Arabian Journal for Science and Engineering (2021) 46:3247–3286
https://doi.org/10.1007/s13369-020-05112-2

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Research on Understanding the Effect of Deep Learning on User Preferences

Garima Gupta[1] · **Rahul Katarya**[1]

Received: 25 May 2020 / Accepted: 2 November 2020 / Published online: 26 November 2020

Abstract
Recommender systems are becoming more essential than ever as the data available online is increasing manifold. The increasing data presents us with an opportunity to build complex systems that can model the user interactions more accurately and extract sophisticated features to provide recommendations with better accuracy. To construct these complex models, deep learning is emerging as one of the most powerful tools. It can process large amounts of data to learn the structure and patterns that can be exploited. It has been used in recommender systems to solve cold-start problem, better estimate the interaction functions, and extract deep feature representations, among other facets that plague the traditional recommender systems. As big data is becoming more prevalent, there is a need to use tools that can take advantage of such explosive data. An extensive study on recommender systems using deep learning has been performed in the paper. The literature review spans in-depth analysis and comparative study of the research domain. The paper exhibits a vast range of scope for efficient recommender systems in future.

Keywords Recommender systems · Machine learning · Deep learning

1 Introduction

A recommender system's goal is to present items to a user, which are most likely to lead to conversion. Conversion might relate to different things in different contexts. For example, for e-commerce, it might mean purchasing the product, and for Netflix, it might mean viewing the content. To achieve this goal, recommender systems have to study the underlying data, consisting of items, users, and their interactions. To study these interactions, we need to extract relevant features and create a system that can learn and model such interactions.

Based on the properties of the network that the system exploits, recommender systems can be categorized into content-based, collaborative filtering-based, and hybrid systems. Recommender systems based on content exploit the item's features to find similar items that the user might like. A collaborative filtering recommender system uses the user–user interaction. The philosophy being user similar to each other might have similar choices. Systems that combine both these aspects in one or the other way are termed as hybrid systems.

Recommender systems use machine learning capabilities at its core to learn the interaction functions and predict items that are most likely to lead to conversion. As the data present online is increasing every day, and the concept of big data is harnessing attention, the dimensionality and modality of data are growing explosively. Hence, there arises a need for a tool that can exploit big data to provide better results. Deep learning is essentially an extension of machine learning that is specifically designed to manage big data. Deep learning algorithms are based on large neural networks with multiple hidden layers. These models can exploit data to learn complex interactions and feature representations. This can help us improve the accuracy of recommendations and bestow new meaning to tackle problems like cold start that plagues traditional recommender systems.

Because of the many advantages, there is a sudden boom in implementing deep learning in RS. For this reason, ACM has been holding a workshop DLRS (deep learning in recommender systems) with RecSys since the year 2016. Several

✉ Rahul Katarya
rahulkatarya@dtu.ac.in

Garima Gupta
garimacsdtu@gmail.com

[1] Department of Computer Science and Engineering, Delhi Technological University, Delhi 110042, India

deep learning techniques are implemented in recommender systems, as discussed in the paper. Some papers have used only a single technique, while others have used an amalgamation of different techniques to improve the results. Deep learning is a powerful tool and is being used to solve a plethora of problems in recommender systems. It has also proved to be effective in improving accuracy in various domains. In this paper, the research work done in RS using deep learning techniques has been explored. With the advancement in big data, practitioners will need better tools to tackle the complexities that come with it, and at the same time, exploit it for better results.

To make useful recommendations, there are various aspects to be kept in mind like the application domain, type of dataset, problem statement, kind of recommender system used, type of deep learning technique used, the side information such as gender of the user, user's login sessions, text semantics, and much more.

This paper will report the following research questions restricted to deep learning in recommender systems, and the answers to these questions have been provided in Sect. 3:

.

RQ 1	Which application domains have been included in the study?
RQ 2	What are the different pre-defined and self-generated datasets other authors have worked on in the past?
RQ 3	What types of recommender systems have been studied and implemented in past studies?
RQ 4	Which deep learning techniques have been implemented in the previous research?
RQ 5	Which metrics have been used to analyze the results?

The paper has been organized as follows. Section 2 furnishes the background study of the literature survey, i.e., a short description of the recommender systems, machine learning, and deep learning. Section 3 is the Research Methodology in which the readers can understand the division of research papers studied for this survey paper. Here, the papers have been classified based on various parameters. Section 4 is the Literature Review, which consists of a detailed study of the research work and encompasses the conclusion and the research gaps observed in the mentioned field of study. The paper concludes with References.

This paper is written to explain the application of deep learning in recommender systems and discuss how recommender systems can be improved and extended by applying different deep learning techniques. To perform this survey, we conducted an extensive study of previous research done in recommender systems, employing deep learning. We surveyed top conferences and journals in deep learning in recommender systems and collated the papers relevant to our research area. The original contribution of the survey is presented in Sect. 4 focusing on a systematic flow of information. First, we explained the basic concept of deep learning models. Afterward, we explored its applications in recommender systems and provided detailed analysis of key recommender systems using those techniques.

2 Background

This section expounds on the major concepts introduced in the paper, namely recommender systems, machine learning, and deep learning.

2.1 Recommender Systems

Recommender systems use machine learning and artificial intelligence algorithms to predict items to users. In today's world, the availability of such a vast pool of data makes it difficult for e-commerce websites to identify user-centric items [1]. If a user intends to select an item, he might not be able to find the item most suitable to him due to the presence of multiple options. To make this task easy, the concept of recommender systems was introduced. RS uses machine learning and artificial intelligence techniques to recommend relevant items to the user.

2.1.1 Content-Based Filtering

In content-based filtering, the recommendations are performed based on the previous choices of the user. As the name suggests, this type of filtering technique depends upon the various parameters of the user items. These parameters, when taken into account, reveal the granular level information about the user's preferences. Content-based filtering aims to exploit such a detailed analysis of the user's liking and recommends items in the present. The recommender system studies the choices user made in the past in different domains and depending upon those choices, the user is recommended items [2]. Such choices made by the user depend on various parameters or features of the items. Since these recommendations match the preferences the user made in the past, hence they are more user-centric. To explain with the help of an example of content-based recommendations, let us say that a user likes to watch movies by the director *Quentin Tarantino*, so the movies which will be recommended to the user will feature the same director. This was content-based filtering based on just one feature. Similarly, we can take multiple features to determine the recommendation list. Mathemati-

cally, content-based RS for multiple users can be represented using Eq. 1:

$$\min_{\theta^1,\ldots,\theta^n} \frac{1}{2}\sum_{j=1}^{n}\sum_{i:r(i,j)=1}\left(\left(\theta^j\right)^T x^i - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\theta_k^j\right)^2. \quad (1)$$

Here, θ^j represents the parameter vector for user *j*, *r(i,j)= 1* if user *j* has rated the movie *i*, else 0, x^i represents the feature vector for movie *i*, and $y^{(i,j)}$ represents the rating for movie *i* by user *j*.

2.1.2 Collaborative Filtering

The second type of recommender system is collaborative filtering-based Recommender System. In this type of filtering, the recommendations are affected by the neighbors of the user. It is seen that if a user's friend buys a particular product, the user is likely to buy the same or a similar product. Based on this principle, collaborative filtering adopts a neighborhood-based approach. In this technique, the user's neighbors' choices are studied, and based on those choices, similar items are recommended to the user. User-based collaborative filtering is essentially a neighborhood-based approach. To explain with an example of how collaborative filtering works, let us say that *user a* has a neighbor *user b. User b* likes a particular product, say *product p*. Now in this filtering technique, the preference of *user a* will be subjected to the preferences of *user b* and vice versa. Hence, *user a* will be recommended *product p* and products similar to *product p*.

Collaborative filtering is further of two types, namely, item-based collaborative filtering and user-based collaborative filtering.

a) Item-based Collaborative Filtering

In this type of collaborative filtering technique, the recommendations are based on the similarity between the items. If a user purchases *product p* and *product q* is similar to *product p* in one or more ways, i.e., if they share some common features, the user is recommended *product q* and other products similar to *product q*. The similarity between the items is calculated using several similarity measures. It can be *cosine-based similarity,* which can be calculated using Eq. 2. It is also known as *vector-based similarity,* and the similarity is determined by calculating the cosine of the angle between the two vectors.

$$\text{sim}(a,b) = \cos\left(\vec{a},\vec{b}\right) = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\| * \|\vec{b}\|}. \quad (2)$$

Here, $\vec{a}$ and $\vec{b}$ are the two item vectors,

Another similarity measure that can be used to deduce the similarity between different items is *Pearson correlation-based similarity.* It can be given by Eq. 3.

$$\text{sim}(a,b) = \frac{\Sigma_{u\varepsilon U}\left(R_{u,a}-\bar{R}_a\right)\left(R_{u,b}-\bar{R}_b\right)}{\sqrt{\Sigma_{u\varepsilon U}\left(R_{u,a}-\bar{R}_a\right)^2}\sqrt{\Sigma_{u\varepsilon U}\left(R_{u,b}-\bar{R}_b\right)^2}}. \quad (3)$$

Here, $R_{u,a}$ denotes the rating given by *user u* for *item a*, $\bar{R}_a$ denotes the average rating of *item a.*

Similarly, $R_{u,b}$ denotes the rating given by *user u* for *item b*, and $\bar{R}_b$ denotes the average rating of *item b.*

b) User-Based Collaborative Filtering

In this type of collaborative filtering technique, the recommendations take place depending upon the preferences of the user's neighbors. If *user u* has a neighbor *user v*, and *user v* likes *product p*, then *user u* will be recommended *product p* and all such products preferred by *user v*.

2.1.3 Hybrid Filtering

The third approach is the hybrid approach. It is an amalgamation of content-based filtering and collaborative filtering. Here, first, the social network analysis of a user's neighborhood takes place. This step helps identify the neighbors' preferences, and items are kept in the account to be recommended to the user. In the second step, the user's history is studied, and depending on the items he bought previously and items having similar content or attributes as the past items; item recommendations are taken into account [3]. Finally, analyzing the results obtained in both the steps, the user is recommended the items. Authors in [4] differentiated user's preferences in generated recommendations by using a deep hybrid recommender system. In order to explain the underlying literature of hybrid filtering, let us take the example used in the previous Sect. 2.1.2. Let us say that *product p* has a similar product, *product q,* along with one or several features. The system will apply content-based filtering to identify products similar to *product p* and *product q*. Upon selecting such products, say *list l*, the system will use collaborative filtering, and *user b* will now be recommended *product p*, *product q*, and other similar products, i.e., products from *list l.*

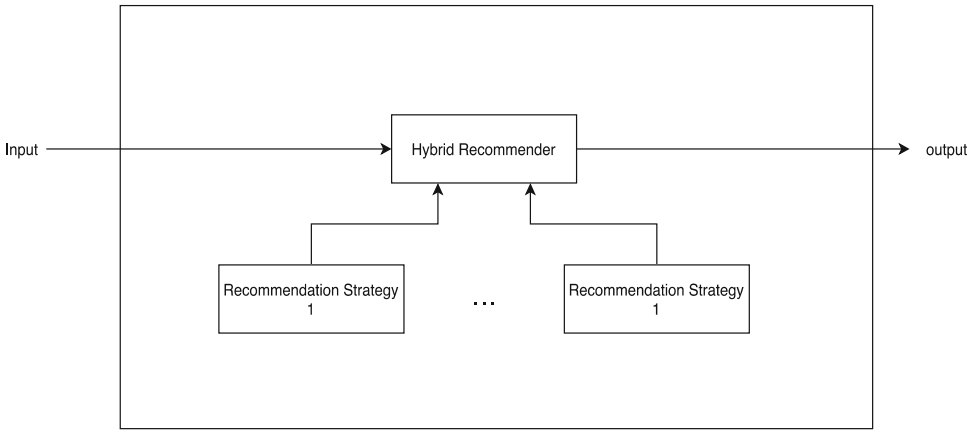

Fig. 1 Monolithic hybrid design

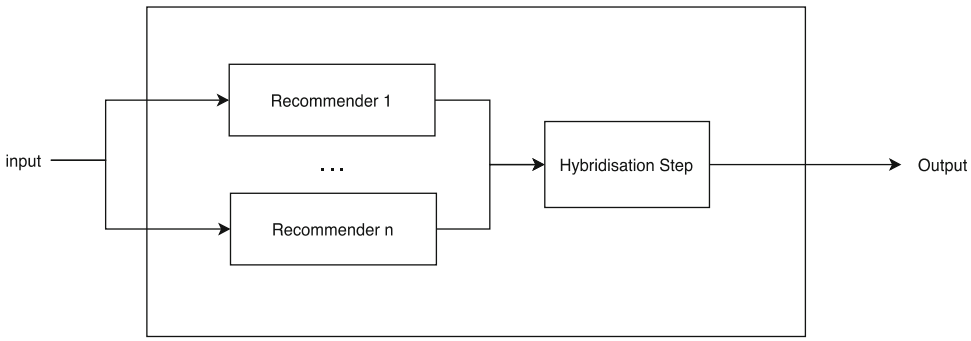

Fig. 2 Architecture of parallelized hybrid design

Any of the three techniques can be used to recommend items to users. Depending upon the chosen technique, the recommendations are made to the user. Hybrid recommender systems can be categorized into the following types:

a) Monolithic Hybrid Design

In this type, there exists a single recommendation component that aggregates multiple recommendation methods by preprocessing and combining numerous knowledge sources. The architecture of monolithic hybrid design is represented in Fig. 1.

b) Parallelized Hybrid Design

In this type, several recommender systems are run in parallel, and the output of each is combined at the later stages using an aggregation mechanism. These are further of three types, i.e., *mixed, weighted,* and *switching*. The architecture of parallelized hybrid design is shown in Fig. 2.

c) Pipelined Hybrid Design

In this type, every recommender system processes the input pertaining to its recommendation mechanism. The output, hence produced, is forwarded as input to subsequent recommender systems mimicking a staged process. The architecture of the pipelined hybrid design is given in Fig. 3.

2.2 Machine Learning

Machine learning is a section in computer science that deals with computing and solving problems intelligently and analyzing the results. The system is presented with a dataset, and various machine learning algorithms can be applied to that dataset to obtain results. Due to the availability of such extensive data and the need to get predictive results, machine learning algorithms are used widely. There exist three types of machine learning techniques, which can be seen in Fig. 4.

2.2.1 Supervised Learning

In supervised machine learning algorithms, the system is presented with the dataset, and the outcome is pre-defined. For example, it is known that the outcome has to be whether a given transaction is fraudulent or genuine for identifying fraudulent bank transactions. Since the result is known beforehand, the algorithm works to guide the expected result [5]. Hence, the entire working is supervised. A few supervised learning algorithms are Logistic Regression, Neural Network, Naïve Bayes, Decision Trees, Nearest Neighbors, and others.

