. Neural Netw. Learn. Syst. (2016).

[31] C. Nisha, A. Mohan, A social recommender system using deep architecture and network embedding, Appl. Intell. (2019).

[32] D. Rafailidis, F. Crestani, Recommendation with social relationships via deep learning, in: SIGIR, ACM, 2017.

[33] P. Sun, L. Wu, M. Wang, Attentive recurrent social recommendation, in: SIGIR, ACM, 2018.

[34] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: WSDM, ACM, 2019.

[35] L. Wu, H.-F. Yu, N. Rao, J. Sharpnack, C.-J. Hsieh, Graph DNA: Deep neighborhood aware graph encoding for collaborative filtering, 2019, arXiv preprint arXiv:1905.12217.

[36] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, G. Chen, Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems, 2019, arXiv preprint arXiv:1903.10433.

[37] L. Wu, P. Sun, R. Hong, Y. Fu, X. Wang, M. Wang, SocialGCN: An efficient graph convolutional network based model for social recommendation, 2018, arXiv preprint arXiv:1811.02815.

[38] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: KDD, ACM, 2018.

[39] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, Q. Li, Deep adversarial social recommendation, 2019, arXiv preprint arXiv:1905.13160.

[40] T. Xiao, H. Tian, H. Shen, Variational deep collaborative matrix factorization for social recommendation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2019.

[41] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, M. Wang, A hierarchical attention model for social contextual image recommendation, TKDE (2019).

[42] H. Bao, L. Wu, P. Sun, Contextual attention model for social recommendation, in: Pacific Rim Conference on Multimedia, Springer, 2018.

[43] C.H. Liu, J. Xu, J. Tang, J. Crowcroft, Social-aware sequential modeling of user interests: A deep learning approach, TKDE (2018).

[44] J. Gao, T. Zhang, C. Xu, A unified personalized video recommendation via dynamic recurrent neural networks, in: ACM International Conference on Multimedia, 2017.

[45] X. Geng, H. Zhang, J. Bian, T.-S. Chua, Learning image and user features for recommendation in social networks, in: ICCV, IEEE, 2015.

[46] Z. Zhao, Q. Yang, H. Lu, T. Weninger, D. Cai, X. He, Y. Zhuang, Social-aware movie recommendation via multimodal network learning, IEEE Trans. Multimed. (2017).

[47] S. Liu, P. Cui, W. Zhu, S. Yang, Learning socially embedded visual representation from scratch, in: ACM International Conference on Multimedia, 2015.

[48] C. Chen, M. Zhang, Y. Liu, S. Ma, Social attentional memory network: Modeling aspect-and friend-level differences in recommendation, in: WSDM, ACM, 2019.

[49] D. Rafailidis, G. Weiss, A neural attention model for adaptive learning of social friends' preferences, 2019, arXiv preprint arXiv:1907.01644.

[50] L. Xiao, Z. Min, L. Yiqun, M. Shaoping, A neural network model for social-aware recommendation, in: Asia Information Retrieval Symposium, Springer, 2017.

[51] W. Fan, Q. Li, M. Cheng, Deep modeling of social relations for recommendation, in: AAAI, 2018.

[52] X. Wang, X. He, L. Nie, T.-S. Chua, Item silk road: Recommending items from information domains to social users, in: SIGIR, ACM, 2017.

[53] Y. Wen, L. Guo, Z. Chen, J. Ma, Network embedding based recommendation method in social networks, in: Conference on World Wide Web, 2018.

[54] C.-Y. Liu, C. Zhou, J. Wu, Y. Hu, L. Guo, Social recommendation with an essential preference space, in: AAAI, 2018.

[55] M. Jamali, M. Ester, Using a trust network to improve top-n recommendation, in: ACM Conference on Recommender Systems, ACM, 2009.

[56] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: KDD, ACM, 2009.

[57] C. Zhang, L. Yu, Y. Wang, C. Shah, X. Zhang, Collaborative user network embedding for social recommender systems, in: SIAM International Conference on Data Mining (SDM), 2017.

[58] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: KDD, ACM, 2016.

[59] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: KDD, ACM, 2008.

[60] H. Zhang, G. Liu, J. Wu, Social collaborative filtering ensemble, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2018.

[61] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. (2010).

[62] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: AAAI, 2015.

[63] P.H. Pinheiro, R. Collobert, Recurrent convolutional neural networks for scene labeling, in: ICML, 2014.