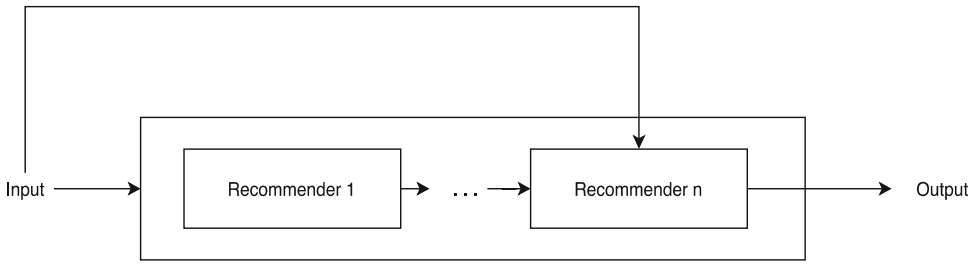

Fig. 3 Architecture of pipelined hybrid design

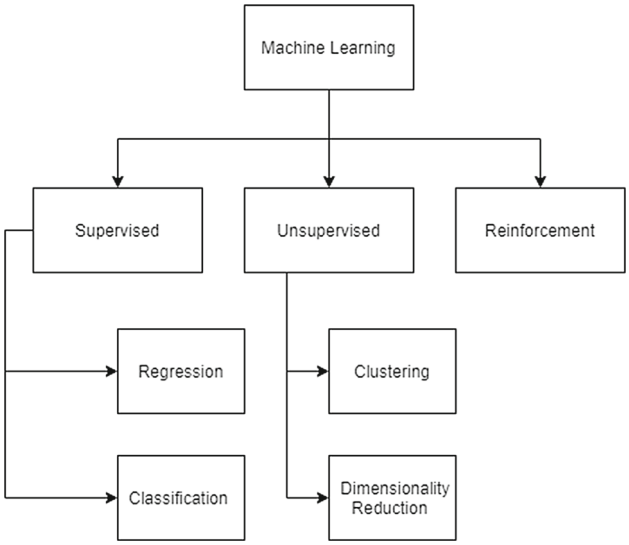

Fig. 4 Machine learning architecture

Semi-supervised learning is another type of technique in machine learning [6]. This technique is used when the expected result is known only for a few data points, i.e., when in a given dataset, not every data point is labeled. It helps in identifying and learning the structure of the input variables.

2.2.2 Unsupervised Learning

The second type of technique is unsupervised learning. In this type of technique, the data is not classified. The expected results are not known before its implementation. The system is presented with the dataset, and any of the unsupervised machine learning algorithms are applied to it, and the results are obtained. Since there is no supervision on the desired results, this technique is called unsupervised learning [7]. The main motive behind using this technique is to identify hidden clusters and patterns in the data. Examples of such algorithms are K-means clustering, self-organizing maps (SOM), hierarchical clustering, and others.

2.2.3 Reinforcement Learning

In reinforcement learning, the algorithm works on the concept of rewards and penalty. None of the data points in the dataset are labeled, i.e., the expected output is unknown, but the learning takes place in such a way that the system learns the environment on-the-go. The system is given an environment, and a set of actions are defined. Depending on the type of action the system takes, it is rewarded [8]. If the system's action takes it toward the goal state, it is rewarded. However, if the system's action takes it away from the goal state, it is charged with a penalty. To carry out this task, an objective function is determined, including all the possible action and state spaces. The main aim is to maximize this objective function, and policy is defined, which is to be followed. The most widely accepted and real-life example of reinforcement learning is how a toddler learns about the positive and negative outcomes of his actions by experiencing the rewards and penalties of those actions.

Machine learning algorithms can be employed in implementing recommender systems. Based on the type of recommender system being used, the data is collected. In collaborative filtering, the data is collected by performing social network analysis and identifying the items bought or liked by user's neighbors. These data points or choices serve as the dataset for implementing the algorithm. If content-based filtering is used, the user's past choices are identified as the data points. For example, for a movie recommender system, the movies user has watched in the past, the genre, director, actors, and others are taken into account, and these bits of information constitute the dataset. Further, algorithms are implemented on this dataset to recommend items. Lastly, hybrid filtering can be used for collating data using both supervised and unsupervised learning and presenting the final data points to the system.

2.3 Deep Learning

Deep learning is a discipline of machine learning algorithms that are layered. Here each layer does nonlinear processing producing different abstractions of the data. Each layer takes the output of previous layers as input, hence producing hierarchical abstractions. For example, in computer vision, the original image matrix is the data that is processed. The first layer takes this data, and it performs feature extraction and transformation in such a way that it identifies the edges. It then gives this extracted feature as an input to the next layer, which may create a new layer of abstraction by identifying

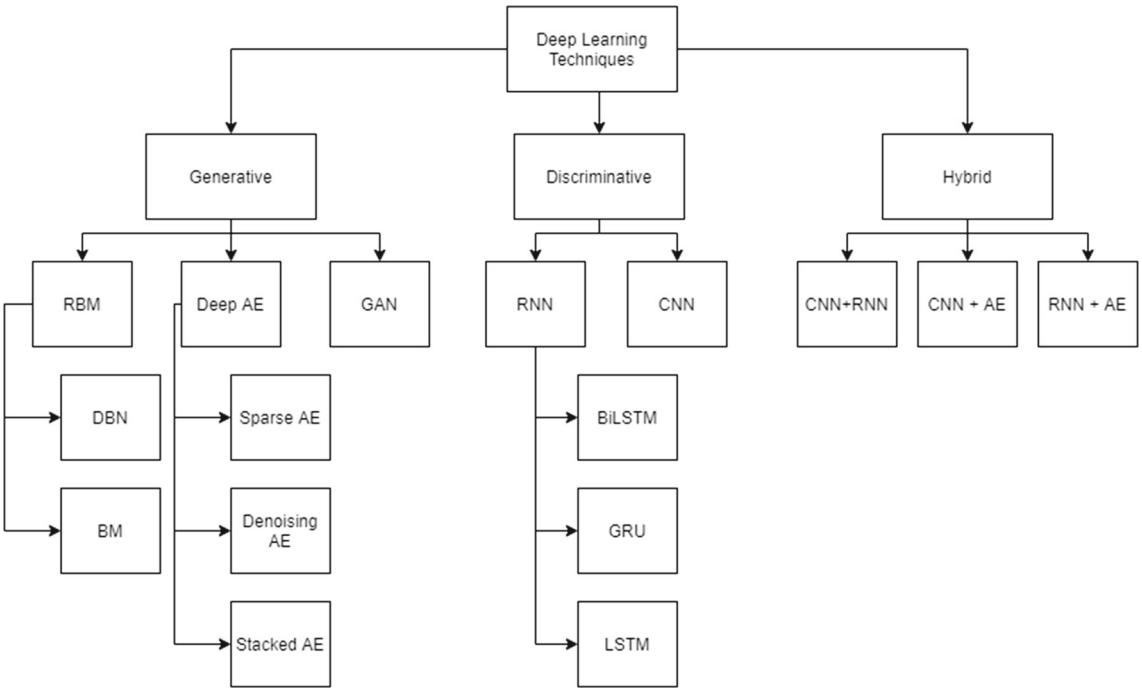

Fig. 5 Deep learning techniques

the orientation of the edges detected by the previous layer. This is repeated numerous times with multiple layers.

In this paper, as depicted in Fig. 5, research work done using several deep learning techniques has been studied.

2.3.1 Why Use Deep Learning?

The web is a vast pool of data; the dimensionality and modality of data present online are very high. To work with such multimodal and complex data with extensive features requires extensive machine learning support. If we use traditional design and technology, the recommendations generated will not be of any use because the system will not exploit the information at hand, and the complex pattern analysis of data would not be done. To solve this problem, we have used deep learning, which can work on highly complex data with hidden features and a high number of training instances. Deep learning's capability provides breakthrough results by extracting the data on a granular level and analyzing the patterns. Hence, the representation of all the details of data in a joint unified framework is made possible by using deep learning [9].

Conventional machine learning models like Matrix Factorization and others are linear models that can work efficiently on linear data, but when we have to deal with nonlinear data, the computations require complex functions like sigmoid, tanh, etc. To catch such intricate user–item interaction patterns, there arises a need for deep learning models.

The web is full of illustrative data; even for making recommendations, there is a lot of description and content attached to data. To understand this information and use it to our advantage, we have employed recommender systems. Another significant advantage of using deep learning is simplifying the process of feature engineering.

Every set of data has sophisticated features attached to it, and the task of deep learning is to understand the raw features and process them using supervised or unsupervised machine learning algorithms. Similarly, even for the content to every set of data, deep learning makes it possible for the system to use all the available data to generate expert recommendations.

Another important reason for using deep learning is sequential pattern mining. The task of sequence modeling that includes Natural Language Processing, speech recognition, etc. is usually performed by either recurrent neural network (RNN), which has internal memory to learn the next in sequence, or convolutional neural network (CNN) that performs the sequence modeling using temporal computations. We have explained this further in Sects. 4.3 and 4.6, respectively (Table 1).

3 Research Methodology

In this paper, various parameters were considered for identifying the relevant research papers to be included in the study. The broad area of consideration was deep learning in recommender systems. In this section, the research methodology adopted in this paper has been explored. Broadly, the research methods have been based upon ten attributes, i.e., publisher

Table 1 Application domains of studies

Application domain	Studies
Item recommendation	[10–19]
News recommendation	[20–25]
Movie recommendation	[26–31]
Cold-start problem	[32–36]
Session-based recommendation	[37–41]
Music recommendation	[42–45]
Text recommendation	[46–50]
Image recommendations	[51–53]
Social network-based recommendation	[54–56]
Hashtag recommendation	[57–59]
POI recommendation	[60–62]
Citation recommendation	[63–65]
Content-based recommendations	[66–68]
Video recommendation	[69, 70]
Quote recommendation	[71, 72]
E-learning recommendation	[73, 74]
Rating prediction	[47, 75]
Job recommendation	[33, 76]
Fashion recommendation	[77, 78]
Blog recommendation	[79]
Venue recommendation	[80]
Consumer preferences	[81]
Co-evolutionary latent feature processes	[82]
Artwork recommendation	[83]
Adaptive user-interfaces	[84]
Audience activity recommendation	[85]
Data sparsity	[86]
Healthcare service recommendation	[87]
Treatment recommendations	[88]

of the paper, year of publication, number of citations, and location of performing the study (Table 2).

3.1 Application Domain of Recommender Implementing Deep Learning

RQ 1 Which application domains have been included in the study?

There is a need for a dataset to apply machine learning or artificial intelligence algorithms to implement a recommender system. The dataset depends on the application domain of the recommender system. Table 1 presents all the application domains of the refereed papers and several studies performed in that domain.

It can be observed from the table that maximum work has been done in the domain of item recommendations, news recommendations, and movie recommendations.

Table 2 Dataset from external sources

Dataset	Studies
MovieLens	[13, 18, 23, 26–29, 31, 47, 48, 65, 74, 81, 85, 89–97]
Amazon	[10, 12, 15, 19, 36, 47, 48, 75, 86, 94, 95, 98–100]
Yelp	[12, 15, 16, 19, 60, 61, 75, 82, 86, 98]
IMDb	[29, 35, 41, 47, 96, 101]
NetFlix	[35, 69, 96, 101, 102]
CiteULike	[49, 89, 92, 102, 103]
Epinions	[52, 55, 87, 93]
Last.fm	[13, 41, 58, 96]
Delicious	[34, 58, 90, 96]
Million song dataset	[43–45, 100]
BookCrossing	[85, 97, 100]
FourSquare	[60, 61, 80]
The echo nest taste profile subset	[42, 44]
Dbbook	[46]
Wikiquote website	[71, 72]
Flixter	[54, 56]
Ciao	[55, 56]
YooChoose	[67, 100]
FilmTrust	[56, 85]
Twitter	[17]
Google news	[57]
Tumblr	[79]
Flickr	[80]
Picasa	[80]
Oxford concise dictionary of proverbs	[71]
Google Play	[104]
IPTV	[82]
YouTube	[70]
UGallery	[83]
Rossmann	[90]
Frappe	[91]
CLEF NewsREEL	[22]
Penn treebank	[100]
CADE web directory	[100]
RefSeer	[63]
Reddit	[41]
Economics thesaurus (STW)	[64]
EconBizRecSys evaluation dataset	[64]
Jester	[14]
PubMed	[65]
EqGraph	[84]
MSWeb	[84]
Assistments	[84]

Table 2 continued

Dataset	Studies
Beer	[15]
Douban	[93]
NAVER news	[24]
Instagram	[77]
Zalando	[77]
Trip.com	[17]
Facebook	[17]
Exact Street2Shop	[78]
Taobao advertising dataset	[18]
Meetup	[96]
DeepSurv	[88]

Table 3 Self-generated dataset

Self-generated dataset	Studies
1. Google chrome extension "Daum News Tracker" 2. Android application "KECI News.	[20]
1. Logs scraped from search engine Bing Web vertical 2 News article browsing history from Bing News vertical 3 App download logs from Windows AppStore 4 Movie/TV view logs from Xbox	[11]
User's news click history between 04/01/2014 and 09/30/2014	[21]
The images and users' information in this dataset were crawled from Flickr through its API	[51]
1. VIDXL—was collected over a 2-month period from a video site 2. CLASS—product view events of an online classified site.	[38]
"starc" platform which is based on Open edx for self-development which serves the fundamental education field	[73]
Collected from real-life users' browsing history of an online European department store over two weeks at the beginning of January 2016	[39]
Scraped Ukiyo-e images from the "the Ukiyo-e search service" and the Ukiyo-e pages of Ritsumeikan University Art Research Center	[52]
Wanted and missing persons' application of the Police of the Czech Republic	[53]
Collected 419,509 check-in records published by 49,823 users among 18,899 locations from August 2012 to July 2013 in Manhattan via the API of Foursquare	[62]
A large collection of quality article selection demonstration with an average length of 900 characters over six months, manually created by professional editors	[50]
1314 resumes which came in as a part of summer research intern application at IBM Research Labs	[76]
Sampled offline dataset collected from a commercial news recommendation application	[25]

Table 4 Type of recommender systems

Type of recommender system	Studies
Content based	[11, 13, 20, 21, 24, 25, 27, 30, 38, 45, 46, 50, 53, 65, 71–73, 78–80, 88, 99, 100]
Collaborative filtering	[10, 15, 17–19, 22, 23, 26, 31, 33–35, 39, 41, 48, 49, 52, 54–56, 58–61, 69, 70, 74, 75, 81, 84, 87, 89, 90, 92–94, 96, 98, 101, 103, 104]
Hybrid	[16, 28, 29, 42, 51, 68, 77, 97, 102]
Context-aware	[32, 47, 62, 63, 67, 85, 86, 95, 104]

3.2 Datasets Involved in the Referred Studies

RQ 2 What are the different pre-defined and self-generated datasets other authors have worked on in the past?

In this section, the datasets used in the research papers have been listed. In some papers, authors used publicly available datasets to perform the recommendation process using their proposed model. In other papers, authors scraped data from websites or applications using APIs or manually collected data to implement recommender systems. Table 2 enlists the external datasets used in previous studies, and Table 3 enlists self-generated datasets.

It is evident from Table 2 that the MovieLens dataset is extensively used for analyzing the effectiveness of implementing deep learning in recommender systems. Other widely used datasets are Amazon and Yelp.

3.3 Types of Recommender Systems Used in the Studied Papers

RQ 3 What types of recommender systems have been used in the past studies?

There are various types of recommender systems, depending on the recommendation technique used. In this section, the types of recommender systems used in the studied papers have been listed along with the studies they were used in.

It is evident from Table 4 that most of the studies have been done in content-based and collaborative filtering algorithms. This presents the readers an opportunity to work in other hybrids, context-aware, and other similar recommendation techniques.

3.4 Deep Learning Techniques Used in the papers

RQ 4 Which deep learning techniques have been used in the past research?

To efficiently implement the recommendation process, several deep learning techniques were used in the referred

Table 5 Classification on the basis of deep learning technique used

Deep learning technique	Studies
Recurrent neural network (RNN)	[10, 13, 21, 24, 37–41, 45–47, 49, 60, 62, 63, 65, 67, 71, 72, 74, 75], [41, 82, 84, 85, 99]
Deep neural network	[14, 16–18, 22, 27, 44, 53, 56, 59, 70, 83, 86, 90, 93, 95, 100, 104, 105]
Convolutional neural network (CNN)	[12, 15, 19, 24, 48, 50, 52, 60, 71, 76–80, 93, 94, 98, 102]
Deep belief network (DBN)	[20, 28, 36, 42, 61, 69, 73, 81]
Deep feed-forward neural network	[29, 31, 55, 57, 88, 104]
Stacked denoising autoencoder (SdA)	[23, 35, 94, 97, 101]
Collaborative deep learning (CDL)	[33, 51, 89, 102]
Sparse autoencoder	[58]
multi-view deep neural network (MV-DNN)	[11]
Supervised neural network	[26]
Deep learning matcher (DLM)	[32]
Variational autoencoder	[103]
Generative adversarial network (GAN)	[92]
Denoising autoencoder	[34]
Deep density networks (DDN)	[68]
Neural matrix factorization (NeuMF)	[96]

Table 6 Metrics used in the papers

	Metric
A	Time
B	Accuracy
C	Mean square error (MSE)
D	Root mean square error (RMSE)
E	Recall
F	Mean average precision (MAP)
G	Precision
H	Mean reciprocal rank (MRR)
I	Mean absolute error (MAE)
J	Rank score
K	Area under curve (AUC)
L	Normalized discounted cumulative gain (nDCG)
M	Pearson correlation
N	Hit ratio
O	F1-score
P	Mean average rank (MAR)
Q	Coverage

research work. Table 5 lists all the techniques used in the papers and hence, classifies the research work.

It can be seen from the table that the recurrent neural network (RNN) is the most used deep learning technique in recommender systems. Hence, users can try to implement other deep learning techniques and perform a comparative study.

3.5 Metrics Used for Analyzing the Results by Deploying Deep Learning in Recommender Systems

RQ 5 Which metrics have been used to analyze the results?

The result obtained by implementing the recommender systems is analyzed by using various metrics. Different papers deploy different metrics to get a multi-dimensional interpretation of the results. Table 6 enlists the metrics used in the papers. Figure 6 graphically analyzes the metrics used.

This analysis of metrics shows that the recall metric is used by maximum researchers to analyze their results. The graph also helps in analyzing the usage gaps between all the metrics (Fig. 7).

4 Related Work and Original Contribution of the Paper

Some of the significant works in deep learning in recommender Systems are covered in Table 7.

For performing the research papers' complete study, various aspects and attributes of these studies were divided into columns, and all such columns were collated together to form a spreadsheet. Such information clusters included the problem statement, the dataset, the proposed model, deep learning technique, type of recommender system used, and others. Later in the Conclusions section, the significant research gaps were identified and reported, extracted from the studied research papers. In this section, a holistic study of all the research papers has been presented.

In this subsection, the problems addressed in different papers have been categorized into eleven clusters of deep learning techniques as presented henceforth. Every subsection expounds a detailed research work done in recommender systems for each deep learning technique. At the end of every subsection, a detailed description of the findings and open issues have been presented.

4.1 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a feed-forward network with one or several computation layers and has nonlinear activations. It has at least one layer that is connected in a

Table 7 Related work

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
1	Toward Twitter Hashtag Recommendation using Distributed Word Representations and a Deep Feed Forward Neural Network	The use of hashtags is restricted due to the additional user effort required in assigning them	1. Use of word2vec model to learn the representation of word 2. Use of deep feed-forward neural network	Google News dataset	Deep Feed-Forward Neural Network	MSE	Content based	Hashtag recommendation
2	Improving Content-based and Hybrid Music Recommendation using Deep Learning	Need to improve the music recommendations because the traditional features are unable to encapsulate all the pertinent information in the audio	Merged the two tech-niques—Deep Belief Network and probabilistic graphical model into an algorithm that learned audio features while making personalized recommendations	The Echo Nest Taste Pro le Subset	Deep Belief Network	Root Mean Square Error (RMSE)	Hybrid and content based	Music Recommendation
3	Deep Learning of Semantic Word Representations to Implement a Content-based Recommender for the RecSys Challenge'14	While reporting a document in the final feature space, there is no consideration of the similarity of the word semantics	Found the current and latent semantic features relating to the data of documents and fitted a linear regression method to estimate user preferences for documents.	Dbbook	Recurrent Neural Network (RNN)	Root Mean Squared Error (RMSE)	Content based	Text Recommendation

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
4	Collaborative Deep Learning for Recommender Systems	In traditional CF-based recommender systems, the ratings are generally very sparse in several applications, this results in CF-based methods to devalue remarkably in their recommendation performance	Implemented collaborative deep learning that carried out deep representation learning for the content information and collaborative filtering for the feedback matrix	CiteULike, Netflix	Collaborative Deep Learning (CDL)	Recall, mean average precision (mAP)	Content based and CF	General
5	The video recommendation system based on DBN	To improvise video recommendations	Expanded the traditional depth of the Deep Belief Network model together with CF-based algorithm to construct a video recommender system	Netflix movie score datasets published in 2005	Restricted Boltzmann Machine (RBM) is the base of DBN (Deep Belief Network)	mean square error (MSE), MAE, RMSE	CF	video recommendation
6	Tag-aware recommender systems based on deep neural networks	Data sparsity, ambiguity, and redundancy are some issues which occur in the user-defined tags	To make recommendations, extracted features were used which were more abstract, dense and representative	Last.fm, Del.icio.us,	Deep Neural Network Sparse Autoencoder	Precision, recall, rank score	User-based CF	tag information

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
7	Multi-Rate Deep Learning for Temporal Recommendation	To remediate the problem of tackling large-scale real-time data to model temporal behavior	Implemented a deep neural network-based model to amalgamate the preferences of static long-term as well as temporal short-term to enhance the recommendations	user's news click history between 04/01/2014 and 09/30/2014	Recurrent Neural Networks (RNN)	Precision, Area Under Curve (AUC), Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR)	Content based	News recommendation system
8	Quote Recommendation in Dialogue using Deep Neural Network	To improvise the quote recommender system	Implemented a deep learning model to merge RNN and CNN to learn the semantic rendering for every dialog	Wikiquote website and Oxford Concise Dictionary of Proverbs	RNN and CNN	MRR, Recall, NDCG, Hit	Content based	quote recommendation
9	Solving Cold-Start Problem in Large-scale Recommendation Engines: A Deep Learning Approach	Remediating the cold-start problem in CF-based recommender systems	Implemented deep learning-based item–item matcher pertaining to the similarity in the documents using doc2vec model	Careerbuilder's CF-based recommendation engine	Deep Learning Matcher (DLM) was built utilizing doc2vec which is considered the current state-of-the-art deep learning algorithm for document embedding and matching	Pearson-r correlation, recall, precision, running time	Context based	Cold-Start Problem

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
10	Deep Learning with Consumer Preferences for Recommender System	To improve the problem of data sparsity	Realized a deep learning technique to anticipate the values of null ratings	MovieLens	Restricted Boltzmann Machines (RBMs) can be stacked and trained in a greedy manner to form so-called deep belief networks (DBNs)	Pearson Correlation Similarity(PCC), MAE	Collaborative filtering(CF)	Consumer Preferences
11	Comparative Deep Learning of Hybrid Representations for Image Recommendations	Certain tasks which are specific to the users, like image recommendations, require efficient representations of images and preferences of users over images.	Proposed a dual-deep net, wherein the two subnetworks map input images and liking of users into the same latent semantic space. Afterward, the distances between images and users in the latent space are deduced to make recommendations.	The images and users' information in this dataset are crawled from Flickr through its API	Collaborative Deep Learning (CDL)	Precision, Recall	Hybrid	Image Recommendations
12	"Tag-Aware Personalized Recommendation Using a Deep-Semantic Similarity Model with Negative Sampling"	To tackle the problem of redundancy, sparsity and ambiguity of social tags.	Implemented deep learning by including the tag-based user and item profiles to an abstract deep feature space, to maximize the resemblance between the deep-semantic users and their desired items		Deep Neural Network	recall, precision, F1-score, mean average precision (MAP) and mean reciprocal rank (MRR)	CF	Tag-Aware Personalized Recommendation

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
13	Wide & Deep Learning for Recommender Systems	To exploit and learn about the content and user interactions to make recommendations	Learned a high-dimensional semantic space independent of user interference, in which items were addressed based on their synonymity, and later learned a user-centric modification function to modify that space into a ranking pertaining to the previous preferences of the user	Google Play	Feed-Forward Neural Network	General	Context based	General
14	Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations	To work on the drawback of real-time recommender systems which make recommendation based on the clicks of the user during a session	Implemented parallel recurrent neural network to realize sessions specific to the clicks and the images and textual features of the clicked items	1. VIDXL—was collected over a 2-month period from a YouTube—like video site 2. CLASS—product view events of an online classified site	Parallel RNN (p-RNN)	Recall, MRR	Content based	Session-based Recommendations
15	"Collaborative Filtering and Deep Learning Based Hybrid Recommendation For Cold Start Problem"	To remediate the data sparsity and cold-start problems	Implemented an recommender system that was a hybrid of item content features extracted from a deep neural network and its application to the timeSVD ++ CF model	Netflix, IMDB	SDAE	RMSE	CF	Cold-start problem

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
16	Application of deep belief nets for collaborative filtering	Working toward the problem of poor recommendation accuracy caused due to data sparsity	Extracted user features using Deep Belief Network and then applied K-nearest neighbor algorithm to make recommendations	MovieLens	Restricted Boltzmann Machines	RMSE	Hybrid	Movie recommendation
17	"DBNCF: Personalized Courses Recommendation System Based on DBN in MOOC Environment"	Low effectiveness of resource recommendation in MOOC environment due to data sparsity of online learners	Implemented Deep Belief Network by combining the online learner course feature vector for mining the learner's interest and using the course score as the class labels	"starc" platform which is based on Open Edx for self-development which serves fundamental education field	DBN	RMSE	Content based	MOOC Courses Recommendation
18	On Deep Learning for Trust-Aware Recommendations in Social Networks	The recommendations depend largely upon initializing the user and item latent feature vectors	Used deep learning to initialize in matrix factorization for trust-aware social recommendations and to study the result of the community in user's trusted friendships	Epinions and Flixster	Deep Autoencoder, which is an efficient approach to nonlinear dimensionality reduction	root mean squared error (RMSE), coverage, F-measure, precision	CF	social network-based recommendation
19	dTrust: a simple deep learning approach for social recommendation	To improve the privacy in social recommender systems	Built a system that depends on the topology of an anonymous trust-user item network which amalgamates user trust relations and rating scores	Epinions and Ciao	Deep Feed-Forward Neural Network	Root Mean Square Error (RMSE) and Mean Absolute Error (MAE)	CF	social recommendation

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
20	Application of Deep Learning to Sentiment Analysis for Recommender System on Cloud	The skewed contextual information result in inefficient sentiment analysis of short texts	Used deep learning to optimize the recommendations based on the sentiment analysis for various reviews retrieved from several online social networking websites.	Amazon	RNN	accuracy, F1-Measure, Response Time	Content based	Text Recommendation
21	A Recommendation Model Based on Deep Neural Network	To solve the problem of data sparsity	Implemented a system which took into account the tag and user information and retrieved the K-nearest neighbors	MovieLens, Epinions	Deep Neural Network (DNN).	Mean Absolute Error (MAE) method and the Root Mean Squared Error (RMSE)	CF	General
22	Neural Citation Network for Context-Aware Citation Recommendation	To solve the problem of appropriate citations in a fast-growing academic world	Proposed a context-aware citation recommender system	RefSeer dataset	Recurrent Neural Network	Recall, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Normalized discounted cumulative gain (NDCG)	Context based	Citation Recommendation
23	A Deep Architecture for Content-based Recommendations Exploiting Recurrent Neural Networks	To curb the problem of varied lengths of data to report the content of items to be recommended	Implemented a deep learning based Long Short-Term Memory (LSTM) network to collectively learn embeddings depicting the items to be recommended and user's preferences	MovieLens, Dbbook	RNN	F1-Measure	Content based	Content-based Recommendations

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
24	Recommendation with Social Relationships via Deep Learning	To improve social recommendations	Proposed a deep learning approach to improve social recommendations	Epinions, Flixster, Ciao and FilmTrust	Deep Neural Network	Recall, Ndcg	CF	Recommendation with Social Relationships
25	Recurrent Latent Variable Networks for Session-Based Recommendation	To curb the limitation of the sequential nature of the addressed predictive setup, and data sparsity	Proposed an algorithm wherein, the network recurrent units were considered as stochastic latent variables with a previous distribution assigned to them		Recurrent Neural Network	Recall, Mean Reciprocal Rank (MRR)	Content based	Session-Based Recommendation
26	A Deep Multimodal Approach for Cold-start Music Recommendation	To solve the problem of cold-start for new artists in music recommender systems	Applied deep learning on text and audio information alongside user feedback data	Million Song Dataset (MSD), Echo Nest Taste Profile Subset	Deep Neural Network	General	collaborative filtering(CF)	Music Recommendation
27	Re-ranking-based Recommender System with Deep Learning	To solve the problem of data overload of scientific papers for researchers and scholars	Proposed a deep learning technique which determined top-k recommendations and re-ranked the results	Economics thesaurus (STW), EconBizRecSys evaluation dataset	Deep Neural Network (DNN)	normalized discounted cumulative gain (nDCG)	Content based	scientific papers