[64] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, J. Zhu, Personal recommendation using deep recurrent neural networks in NetEase, in: ICDE, IEEE, 2016.

[65] T. Donkers, B. Loepp, J. Ziegler, Sequential user-based recurrent neural network recommendations, in: ACM Conference on Recommender Systems, 2017.
[66] C.-Y. Wu, A. Ahmed, A. Beutel, A.J. Smola, H. Jing, Recurrent recommender networks, in: WSDM, ACM, 2017.
[67] C.-Y. Wu, A. Ahmed, A. Beutel, A.J. Smola, Joint training of ratings and reviews with recurrent recommender networks, 2016.
[68] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: NeurIPS, 2014.
[69] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
[70] Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, Q. Xiong, A minimax game for generative and discriminative sample models for recommendation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2019.
[71] G. Karamanolakis, K.R. Cherian, A.R. Narayan, J. Yuan, D. Tang, T. Jebara, Item recommendation with variational autoencoders and heterogeneous priors, in: Workshop on Deep Learning for Recommender Systems, ACM, 2018.
[72] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: KDD, ACM, 2014.
[73] M.-T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv preprint arXiv:1508.04025.
[74] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: ICML, 2015.
[75] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: NeurIPS, 2008.
[76] A.M. Elkahky, Y. Song, X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in: WWW Conference, 2015.
[77] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature (2015).
[78] S. Alonso, J. Bobadilla, F. Ortega, R. Moya, Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems, IEEE Access (2019).
[79] F. Zhang, Y. Lu, J. Chen, S. Liu, Z. Ling, Robust collaborative filtering based on non-negative matrix factorization and R1-norm, Knowl.-Based Syst. (2017).
[80] H. Yu, R. Gao, K. Wang, F. Zhang, A novel robust recommendation method based on kernel matrix factorization, J. Intell. Fuzzy Systems (2017).
[81] X. He, Z. He, X. Du, T.-S. Chua, Adversarial personalized ranking for recommendation, in: SIGIR, 2018.
[82] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, 2019, arXiv preprint arXiv:1901.03407.
[83] Y. Ren, D. Ji, Learning to detect deceptive opinion spam: A survey, IEEE Access (2019).
[84] S. Zhao, Z. Xu, L. Liu, M. Guo, Towards accurate deceptive opinion spam detection based on word order-preserving CNN, 2017, arXiv preprint arXiv:1711.09181.
[85] W. Zhang, Y. Du, T. Yoshida, Q. Wang, DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network, Inf. Process. Manage. (2018).
[86] L. Li, W. Ren, B. Qin, T. Liu, Learning document representation for deceptive opinion spam detection, in: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer, 2015.
[87] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.
[88] X. Wang, K. Liu, J. Zhao, Detecting deceptive review spam via attention-based neural networks, in: National CCF Conference on Natural Language Processing and Chinese Computing, Springer, 2017.
[89] X. Wang, K. Liu, J. Zhao, Handling cold-start problem in review spam detection by jointly embedding texts and behaviors, in: ACL, 2017.
[90] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
[91] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. (1997).
[92] Y. Ren, Y. Zhang, Deceptive opinion spam detection using neural network, in: Proceedings of COLING, 2016.
[93] C.-C. Wang, M.-Y. Day, C.-C. Chen, J.-W. Liou, Detecting spamming reviews using long short-term memory recurrent neural network framework, in: International Conference on E-Commerce, E-Business and E-Government, ACM, 2018.
[94] X. Wang, K. Liu, S. He, J. Zhao, Learning to represent review with tensor decomposition for spam detection, in: Conference on Empirical Methods in Natural Language Processing, 2016.
[95] M. Dong, L. Yao, X. Wang, B. Benatallah, C. Huang, X. Ning, Opinion fraud detection via neural autoencoder decision forest, Pattern Recognit. Lett. (2018).
[96] H. Aghakhani, A. Machiry, S. Nilizadeh, C. Kruegel, G. Vigna, Detecting deceptive reviews using generative adversarial networks, in: IEEE Security and Privacy Workshops (SPW), 2018.
[97] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: WSDM, ACM, 2016.