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
28	Attention-based Recurrent Neural Network for Location Recommendation	To improve the shortcomings of the traditional POI recommender system	Proposed a deep neural network to analyze the sequential check-ins of users and capture their life patterns	collected 419,509 check-in records published by 49,823 users among 18,899 locations from August 2012 to July 2013 in Manhattan via the API of Foursquare	Recurrent Neural Network (RNN)	Precision, Recall	Context based	Point-of-Interest (POI) recommendation
29	Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks	To work toward the inclusion of the side information in Recurrent Neural Networks	Implemented a Context-aware Recurrent Neural Network (CRNNs) that combined the contextual and the item embeddings	YooChoose dataset	Recurrent Neural Networks (RNNs)	Recall	context-aware recommendation	Recommendations involving contextual/side information
30	Product Recommendation: A Deep Learning Factorization Method Using Separate Learners	To improve the products recommendations	Implemented a deep neural network factorization method for separate learners	Jester	Deep Neural Network	Root mean squared error (RMSE)	Content based	product recommendation
31	Online News Recommender Based on Stacked Autoencoder	To solve the problems of data overloading, high-dimensionality, and data sparsity	Implemented a denoising stacked autoencoder to extricate the important low-dimension features from the sparse user–item matrices	MovieLens	Stacked Denoising Autoencoder (SDAE)	Mean Absolute Error (MAE), Precision, Recall	CF	News Recommendation

Table 7 continued

S. No.	Title	Problem	Solution	Dataset	Deep learning technique	Metric used	RS technique	Application domain
32	Deep Sequential Recommendation for Personalized Adaptive User Interfaces	To remediate the problem of data sparsity in Adaptive user-interfaces (AUIs).	Proposed an RNN based model that deployed Gated Recurrent Units to map user interaction histories to vectors in a Euclidean space exchanged among the user and desired item vectors	EqGraph, MSWeb and Assistments'15	RNN	mean average precision (MAP) and normalized discounted cumulative gain (NDCG)	CF	Adaptive user-interfaces (AUIs) based recommender
33	Spatial-Aware Hierarchical Collaborative Deep Learning for POI Recommendation	To improve data sparsity and cold-start problem	Implemented a Spatial-Aware Hierarchical Collaborative Deep Learning model (SH-CDL). The model carried out deep representation learning for Point of Interests from heterogeneous features and hierarchically additive representation learning for spatial-aware personal choices	Yelp, Foursquare	Deep Belief Network	Mean Absolute Error (MAE), Accuracy	CF	Point-of-interest (POI) recommendation

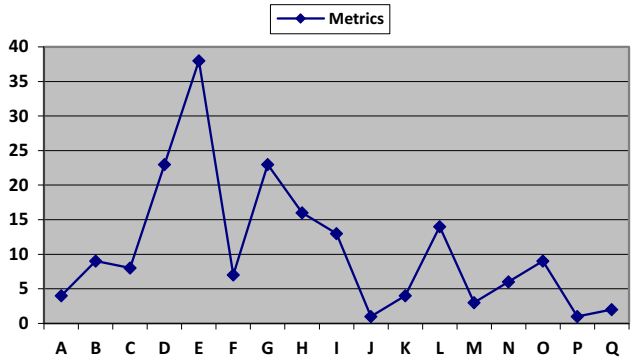

Fig. 6 Graphical analysis of metrics

feed-forward manner [106]. It can also be used to transform the linear methods of recommender systems into nonlinear models. Convolutional neural networks (CNN) are also a kind of feed-forward network.

In a deep forward neural network, the information flows in one direction across multiple layers, as shown in Fig. 7. The output from one layer becomes the input into the next layer. This architecture does not consist of any cycles, and hence it is called "feed-forward."

For a feed-forward neural network with D inputs, $x = [x_1,\ldots..,x_D]$, a layer with K hidden nodes $h = [h_1,\ldots..,h_K]$, then the output node y is given by Eq. 4:

$$y = v^T h = v^T f\left(w^T x\right) \tag{4}$$

where $v = [v_1,\ldots..,v_K] \in \mathbb{R}^K$, $W = [w_1, \ldots.., w_K] \in \mathbb{R}^{D\times K}$, f represents the nonlinear activation function.

The given Eq. 5 mathematically represents the value of every hidden node.

$$h_k = f\left(w_k^T x\right) = f\left(\sum_{d=1}^{D} w_{dk} x_d\right). \tag{5}$$

4.1.1 Wide & Deep Learning

The wide and deep learning model is capable of solving both regression and classification problems [107]. The wide learning element is a generalized linear model consisting of a single layer perceptron, whereas the deep learning element consists of the *multilayer perceptron*. The blend of the two above stated elements leads to the inclusion of both *memorization* and *generalization*. The ability of wide learning element to capture prominent features from historical data results in *memorization*. Whereas the ability of deep learning element to create general and abstract representations results in *generalization*. The amalgamation of the two results in diverse results and better performance of the recommender system.

The architecture of this model is given in Fig. 8.

This concept can be represented mathematically using Eq. 6, which represents the wide learning element, and Eq. 7, which represents the deep learning element, respectively. The final wide & deep learning model is represented by Eq. 8:

$$y = W_{\text{wide}}^T \{x, \phi(x)\} + b. \tag{6}$$

Here, W_{wide}^T and b represent the model parameters, x represents the raw input feature, and $\phi(x)$ represents the transformed feature.

$$\alpha^{l+1} = f\left(W_{\text{deep}}^l a^l + b^l\right). \tag{7}$$

Here, l represents the lth layer, $f(\cdot)$ represents the activation function, W_{deep}^l represents the weight term, and b^l represents the bias term

$$P\left(\hat{r}_{ui} = 1|x\right) = \sigma\left(W_{\text{wide}}^T \{x, \phi(x)\} + W_{\text{deep}}^T a^{l_f} + \text{bias}\right). \tag{8}$$

Here, $\sigma(\cdot)$ represents the sigmoid function, $\hat{r}_{ui}$ represents the binary rating label, a^{l_f} represents the final activation.

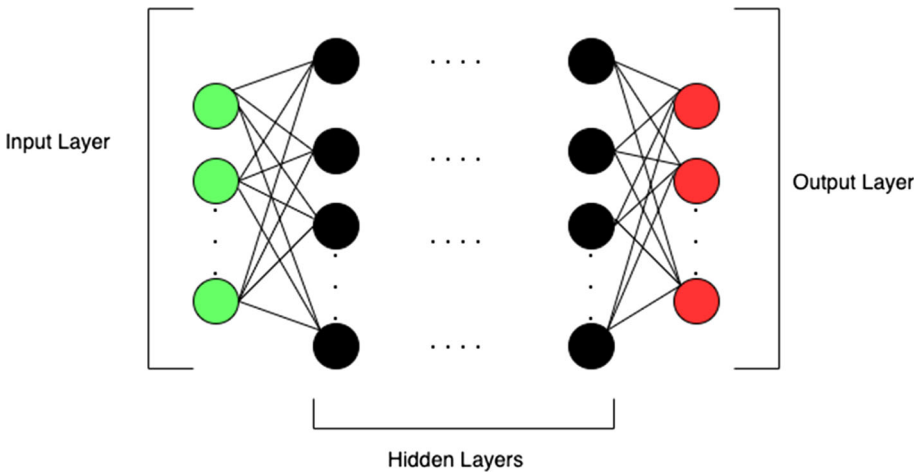

Fig. 7 Architecture of deep feed-forward neural network

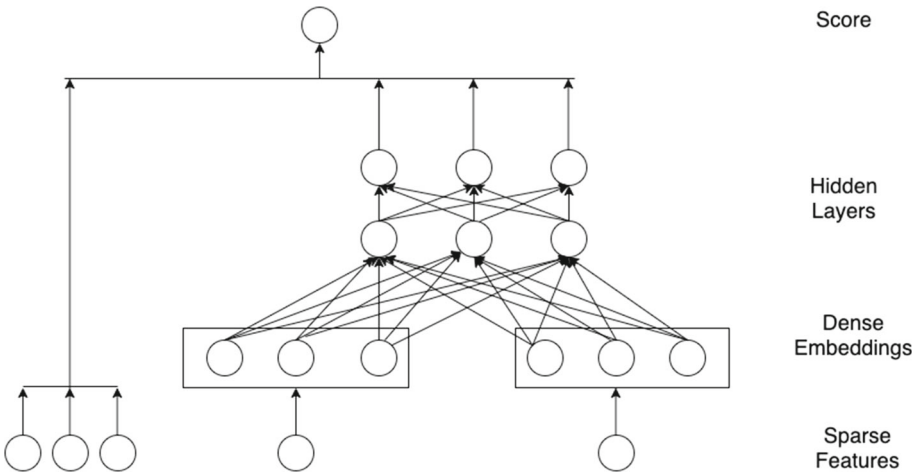

Fig. 8 Architecture of wide & deep network

Authors in [108] used a feed-forward multilayer neural network for collaborative filtering recommender systems. Once the model learned the hidden factors and features, the system feed-forwarded the latent features to identify <user,item> pairs having NULL ratings. The authors [57] realized that the usage of hashtags in social media required additional efforts by the user. Hence, they used the deep forward neural network to recommend relevant hashtags to the users. To carry out this task, first, they performed tweet collection and data preprocessing, followed by feature engineering by vector generation. In the end, they performed training and evaluation. In another work, the authors addressed the sparsity of data by using Wide & Deep learning by training comprehensive linear models with feed-forward deep neural networks [104].

It was found out that rating prediction techniques often rely on user's private information that is a threat to privacy. Hence, the authors came up with a feed-forward neural network-based dTrust model that used the topology of anonymous user–item interactions that combined user's trust relations with user rating scores [55]. In another implementation of the feed-forward deep neural network, the researchers addressed the 0/1 recommendation problem by combining the ratings given by users and Natural Language Processing (NLP) of texts [29]. For using hashtags, user-defined tags usually suffer from various problems like data sparsity, redundancy, ambiguity, and others. To solve this problem, the authors [58] abstracted features that were used to make recommendations instead of raw data. The authors also addressed this problem, and they solved it by using deep neural network [59].

In another work, the authors observed that since recommender systems influence both content and user interactions to create recommendations that adapt to the users' preferences, this can be used as a leverage to improve recommendations [27]. They proposed a Deep Space model that learned a user-independent high dimensional based on their substitutability, semantic space items were positioned, and according to the user's past preferences, it learned user-specific transformation function to convert this space into a ranking. In news recommender systems, modeling temporal behavior, the cost of estimating the parameters also increases, making the recommendations costly. The authors in [22] used the deep neural network to address this issue. It was observed by some researchers [56] that the preference of choosing friends in social media did not always match. Hence, it became challenging to recommend friends on online social networks. Thus, they introduced a deep learning approach to learn about both user preferences and the social influence of friends for an effective recommendation. In the paper [44], as a solution to the cold-start problem in music recommenders, the authors combined text and audio information with user feedback data using deep neural network frameworks.

Some authors in [53] observed that sometimes erroneous police photo lineups resulted in the conviction of innocent suspects. To avoid this, they came up with a two leveled approach. The first one was based on the visual descriptors of the deep neural network, and the other was based on the content-based features of persons. The authors [14] proposed a model based on the deep neural network for product recommendations, which required only ratings for making recommendations. In another work [18], the authors realized that the amount of calculation for the learned model to predict all user–item pairs' preferences was humungous. To solve this problem, they proposed a TDM attention-DNN (tree-based deep model using attention network). For an appropriate research paper recommendation to the scholars, a study [64] used the deep neural network with the paragraph vectors re-ranking method for adequate recommendations. It was observed by some researchers that side information written for business reviews were seldom taken into account for the recommendation [16]. Hence, they used an artificial neural network in a hybrid recommender with the inclusion

of side information. Some authors [17] addressed the issue of cross-domain in social recommendation. To solve this problem, they introduced the model Neural Social Collaborative Ranking (NSCR), which immaculately integrated user–item interactions of the information sector and user–user social relations of the social sector.

In a study, it was observed that the existing recommender systems did not take into account the effect of distrust among users [87]. They generate recommendations pertaining to the trust relations among users. As a solution, items were recommended for both the trust and distrust relations among active users. In this paper [109], the authors proposed a novel recommender system *RecDNNing* that combined the user and item embeddings using a deep neural network. First, the authors created deep embedding for users and items, and later the average and concatenated values of those embeddings are given as input to the system. The deep layers generated recommendations using the forward propagation method. The authors in [110] used the hybrid of content-based and collaborative filtering approaches to create a deep classification model for generating efficient music recommendations. To carry out this implementation, they came up with the Tunes Recommendation Systems (T-RecSys) algorithm. Authors in [111] proposed a deep learning-based novel collaborative filtering algorithm. The input of the system were the normalized values of user and item rating vectors. This resulted in decreasing the time complexity as the system need not learn the features of users and items. In paper [112], the researchers applied deep learning in the domain of agriculture. They used the Twitter platform to scrape agriculture tweets and applied sentiment analysis on those tweets. This helped the authors to predict the sentiment range of agriculture tweets.

4.2 Findings and Open Issues

Deep feed-forward networks Multilayer Perceptron is used to approximate any measurable function to any given degree of accuracy. It also acts as a basis of typical advanced approaches and is widely adopted in numerous domains. Implementing Multilayer Perceptron for feature representation is very elementary and remarkably efficient, although it might not be as demonstrative as autoencoders, CNNs, and RNNs.

However, one major limitation of deep feed-forward neural networks is that they do not have memories or loops for remembering preceding computations [113].

One of the primary reasons for employing Deep Neural Networks is that they are synthesized so that multiple neural networks can be merged into one big differentiable function and trained end-to-end. This application becomes a necessity because of the abundant availability of multimodal data. The DNN framework for recommender systems typically extracts user and item feature vectors or latent and explicit features. Another significant advantage of Deep Neural Network is that it can effectively extract essential features from raw data automatically.

The effect of upper-case letters in hashtags used for hashtag recommendations can be explored, and its effect on the recommendations can be analyzed in detail. Another scope in the future is to analyze the effect of language modeling on prediction performance. Another scope is to remediate the cold-start problem for user-based Collaborative Filtering by learning the similarities between user-to-user and user-to-job. This can be implemented on a multimodal document embedding.

4.3 Multi-view Deep Neural Network (MV-DNN)

Multi-view deep neural network (MV-DNN) is excellent in modeling domain recommendations [107]. It considers users as the main view and every other domain, say Z, as a secondary view. For every secondary or auxiliary user-domain pair, there exists a specific similarity score. The loss function of MV-DNN can be computed using Eq. 9. The architecture of MV-DNN is represented in Fig. 9.

$$\mathcal{L} = \text{argmin}_{\theta} \sum_{j=1}^{Z} \frac{\exp\left(\gamma \cdot \cos\left(Y_u, Y_{a,j}\right)\right)}{\sum_{X' \varepsilon R^{da}} \exp(\gamma \cdot cos(Y_u, f_a(X')))}. \quad (9)$$

Here, θ represents the model parameters, γ represents the smoothing factor, Y_u represents the user's output view, a represents active view's index, and R_{da} represents view a's input domain.

Figure 10 showcases the structure of the Deep Structured Semantic Model (DSSM) [11]. The crude textual features are fed as input in the form of a high-dimensional vector. The DSSM forwards these inputs to two neural networks separately and maps them into semantic vectors into a joint semantic space.

In Eqs. 10, 11, and 12, x represents the input vector, y represents the output vector, W_i represents the ith weight matrix, and l_i represents the hidden layers, such that $\forall i \in \{1, \ldots.., N-1\}$. q represents the query and d represents the document.

$$l_1 = W_1 x \quad (10)$$

$$l_i = f(W_i l_{i-1} + b_i), i \in \{2, \ldots\ldots, N-1\} \quad (11)$$

$$y = f(W_N l_{N-1} + b_N). \quad (12)$$

The activation function is given by Eq. 13:

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \quad (13)$$

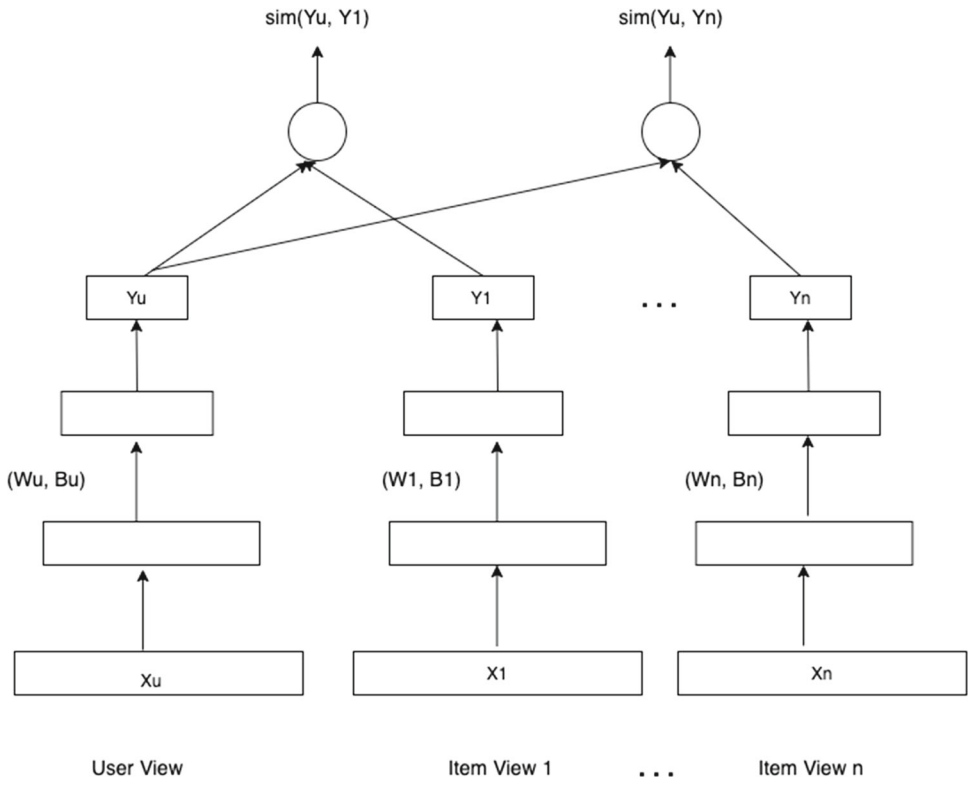

Fig. 9 Architecture of MV-DNN

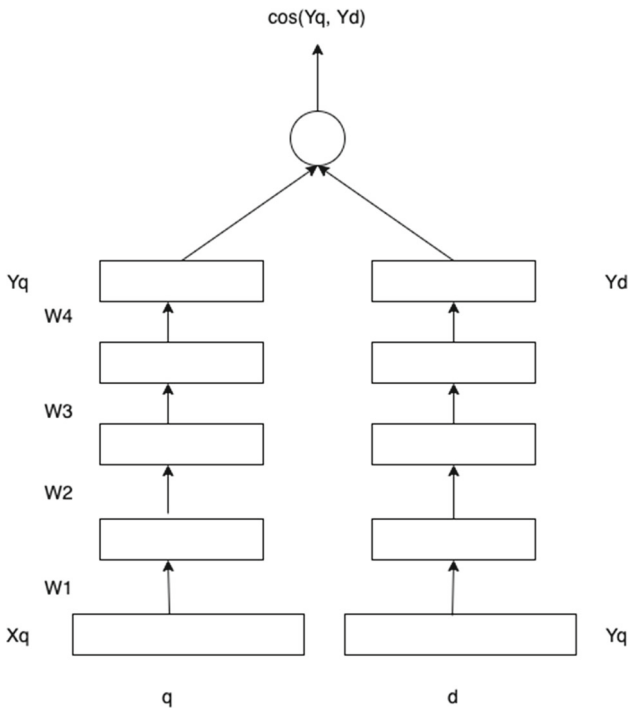

Fig. 10 Architecture of DSSM

The semantic relevance score between a query Q and a document D is given by Eq. 14:

$$R(q,d) = \cos\left(Y_q, Y_d\right) = \frac{Y_q^T Y_d}{\|Y_q\| \cdot \|Y_d\|}. \tag{14}$$

MV-DNN can be implemented in several domains. Conforming to the concepts of user-based collaborative filtering, as discussed in Sect. 2.1.2, users having similar preferences in one domain tend to have similar preferences in other domains as well. However, in many cases, this assumption may be rendered ineffective. Hence, the basic knowledge of the correlations between different domains is an essential aspect of MV-DNN. It is based on a Deep Structure-based Semantic Model (DSSM). In MV-DNN, the architecture of DSSM contains multiple views to map sparse features in high dimensions into the low-dimensional dense matrix. The authors observed that the online services present humungous content to the users, making this content user-centric [11]. Hence, they came up with the Deep Structured Semantic Model (DSSM), which used content-based filtering to improve the recommendations.

The authors on [30] observed that current CF-based techniques could only comprehend a single type of relations. RBM, for example, considers either user–user or item–item relations. The matrix factorization approach considers use-item relations only. To fix this problem, the authors propose a framework that first learns low-dimension user and item

vectors. This captures the user–user and item–item relations. This is then passed to a multi-view deep neural net, which models the user–item interaction.

4.4 Findings and Open Issues

Initially, Multi-view deep neural network (MV-DNN) was incepted for performing cross-domain recommendations.

However, the understanding of cross-domain recommendations may not be fruitful for MV-DNN. This is said essentially because the underlying literature of generating recommendations states that if a user likes an item a and there exists an item b similar to item a, then the user will like item b as well. However, this is not always true. For the times this hypothesis fails, this assumption becomes obstructive for the implementation of MV-DNN.

4.5 Convolution Neural Network (CNN)

For bio-inspired Multilayer Perceptrons and Computer Vision, CNN are the most used deep learning models. The essential elements of CNN are the convolutional layers and the subsampling layers. The convolutional layers work as a filter for the output from previous layers. These convolutional layers hence produce filtered outputs. The subsampling layers subsample the convolution output based on their activations [114]. To create a deep CNN model, the convolutional layers and the subsample layers are added alternatively. Hence, such a deep model of CNN can learn a hierarchy of complex features. Another massive advantage of CNN is that it has fewer parameters than the traditional feed-forward neural networks. This method is based on the feed-forward deep neural network. To reduce the preprocessing, multilayered perceptrons are used in this model. The architecture of CNN is shown in Fig. 11.

Cooperative neural network (CoNN) is a model where two neural networks work in tandem [15]. To discover novel user and item features, user and item reviews are given as input to the system. A common shared layer is added on the top of the two neural networks which couples them and the user and item features are mapped together into a common feature vector space to enable the interaction among them. Figure 12 represents the architecture of CoNN. Another latent layer is added to the architecture to enable the interaction of user and item latent features.

The two neural networks, i.e., neural network for items (NN_i) and neural network for users (NNu), run parallel. The item and user ratings are fed as inputs to the two neural networks, respectively. In the lookup layer, the user and item review texts are placed as matrices of word embeddings. The subsequent layers perform the functions of convolution, max pooling, and full connection, respectively. This layer also acts as a platform for computing the objective function for calculating the rating prediction error. One major drawback of Cooperative Neural Network is that it is incapable of addressing users and items which do not have ratings.

Many researchers addressed data sparsity, and they used CNN in their papers to solve this problem.

The authors in [15] observed that maximum recommender systems ignored the reviews leading to increased data sparsity problem. Thus, they proposed Deep Cooperative Neural Networks (DeepCoNN) model to learn item properties and user behaviors in the review text. Similarly, to solve the issue of the sparsity of data, authors integrated convolutional neural network (CNN) into probabilistic matrix factorization (PMF), resulting in convolutional matrix factorization (ConvMF) [48]. In the research work [92], the researchers solve the data sparsity problem by including tag and user information. First in nearest neighbors set, the most significant user impact is found using similarity metric, to process item information CNN is used. To get the results, the prediction matrix is decomposed by the probability matrix.

The profound use of CNN in the fashion industry was acknowledged in some research papers. The authors in [77] observed that not much work had been done on complicated recommendation scenarios involving knowledge transfer across multiple domains in fashion recommendations. Similarly, for online clothing shopping, current methods did not address the challenges in the cross-domain clothing retrieval scenario completely. The intra-domain and cross-domain data relations were considered together, and the numbers of matched and unmatched cross-domain pairs were imbalanced. Hence, the authors [78] proposed a deep cross-triplet embedding algorithm and a cross-triplet sampling technique to provide improved recommendations.

Some other applications of CNN have been included in the survey. For efficient venue recommendation, the authors proposed a City Melange framework that matched the interacting user to social media platforms with similar interests [80]. As a result, this approach could recommend both on- and off-the-beaten-track locations to the users. In another work, the authors proposed a hybrid music recommender system that used CNN to model real-world users' information and high-level rendering of audio data [102]. The authors observed in [52] that finding appropriate Ukiyo (a Japanese firm) e-prints that intrigue a novice is challenging. Thus, they proposed Ukiyo-e recommendation using a deep learning-based CNN model to present recommendations. The input to the recommender system is an image provided by the user. The CNN model is used to create a classifier that takes input images and other information related to the image and outputs whether the user will like or dislike it. The authors in [93] used tag information and user information and improved the recommendation by obtaining the nearest neighbor set that significantly impacted on the target user. They named their model convolution deep learning model on Label Weight

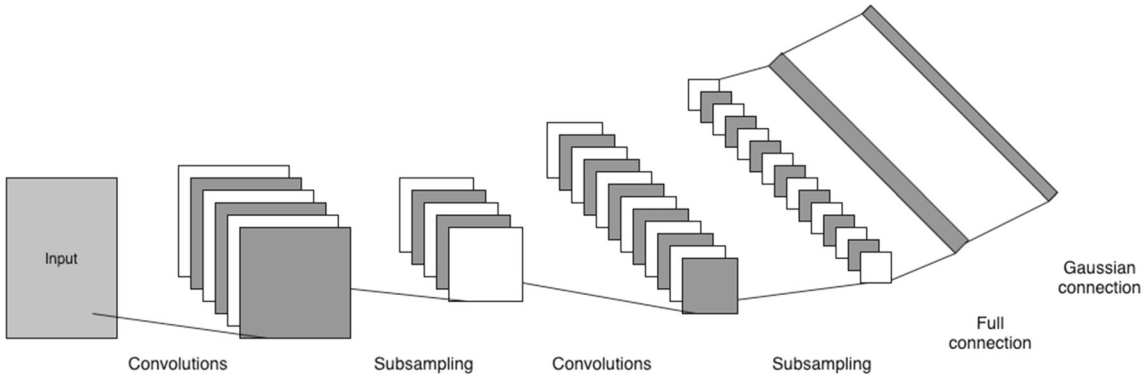

Fig. 11 Architecture of CNN

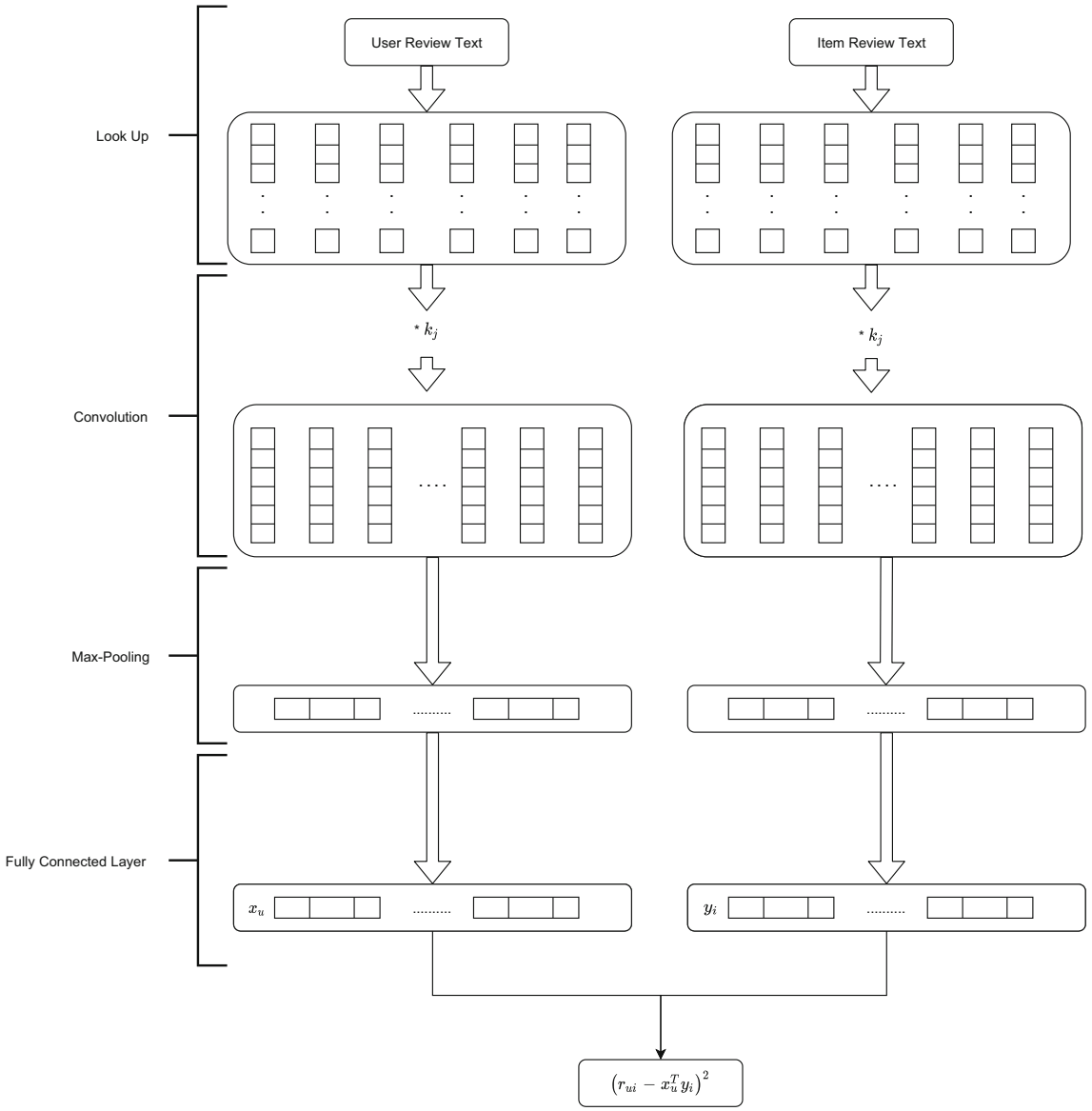

Fig. 12 Architecture of CoNN

Nearest neighbors (LWNCDL). Traditionally, candidate articles were handpicked from a vast pool of articles, and hence, its recommendation was manual and laborious. To solve this problem, the Dynamic Attention Deep Model (DADM) was proposed, which used multiple neural networks [50]. As a result, the authors used the convolutional neural network to study the connection between visual and textual inputs and examine the possibility of knowledge transfer from complex domains for expert recommendations.