[98] F. Strub, R. Gaudel, J. Mary, Hybrid recommender system based on autoencoders, in: Workshop on Deep Learning for Recommender Systems, ACM, 2016.
[99] H. Wang, S. Xingjian, D.-Y. Yeung, Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks, in: NeurIPS, 2016.
[100] F. Yuan, L. Yao, B. Benatallah, Adversarial collaborative neural network for robust recommendation, in: SIGIR, ACM, 2019.
[101] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, Z. Huang, Neural memory streaming recommender networks with adversarial training, in: KDD, ACM, 2018.
[102] Y. Jhamb, T. Ebesu, Y. Fang, Attentive contextual denoising autoencoder for recommendation, in: SIGIR, ACM, 2018.
[103] C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen, J. Gao, ATRank: An attention-based user behavior modeling framework for recommendation, in: AAAI, 2018.
[104] P. Loyola, C. Liu, Y. Hirate, Modeling user session and intent with an attention-based encoder-decoder architecture, in: ACM Conference on Recommender Systems, 2017.
[105] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, J. Wu, Sequential recommender system based on hierarchical attention networks, in: IJCAI, 2018.
[106] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, STAMP: short-term attention/memory priority model for session-based recommendation, in: KDD, ACM, 2018.
[107] S. Seo, J. Huang, H. Yang, Y. Liu, Representation learning of users and items for review rating prediction using attention-based convolutional neural network, in: 3rd International Workshop on Machine Learning Methods for Recommender Systems (MLRec)(SDM'17), 2017.
[108] Y. Tay, L. Anh Tuan, S.C. Hui, Latent relational metric learning via memory-based attention for collaborative ranking, in: WWW Conference, 2018.
[109] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T.-S. Chua, Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention, in: SIGIR, ACM, 2017.
[110] N. Tintarev, J. Masthoff, Designing and evaluating explanations for recommender systems, in: Recommender Systems Handbook, Springer, 2011.
[111] A. Sharma, D. Cosley, Do social explanations work?: studying and modeling the effects of social explanations in recommender systems, in: WWW Conference, 2013.
[112] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: WWW Conference, 2018.
[113] X. Chen, Y. Zhang, H. Xu, Y. Cao, Z. Qin, H. Zha, Visually explainable recommendation, 2018, arXiv preprint arXiv:1801.10288.
[114] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, X. Xie, A reinforcement learning framework for explainable recommendation, in: ICDM, IEEE, 2018.
[115] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, R. Mehrotra, Explore, exploit, and explain: personalizing explainable recommendations with bandits, in: ACM Conference on Recommender Systems, 2018.
[116] B. Abdollahi, O. Nasraoui, Explainable restricted boltzmann machines for collaborative filtering, 2016, arXiv preprint arXiv:1606.07129.
[117] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, X. Luo, A context-aware user-item representation learning for item recommendation, Trans. Inf. Syst. (2019).
[118] X. Chen, Y. Zhang, Z. Qin, Dynamic explainable recommendation based on neural attentive models, in: AAAI, 2019.
[119] M. Hou, L. Wu, E. Chen, Z. Li, V.W. Zheng, Q. Liu, Explainable fashion recommendation: A Semantic Attribute Region guided approach, 2019, arXiv preprint arXiv:1905.12862.
[120] P. Li, Z. Wang, Z. Ren, L. Bing, W. Lam, Neural rating regression with abstractive tips generation for recommendation, in: SIGIR, ACM, 2017.
[121] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: WSDM, ACM, 2018.
[122] D. Cong, Y. Zhao, B. Qin, Y. Han, M. Zhang, A. Liu, N. Chen, Hierarchical attention based neural network for explainable recommendation, in: International Conference on Multimedia Retrieval, ACM, 2019.
[123] F. Costa, S. Ouyang, P. Dolog, A. Lawlor, Automatic generation of natural language explanations, in: International Conference on Intelligent User Interfaces Companion, ACM, 2018.
[124] Y. Lu, R. Dong, B. Smyth, Why I like it: multi-task learning for recommendation and explanation, in: ACM Conference on Recommender Systems, 2018.
[125] S. Ouyang, A. Lawlor, F. Costa, P. Dolog, Improving explainable recommendations with synthetic reviews, 2018, arXiv preprint arXiv:1807.06978.