The researchers in [76] addressed the problem of job recommendation by analyzing semi-structured resumes of candidates. To recommend a job, they used CNN to process the entire input at once, which is important since any part of the text in a resume can affect its semantics. The authors in [115] proposed an AODR model that considers user and item rating texts to infer item aspects and user opinions by applying deep learning. Such item aspects and user opinions were later merged using collaborative filtering to learn insightful information. In paper [106], the researchers introduced an algorithm, Aspect-Based Opinion Mining (ABOM), which analyzed the user reviews extensively to learn hidden features that further improve the recommendation accuracy. To carry out this task, they used multichannel deep CNN (MCNN) and Tensor Factorization (TF) machines. In paper [116], the authors used decentralized knowledge graphs in deep recommender systems to validate the efficiency of the system. The knowledge graph was constructed by crowdsourcing. The authors in [117] created a passenger hunting recommender system. The proposed system consists of two components, i.e., offline training and online implementation. In the former component, the authors applied deep CNN, and in the latter component, the authors proposed the DL-PHRec method. Hence, the authors could generate a personalized ranking list of destinations regions for each taxi driver.

4.6 Findings and Open Issues

CNNs are efficient in handling unstructured multimedia data with convolution and pooling functions. The majority of CNN based recommender models use CNNs for performing feature extraction. CNN based models are instrumental in learning deep features to model user and item latent factors. The significant advantage of using CNN is its implementation of pooling operation for reducing training data dimensions. CNN has been used for feature extraction to efficiently model user and item representations for Recommender Systems. It is also useful in image processing.

Another new scope is to embed the user's rating information in a matrix. A Convolution Neural Network can be trained on this matrix, treating it like an image where each data point represents a particular feature of the user. Exploiting Convolution Neural Networks for prediction of risk with medical imaging can be explored.

One major limitation is that it requires massive hyperparameter tuning to extract optimal features. In addition to this, it is also challenging to support intricate activity details. Although CNN uses feed-forwarding, it has fewer parameters than traditional deep feed-forward networks [114].

4.7 Autoencoders (AE)

Autoencoders are typically unsupervised neural networks that are trained to mirror its input as output. It is a tri-layered neural network, consisting of the input layer, hidden layer, and the output layer. The input layer is fed with the complex representations of the dataset, and in the hidden layer, such complex representations are transformed into low-dimensional representations. It essentially mirrors the working of an encoder, which encodes the complex, high-dimensional representations into low-dimensional representations. Mirroring the operations in a reverse manner, the low-dimensional representations are converted into high-dimensional representations as the data travels from hidden layer to output layer. This can also be termed as the working of a decoder. The architecture of denoising autoencoder is given in Fig. 13.

The feature extraction performed as the encoding is not robust enough. It was observed that the addition of Gaussian noise improved the problem described above [118]. Hence, to make the system more robust and training the hidden layers to identify hidden data features, denoising autoencoders were introduced. The structure of such autoencoders encodes the input while preserving the information about the input, and reduces the effect of the alteration process, applied stochastically to the input of the autoencoder. The authors in this work attempted to improve the automatic recommendations [26]. This was done by extracting the features of input data and reconstituting the input to perform recommendations. Authors in [119] extended generative models for the task of Collaborative filtering. To do this, they used Wasserstein autoencoders.

4.7.1 Stacked Denoising Autoencoders (SDAE)

A stacked autoencoder is a neural network having several layers of sparse autoencoders wherein the output of each layer is forwarded as the input of the successive layer. The hidden layers in a denoising autoencoders can be colluded to create a deep network by forwarding the output of the previous layer as the input to the next layer. Such a strong feature extraction capability makes stacked denoising autoencoder aptly suited for recommender systems. The architecture of a stacked denoising autoencoder is given in Fig. 14.

The method of generating top-N recommendations using *Stacked denoising Autoencoder* starts with selecting a dataset with user reviews [23]. Then, the similarity between the items

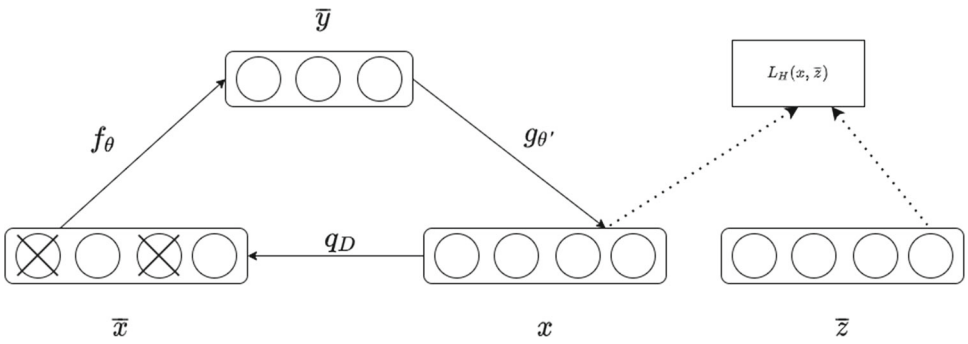

Fig. 13 Architecture of denoising autoencoder

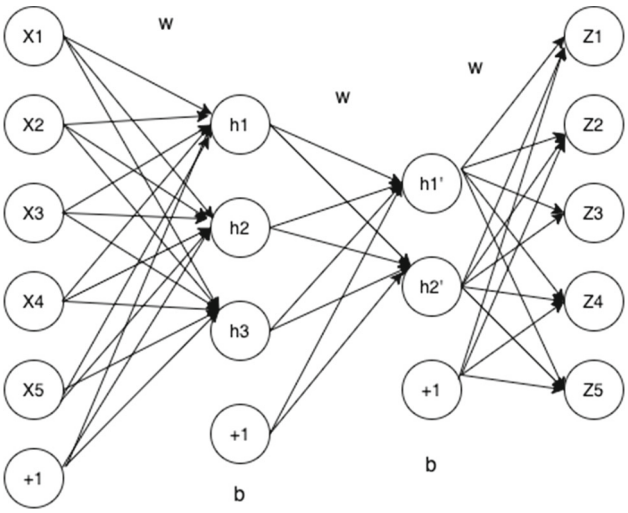

Fig. 14 Architecture of stacked denoising autoencoder

liked by the user is calculated for each user. Afterward, the nearest data points with a defined length, say K, are selected. Such K-nearest datasets are combined, and the user reviewed movies are excluded from such datasets to form another dataset. The similarity between the two datasets is calculated, and the top-N similar movies among the two datasets are selected and recommended to the user.

Upon stacking the denoising autoencoder, the deep model hence created increases the capability of feature extraction. To make this task robust, Gaussian noise is added to the system.

The authors proposed a Hybrid Recommendation system with CF and Deep learning (HRCD) [101]. It explored the content features of the items learned from a deep learning neural network and applied later to the timeSVD ++ CF model. The authors in [23] addressed the problem of information overloading and data sparsity. As a solution, stacked autoencoders were used with denoising to extract low-dimensional features from the sparse user–item matrix. It was observed that personalized recommendations often led to sparse observations of users' adoption of items. As a solution to the problem, the authors used Collaborative Topic Regression with Denoising Autoencoder or (CTR-DAE) [34]. Here, the user's community-based preference was bridged with his topic-based preference.

Similarly, some other researchers addressed data sparsity by proposing a hybrid model that executed deep learning of users and items' latent factors from side information and CF obtained from the rating matrix [97]. According to some authors, CF recommender systems suffer from Complete Cold-Start (CCS) problem where zero rating records are present and Incomplete Cold-Start (ICS) problems where significantly less rating records are available for new items or users in the system [35]. Hence, they used timeSVD++ models and used temporal dynamics of user preferences and item features for improvised recommendations.

4.7.2 Variational Autoencoders (VAE)

Due to the drawbacks of collaborative based filtering, the authors in [103] used variational autoencoders to include rating and content information for the recommendation. The novelty in the framework is that the latent distribution for content is learned in latent space rather than observational space. In another paper, the authors demonstrated the enhancement of modeling in CF with side information by using Variational Autoencoders-based Collaborative Filtering (VAE-CF) [92]. The authors [120] proposed a novel healthcare recommender system collaborative variational deep learning (CVDL) to eliminate the limitations of data sparsity and cold-start problem from recommender systems to generate useful recommendations. The paper [121] proposes a deep autoencoder model to learn low and high-dimensional features to remediate data sparsity and data imbalance problems in recommender systems.

To make the task of feature selection simpler and more efficient, this paper [122] proposed a fuzzy entropy-based deep learning recommender system to decrease data dimensionality and eliminating unnecessary noise from data. This paper [123] proposed an end-to-end recommender model by taking several content sources, for example, textual content, graphical content, and others. The authors used a stacked autoencoder to carry out this implementation. In this paper [124], the authors identified the limitations of side infor-

mation in RS. Apart from having extensive detail about the rating, side information also contains a tremendous amount of noise. Hence, the process of feature extraction from such side information becomes challenging. So, the authors proposed a Stacked Discriminative Denoising Autoencoder by merging deep learning with matrix factorization-based RS.

4.8 Findings and Open Issues

Autoencoder is an efficient feature representation learning method that learns feature representations from user–item content features. Autoencoder can be used in recommender systems by learning low-dimensional feature representations at the outer layer or by building the interaction matrix directly in the reconstruction layer. Autoencoder models have a high capability to work with noisy data to learn the complicated and hierarchical structure from the input data. Sparse autoencoders are very efficient for low-dimensional feature extraction from input data using a supervised learning technique. Denoising autoencoders are trained by initializing layers wherein each layer produces input data for the next layer. One of its significant advantages lies in implementing recommender systems as denoising autoencoders can be stacked to decrease the processing errors.

Autoencoders suffer from certain limitations. Autoencoder based Collaborative Filtering (ACF) is incapable of incorporating non-integer ratings, and partially observed features result in low recommendation accuracy. Another major disadvantage of using deep autoencoders is that they are incapable of searching for an optimal solution, and due to high parameter tuning, the time complexity of the training process increases manifold.

4.9 Neural Matrix Factorization (NMF)

Matrix factorization (MF) is one of the most widely and commonly used Collaborative Filtering approaches. Such MF models learn the low-dimensional embeddings of items and users in common latent factor space. The system generates a user–item matrix where the ratings provided for every item by the users are captured. However, due to the problem of data sparsity, the user–item matrix is sparse. Such a sparse matrix is factorized into a dense matrix to make computations simple. This is done by taking the product of two low-rank matrices consisting of user–item embeddings. Upon learning the latent factors, the similarity between user and item is computed. In addition to this, the newly discovered preferences are considered by analyzing the user and item latent factor representations

To learn such latent user–item factors, loss function has to be calculated as given in Eq. 15:

$$min_{v_u v_i} \sum_{(u,i)\in S} (r_{ui} - v_u, v_i)^2 + \lambda\Big(\|v_u\|^2 + \|v_i\|^2\Big). \tag{15}$$

Here, v_i represents the latent factor of item i, v_u represents the latent factor of user u, r_{ui} represents the rating given to item i by user u, and λ represents the item bias to prevent overfitting.

Factorization Machines (FM) are supervised learning models that are used in various prediction problems like regression, classification, and others. These models map random real features into low-dimensional latent feature vector space. They can determine the parameters of the model precisely given a sparse user–item matrix and train with linear complexity. Such characteristic makes FMs ideal for implementing real-world recommender systems.

4.9.1 Deep Factorization Machines (DeepFM)

DeepFM is a model wherein the capabilities of Factorization Machines are combined with deep feature learning to form a novel neural architecture, wherein Factorization Machines and Deep Neural Network (DNN) are integrated to model both low-dimensional and high-dimensional feature vectors.

As the name suggests deepFM is a dual component model, i.e., the *factorization machine* component and *deep learning* component. These components share the same input. These two components are jointly trained for the prediction model computed using Eq. 16.

$$\hat{y} = \text{sigmoid}(y_{\text{FM}} + Y_{\text{DNN}}). \tag{16}$$

Here, $\hat{y}\varepsilon(0, 1)$, y_{FM} represents the output of the FM component, and Y_{DNN} represents the output of the deep learning component.

The main advantages of using this model are no requirement for pre-training the model, its capability to learn both low-dimensional and high-dimensional features. The shared feature embedding of FM and deep components enables the system to avoid extensive feature engineering.

Rather than conventional push–pull methodology among the user and item pairs, the authors proposed LRML (Latent Relational Metric Learning) model to learn latent relations that described every user–item interaction [96]. The authors observed in [114] that although *Weighted Matrix Factorization (*WMF*)* can learn latent features from implicit feedback, these features are still not good enough to train a CNN model. The authors in [95] proposed a novel matrix factorization method with neural network architecture and novel use of binary cross-entropy loss function.

In another work, the authors identified that Factorization Machines (FM) are incapable of modeling the nonlinear aspect of the real-world data [104]. In [54], the authors observed that the recommendation systems primarily relied on initializing the user and item latent feature vectors. In this way, they used deep learning to estimate the initialization in MF for trust-aware social recommendations and to differentiate the neighborhood effect in the user's trusted social circle. This was achieved with the help of the deep learning-based matrix factorization (DLMF) model.

4.10 Findings and Open Issues

Neural matrix factorization provides exemplary results for latent feature models; however, it is influenced by methods involving local graph structure.

One limitation of neural matrix factorization is that to investigate the scope of system architectures, activation functions, regularization techniques, and cross-validation strategies. There is a risk of overfitting, which may lead to erroneous or insignificant insights.

4.11 Neural Collaborative Filtering (NCF)

In recommender systems, the users and items collate together to create a mapping that helps to under the similarity between the two and further recommend items to the user. Since the phenomenon of generating recommendations in a dual process, which involves both user latent factors and item latent factors. Hence, to model this two-way interaction between user and item, Neural Collaborative Filtering (NCF) is used. The scoring function of NCF can be computed using Eq. 17:

$$\hat{r}_{ui} = f\left(U^T \cdot s_u^{\text{user}}, V^T \cdot s_i^{\text{item}} | U, V, \theta\right). \tag{17}$$

Here, $f(\cdot)$ represents the multilayer perceptron, s_u^{user} represents the side information of the user profile, s_i^{item} represents the side information of the item features, and θ represents the parameters of the network.

It becomes easier to combine the neural representation of MF with Multilayer Perceptron to create a model that incorporates the linear characteristics of matrix factorization and nonlinear characteristic of MLP. This results in improved performance of the recommender engine. The cross-entropy loss for implicit and explicit feedback is given by Eq. 18:

$$\mathcal{L} = \; - \sum r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log\left(1 - \hat{r}_{ui}\right). \tag{18}$$

Here, u represents the user, i represents the item, and r_{ui} represents the rating of item i given by user u.

4.11.1 Collaborative Deep Learning (CDL)

This technique is an amalgamation of deep representation of content and collaborative filtering for the ratings. CDL is a deep learning recommender model that learns from the text of the reviews given by the users [114]. It combines the autoencoder and *Click Through Rate* (CTR) to model ratings.

To learn from such review texts, CDL implements *Bag-Of-Words* (BOW) technique. This model has certain limitations, such as it models only item review texts, limiting the capabilities of a recommender system, and due to its usage of the BOW method, it only considers the frequency of words in a text, and not the similarities between different texts. Another limitation of the CDL model is that it considers the order of words in a text which sometimes contradicts the semantics of the text. Figure 15 represents the architecture of SDAE, upon which the SDAE is built. Figure 16 represents the architecture of CDL.

The authors argued that the sole criteria for recommending items were user reviews [102]. However, this often led to data sparsity, hence degrading the recommendation quality. To avoid this, the authors came up with a collaborative deep learning (CDL) model, which simultaneously performed deep learning of content and collaborative filtering for the ratings. In another work, the authors used CDL-based Bayesian deep learning for Recommender Systems model to achieve integrated intelligence that involved both perception and inference in a principled probabilistic framework [89]. In [70], the researchers attempted to improve YouTube recommendations by using a deep collaborative filtering model that efficiently integrated various signals and added layers of depth to the interaction. Some authors in their paper [83] compared the Deep Neural Network (DNN) features with visual features for artwork recommendation and concluded that DNN features outperformed the other.

The authors in [90] presented a novel neural network model, Neighborhood-based Neural Collaborative Filtering (NNCF), which jointly characterized both user–item interactions and neighborhood information for the recommendation.

The authors in [125] indicated that the Collaborative Filtering algorithms primarily deal with low-dimensional and linear interactions between users and items. To address this issue, the authors proposed a novel deep CF recommender algorithm for service recommendations. The authors in [126] used MLP to learn the interaction function to implement the recommender system. The authors [127] proposed a deep multi-criteria collaborative filtering RS to generate accurate multi-criteria recommendations.

4.12 Findings and Open Issues

Collaborative deep learning (CDL) can perform deep learning of content and collaborative filtering for the ratings of

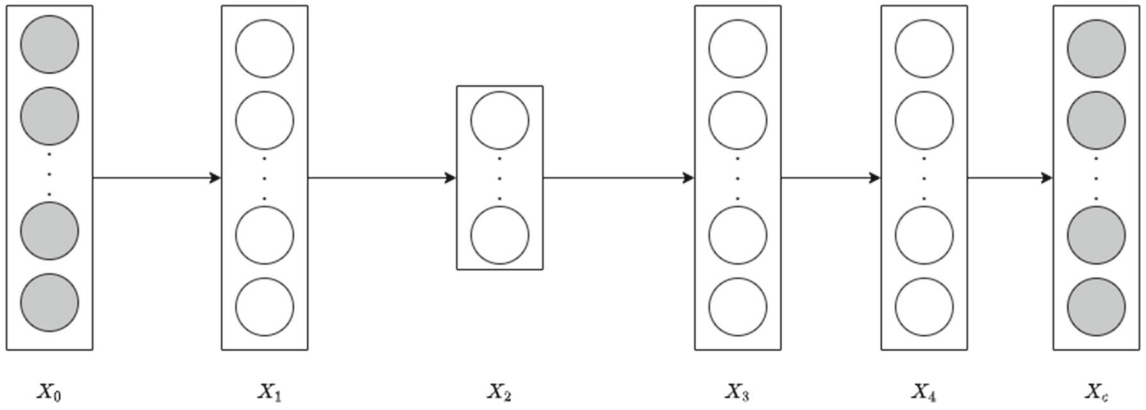

Fig. 15 Architecture of SDAE

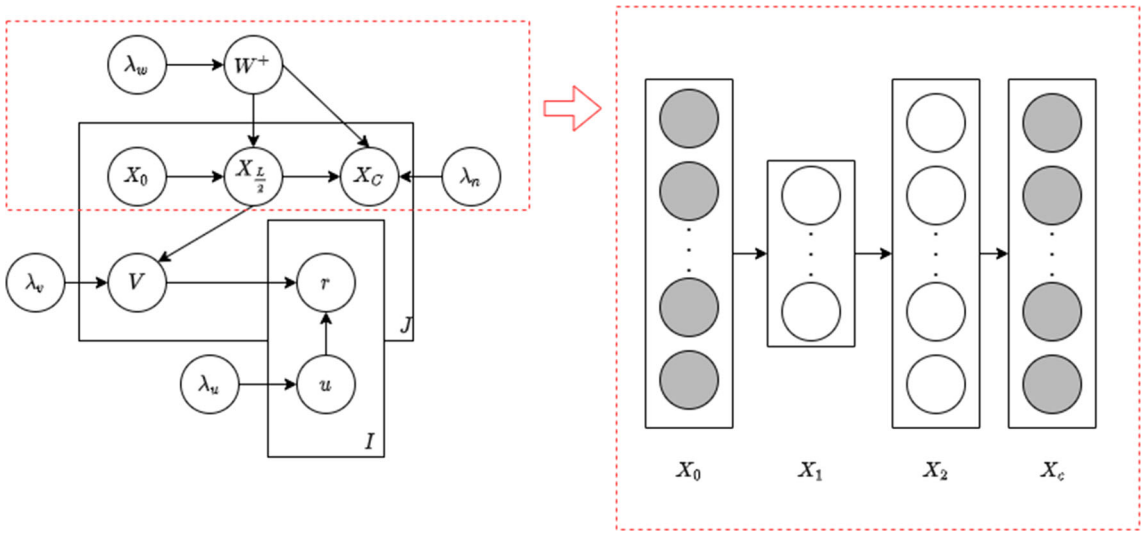

Fig. 16 Architecture of CDL

items. This allows the model to balance the effects of side information and interaction history.

One significant limitation of the CDL model is that it cannot apprehend contextual information with word embeddings and convolutional filters.

4.13 Deep Belief Network (DBN)

It is a class of deep learning where several layers of hidden units are stacked together to form a deep net. The hidden units present in each layer are not directly connected. They can be viewed as a class of unsupervised networks like *Restricted Boltzmann Machines (RBM)* [128]. Authors [129] observed that classic neural networks suffer from the problem of optimization. DBN was introduced to solve this problem [130]. This technique integrates supervised and unsupervised learning by implementing a local search to get an optimized result and learn the data distribution without prior knowledge [129]. DBN employs a stacked RBM structure for extracting deep features of data [69]. Thus, it can be viewed as a generative probabilistic model [36]. In DBN, the hidden layers are trained in a bottom-up manner. This pre-training of the layers is termed as the *pre*-train stage, and upon the addition of the objective layer, the training of the model becomes supervised [131]. Figure 17 represents the architecture of the Deep Belief Network.

The standard RBM is given by Eq. 19

$$E(\vec{u}, v) = -\sum_{i=1}^{p}\sum_{j=1}^{q} w_{i,j}u_i v_j - \sum_{i=1}^{p} x_i u_i - \sum_{j=1}^{q} y v_j, \quad (19)$$

where $\vec{u}$ is the visible unit vector and $\vec{v}$ is the hidden unit vector. w represents the weight matrix and $\vec{x}$ is the visible bias vector and $\vec{y}$ is the hidden bias vector.

In video recommender systems, generating recommendations for videos becomes complicated because extracting intricate features from graphics is difficult. The workflow of such a recommender system is presented in Fig. 18. To make

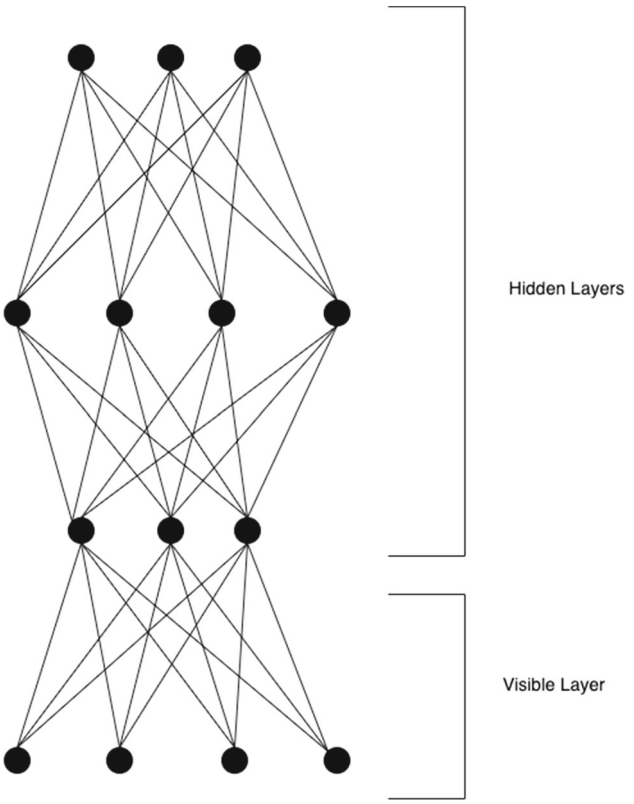

Fig. 17 Architecture of deep belief network (DBN)

this task simpler, authors in [69] divided the user–item matrix into 0–1 matrices and feed them as input to the DBN model. Hence, post-training, the difference between trained features and raw features becomes prominent. Afterward, UBCF calculates the similarity between neighboring users to generate recommendations.

4.14 Findings and Open Issues

One of the most significant advantages of DBN is that Deep Belief Network has been successfully employed to perform extensive feature engineering. It is useful in extracting hidden and useful features from audio data [106]. Another significant advantage of using DBN is that it is also used for dimensionality reduction. DBN has also proved to help solve the data sparsity problem, which leads to the cold-start problem and generation of poor-quality recommendations.

However, DBN comes with several limitations as well. Due to extensive parameter initialization, the DBN model is challenging to train. Another major disadvantage of DBN is that it cannot learn the representation of features from labeled and unlabeled data[132].

The training of the Deep Belief Network model can be further optimized to improve the recommendations.

4.15 Recurrent Neural Network (RNN)

Unlike feed-forward deep neural network, the layers present in a recurrent neural network form a graph, and hence the layers interact with each other. The graphs thus formed can either be cyclic or acyclic [107]. Essentially RNN captures the sequential data temporally, which is used for training the model [133]. The RNN takes the input and computes the weighted sums as given in Eq. 20. The architecture of the recurrent neural network is presented in Fig. 19.

$$z_{hj} = \sum_{i=1}^{n_x} x_i w_{ij}^{xh} + \sum_{i=1}^{n_h} h_i w_{ij}^{hh}, \tag{20}$$

where n_x are the input nodes and n_h are the hidden nodes.

The weights between input and hidden nodes are given by w^{xh}, and w^{hh} gives the weights between the hidden nodes. Using Eq. 21, the activation values can be calculated.

$$h_j = \tanh\left(z_{hj}\right) = \tanh\left(\sum_{i=1}^{n_x} x_i w_{ij}^{xh} + \sum_{i=1}^{n_h} h_i w_{ij}^{hh}\right). \tag{21}$$

The weighted sums of the output layer can be calculated as given in Eq. 22

$$z_{yj} = \sum_{i=1}^{n_h} h_i w_{ij}^{hy}. \tag{22}$$

4.15.1 Gated Recurrent Unit for Recommender System (GRU4REC)

GRU4REC is a session-based recommender system proposed by [134]. The input given to the model is an encoding of 1 of N, where N represents the number of items. The output of the system determines the possibility of an item to move to the subsequent session. If the item is active in the current session, the coordinate is assigned value 1, otherwise 0. The architecture of GRU4REC is given in Fig. 20.

The ranking loss function for this model is computed using Eq. 23:

$$\mathcal{L}_s = \frac{1}{S} \sum_{j=1}^{S} \sigma\left(\hat{r}_{sj} - \hat{r}_{si}\right) + \sigma\left(\hat{r}_{sj}^2\right). \tag{23}$$

Here, S represents the sample size, $\hat{r}_{si}$ represents the scores on negative item i at session s, $\hat{r}_{sj}$ represents the scores on positive item j at session s, and σ represents the logistic sigmoid function.

In GRU-based recurrent neural networks (RNN), the input given to the system depicts the current state of the system, and the output of the system depicts the subsequent event of the system. Unlike in GRU4REC, where 1-of-N encoding is