[126] G. Zhao, H. Fu, R. Song, T. Sakai, X. Xie, X. Qian, Why you should listen to this song: Reason generation for explainable recommendation, in: International Conference on Data Mining Workshops, IEEE, 2018.
[127] T. Suzuki, S. Oyama, M. Kurihara, Toward explainable recommendations: Generating review text from multicriteria evaluation data, in: International Conference on Big Data, IEEE, 2018.
[128] W.-C. Kang, C. Fang, Z. Wang, J. McAuley, Visually-aware fashion recommendation and design with generative image models, in: ICDM, IEEE, 2017.
[129] S. Kumar, M.D. Gupta, cGAN: complementary fashion item recommendation, 2019, arXiv preprint arXiv:1906.05596.
[130] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, H. Zha, Sequential recommendation with user memory networks, in: WSDM, ACM, 2018.
[131] J. Lin, Y. Liu, A neural network based explainable recommender system, 2018, arXiv preprint arXiv:1812.11740.
[132] H. Bharadhwaj, Layer-wise relevance propagation for explainable recommendations, 2018, arXiv preprint arXiv:1807.06160.
[133] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: WSDM, ACM, 2017.
[134] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, in: Eleventh Annual Conference of the International Speech Communication Association, 2010.
[135] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: ICDM, 2008.
[136] R. He, J. McAuley, VBPR: visual bayesian personalized ranking from implicit feedback, in: AAAI, 2016.
[137] Q. Liu, S. Wu, L. Wang, DeepStyle: Learning user preferences for visual recommendation, in: SIGIR, ACM, 2017.
[138] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, H. Liu, What your images reveal: Exploiting visual contents for point-of-interest recommendation, in: WWW Conference, 2017.
[139] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, Z. Qin, Aesthetic-based clothing recommendation, in: WWW Conference, 2018.
[140] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, IEEE, 2016.
[141] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Conference on Uncertainty in Artificial Intelligence, 2009.
[142] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: ICCV, IEEE, 2017.
[143] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: CVPR, IEEE, 2006.
[144] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.
[145] M.S. Sajjadi, G. Parascandolo, A. Mehrjou, B. Schölkopf, Tempered adversarial networks, 2018, arXiv preprint arXiv:1802.04374.
[146] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One (2015).
[147] L. Gatys, A.S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, in: Advances in Neural Information Processing Systems, 2015, pp. 262–270.
[148] J. Vinagre, A.M. Jorge, J. Gama, An overview on the exploitation of time in collaborative filtering, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. (2015).
[149] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: WWW Conference, 2017.
[150] E. Kiperwasser, Y. Goldberg, Simple and accurate dependency parsing using bidirectional LSTM feature representations, ACL (2016).
[151] Y. Goldberg, J. Nivre, A dynamic oracle for arc-eager dependency parsing, in: Proceedings of COLING, 2012.
[152] S. Wang, L. Cao, Y. Wang, Q.Z. Sheng, M.A. Orgun, D. Lian, A survey on session-based recommender systems, ACM Comput. Surv. 54 (7) (2021) 1–38.
[153] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: International Conference on Machine Learning, PMLR, 2016, pp. 1842–1850.
[154] M. Dong, F. Yuan, L. Yao, X. Xu, L. Zhu, Mamo: Memory-augmented meta-optimization for cold-start recommendation, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 688–697.
[155] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, R. Jia, T. Dou, Social recommendation using Euclidean embedding, in: IJCNN, IEEE, 2017.
[156] H. Park, H. Jeon, J. Kim, B. Ahn, U. Kang, Uniwalk: Explainable and accurate recommendation for rating and network data, 2017, arXiv preprint arXiv:1710.07134.
[157] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: NeurIPS, 2016.
[158] Y. Ouyang, B. Guo, X. Tang, X. He, J. Xiong, Z. Yu, Learning cross-domain representation with multi-graph neural network, 2019, arXiv preprint arXiv:1905.10095.
[159] J. Vanschoren, Meta-learning: A survey, 2018, arXiv preprint arXiv:1810.03548.
[160] Z. Du, X. Wang, H. Yang, J. Zhou, J. Tang, Sequential scenario-specific meta learner for online recommendation, 2019, arXiv preprint arXiv:1906.00391.
[161] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, H. Larochelle, A meta-learning perspective on cold-start recommendations for items, in: NeurIPS, 2017.
[162] L. Yao, X. Wang, Q.Z. Sheng, S. Dustdar, S. Zhang, Recommendations on the internet of things: Requirements, challenges, and directions, Internet Comput. (2019).
[163] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, et al., Blockchain technology: Beyond bitcoin, Appl. Innov. (2016).
[164] J.-S. Weng, J. Weng, M. Li, Y. Zhang, W. Luo, DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive, IACR Cryptol. ePrint Arch. (2018).