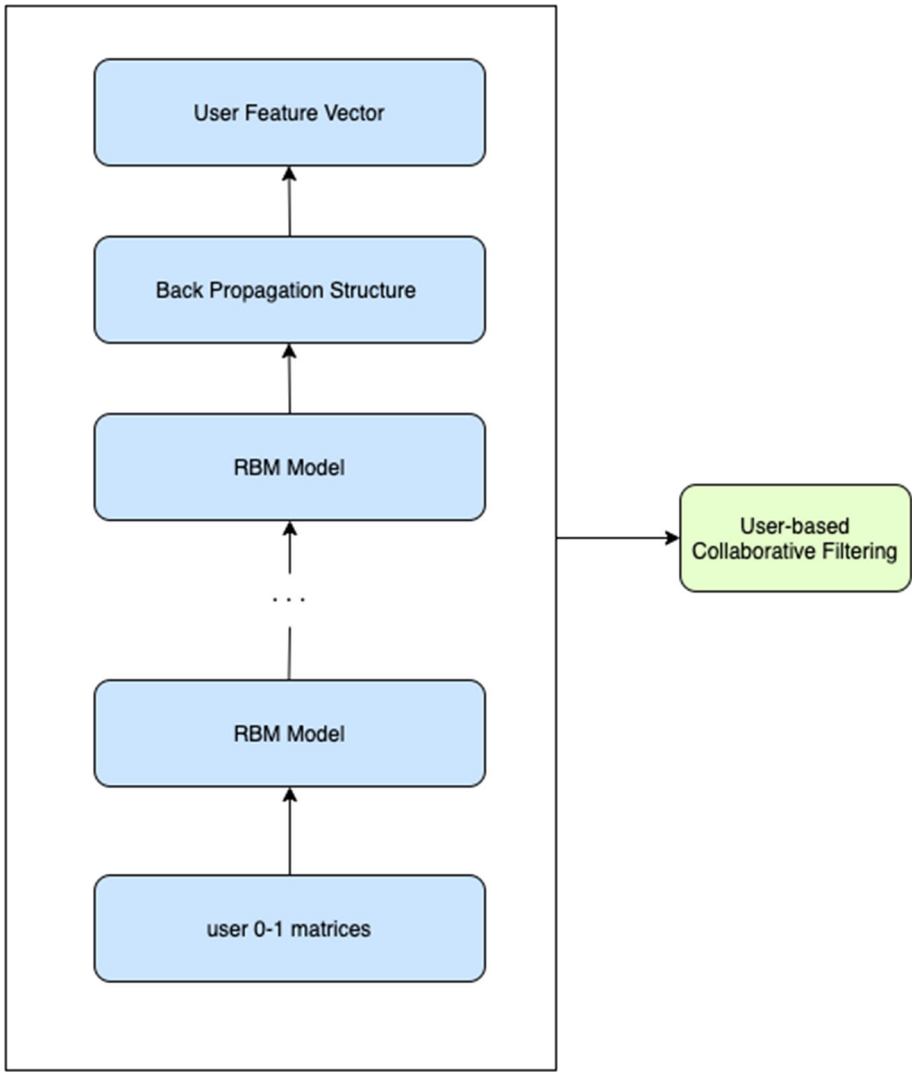

Fig. 18 Video recommender system implementing DBN and CF

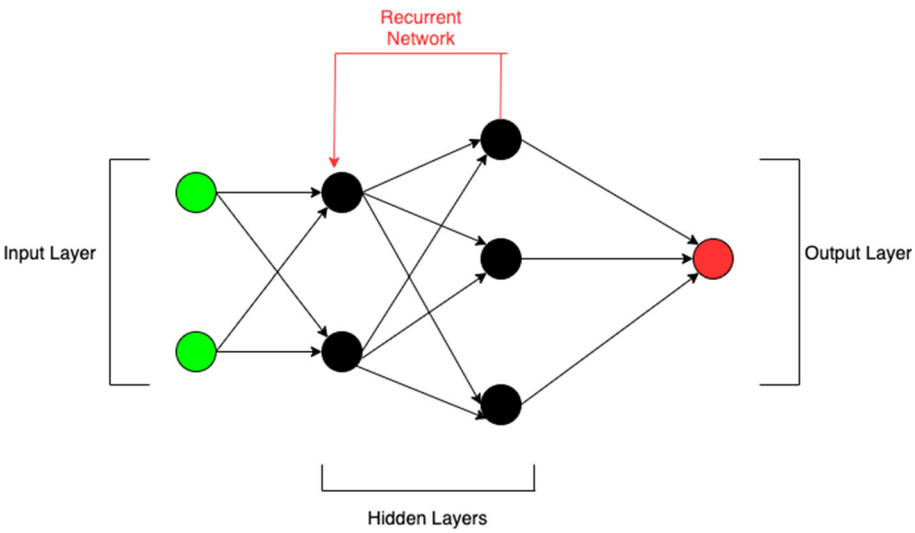

Fig. 19 Architecture of recurrent neural network

used, in GRU based RNN for session-based recommender systems, a weighted sum of the representations is used. In such cases, the events which occurred in the past are given as discounted when fed as an input. Hence, to make the system stable, the input vector is normalized.

When multiple feed-forward layers are added to the system, the output predicts whether the item will be included in the next session or not. The addition of multiple GRU layers uses the output of the previous layer as the input to the next layer. This deep connection of layers improves the performance of the system.

In some research studies, the side information was included to make the recommendations efficient. It was observed in [10] that recommender systems based on collaborative filtering could be enhanced by including side information like natural language reviews. To carry out this task, the authors used the bag-of-words technique along with RNN. In [67], according to the authors, current recommendation techniques using RNN only took into account the user's past activities and did not consider the essential side information. To cater to this problem, they used contextual recurrent neural networks (CRNN) to include the side information to give recommendations. Some authors considered the temporal while making the recommendations and studied its effects. To model the temporal behavior in recommendation systems, a group of researchers [21] introduced the Temporal Deep Semantic Structured Model (TDSSM) based on RNN, where they used long-term static and short-term temporal user choices to improvise the performance of the recommender system.

Similarly, researchers in [62] observed that the conventional POI recommender systems did not consider the temporal aspect while generating the recommendation. Hence, they came up with the idea to capture sequential check-in data of users and used RNN based deep neural network to provide recommendations. Catering to the same issue, the authors in [85] observed that a notable problem with recommender systems is that they do not encompass the context of time. So, they came up with the idea of using RNN and including the temporal shift while performing recommendations.

In a few research papers, the RNN technique was applied for session recommendations. In [37], the authors used the recurrent neural network for session-based recommendations. They incorporated two procedures for improving the model. The first one was data augmentation, and the second was to observe shifts in the input data distribution. Similarly, it was observed in [38] that in the real-life recommender systems, the session-based recommendations were based solely on the clicks of a user session. To solve it, the authors introduced several parallel RNN (p-RNN) frameworks to model sessions depending on the clicks and the features of the clicked items. The researchers in [41] proposed a model for session-based recommendations, where the recommendations were made at the starting of the sessions, which avoided the cold-start problem. In [72], the authors used Long Short-Term Memory (LSTM), a neural network-based RNN technique for quotes recommendation. Similarly, to make recommendations using the data content describing the items, the authors proposed Ask Me Any Rating (AMAR), which used Long Short-Term Memory (LSTM) networks to simultaneously learn two embeddings representing the items to be recommended and user preferences [65]. The concept of LSTM was also used in [41], where the authors proposed Recurrent Recommender Networks (RRN) to incorporate dynamic recommendations using Long Short-Term Memory (LSTM) autoregressive model.

In some papers, the problem of data sparsity was addressed and rectified using RNN. The authors in [40] introduced a ReLaVaR model that addressed data sparsity by considering the network recurrent units as stochastic latent variables with a former distribution performed over them. Similarly, it was observed that personalizing Adaptive User-Interfaces (AUIs) became difficult with several interaction modules as the item-user matrix is sparse. The authors in [84] used architecture consisting of RNN that performed sequential recommendations of content and control elements to solve this problem. Several other independent research works were performed using RNN. The authors in [46] analyzed that in documents, the semantic similarities between texts were not considered for making recommendations. They were able to effectively model the content of the abstracts of the documents using linear regression based on RNN. In another work, it was observed that the evolution and drifting of features might take place over time as users interact with different items. Hence, the authors utilized the recurrent neural network to learn a representation of impact from drift, evolution, and co-evolution of user and item attributes to make recommendations [82]. To improve the performance of multi-task learning, the authors used deep RNN to encode the text sequence into a latent vector trained on the collaborative filtering procedure [49].

Performing recommendations using sentiment analysis of short texts like sentences did not yield correct results due to less contextual information. Hence, the authors used the deep learning technique recurrent neural networks to solve this problem [99]. To find relevant citations or related work, the authors came up with Neural Citation Network (NCN) model to provide a curated list of high-quality candidates from a short passage of text [63]. For generating sequential recommendations, the authors in [13] came up with user-based RNN, which allowed generating personalized suggestions. In another work, to find relevant research articles, researchers usually rely only on the keyword-based search or by following citations. In the paper, the authors analyzed user activities to provide recommendations [65]. For song recommendations, it was observed by the authors that the combination

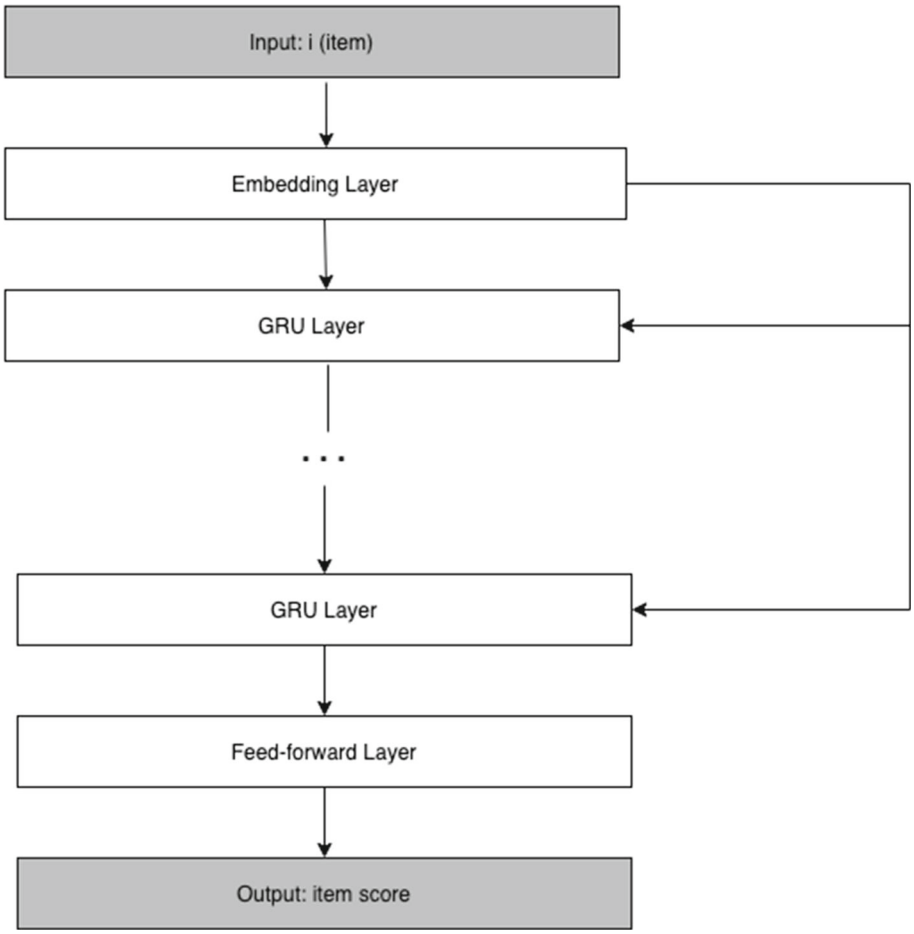

Fig. 20 Architecture of GRU4REC

factor of lyrics and genre was not included for the recommendation [45]. Hence, they used RNN to predict the user's next song of interest-based on the similarity factor. Some authors analyzed that the recommendation quality of rating predictions in traditional systems had scope for improvement [75]. Hence, they proposed a Neural Rating and Tips generation (NRT) framework, which could predict accurate ratings and create conceptual tips and strong linguistic quality. In another work, the authors addressed the limitation of efficient methods in the e-learning area [74].

To improve the recommendations, the authors used Gated Recurrent Unit (GRU) neural networks, a type of RNN. In another work, the authors deployed TARMF (Topical Attention Regularized Matrix Factorization) model that co-learned user and item details from ratings and customer reviews by optimizing matrix factorization and an attention-based GRU network [19].

To fully exploit and understand the user sentiments present in the review texts, the authors in this [135] paper implemented a deep learning-based neural network model, SDRA. Gated Recurrent Unit (GRU) is a type of Long Short-Term Memory (LSTM) model, which is an extension of an RNN model. GRU is a much simpler version of the LSTM model; however, the parameter setting in GRU for better accuracy is much easier as compared to LSTM [131].

4.16 Findings and Open Issues

Recurrent neural network is efficient for using sequential data. Also, for subsuming side information like time, logs, etc., RNNs are beneficial.

An important open issue is for music recommendation systems, the recommendations can be improved by performing temporal analysis of the features of music using recurrent neural network, and fascinating characteristics of music can be uncovered by interpreting automatically learned features. Another scope is to explore recurrent neural networks (RNNs) with bi interaction pooling to model sequential data. To further improve the performance of LSTM models, attention-based memory networks can be explored. The possibility of using a sequence-to-sequence (seq 2seq) learning framework can be explored, aiming to create a convincing recommendation description to aid customers in making a better purchase.

An important limitation of RNN that can also be considered as an open issue is that although applying the

neighborhood approach in RNN can lead to good results, but it is also suggested that some baselines in recent research studies are not well-justified and correctly evaluated.

Gated Recurrent Unit primarily addresses the problem of the vanishing gradient. GRUs often combine several memories and gates to record sequential activities. Another advantage of this model is that it can perform rating prediction and create general tips with linguistic quality. This model efficiently stores contextual information for rendering user–item latent features into a brief sentence.

4.17 Hybrid Networks

4.17.1 word2vec and Convolutional Neural Networks

The authors in [79] worked on a blog recommendation. They stated that due to a large number of blogs surfacing each day, it was crucial to recommend the right blogs to the right users. For this, they came up with the Boosted Inductive Matrix Completion method. Hence, they used the side information of users and blogs for adequate recommendations.

4.17.2 RNN and CNN

The authors worked toward improving quote recommendation. For this, first, they used RNN to model the tweet sequences, and then they used CNN for mapping tweets to intermediate vectors [71]. The authors in [60] observed that for POI recommendations, modeling multi-source information was one-dimensional. Hence, they proposed a Deep Context-aware POI Recommendation (DCPR) model, which consisted of different layers. One layer performed feature mining using convolutional neural network; the second layer was based upon recurrent neural network, and the third layer modeled these together. In another work [24], it was observed that the main challenge with news recommendation was to recommend the latest news articles to the users. To solve this issue, the authors presented an improved session-based recurrent neural network (RNN) model, which studied the users' history of reading news articles and thus made recommendations. The authors in [86] used the dual-regularized matrix factorization technique, which consisted of a multilayered neural network model by simultaneously implementing a convolutional neural network and gated recurrent neural network, to create an independently distributed rendering of contents of users and items.

4.17.3 CNN and Stacked Denoising Autoencoder

To address data sparsity, the authors used the Probabilistic model of the Hybrid Deep collaborative filtering model (PHD) and combined a stacked denoising autoencoder and a CNN alongside the auxiliary side information to extract users and items' latent factors [94].

4.18 Findings and Open Issues

Several deep learning-based recommendation methods employ more than one deep learning techniques. Deep neural networks give the system an edge over every other method by combining numerous neural computations and complementing each other to form a more efficient hybrid model. Each combination of a neural network technique is very specific and is tailored to each use case, as every problem statement requires a unique set of neural network operations.

An open issue is exploring all the possible combinations of deep learning techniques since many combinations have not been exploited yet and may provide useful insight into the increasing efficiency of recommender systems.

5 Conclusion and Future Work

In this paper, a literature survey has been performed on deep learning in recommender systems. It was observed that deep learning provides a considerable advantage in performance, especially when data is available in abundance. It was also observed that by using deep learning, we could extract feature representations that are much more comprehensive and better performing than the features extracted using traditional feature engineering. Deep learning can also incorporate side information like time using models that can incorporate temporal data like LSTM.

However, deep learning only works well when data is available in abundance. In situations where data sparsity exists, deep learning is not the best fit. Also, deep learning requires extensive hyperparameter tuning to achieve the desired accuracy. Quintessentially, hyperparameter tuning is a problem that plagues all machine learning algorithms, and deep learning models have a lot of additional hyperparameters making the problem even worse. Poor tuning can lead to overfitting or underfitting of data points. As future work, these limitations of deep learning can be worked upon using the deep learning models specified in Sect. 4.

Deep learning is a great tool to exploit the ever-explosive data available online and has shown to improve performance in all domains of recommender systems. However, creating and training deep learning models is a sensitive process, and proper hyperparameter tuning is required to get the best results out of it. Deep learning also lacks interpretability and often functions as a black box. Despite these limitations, deep learning has become an integral part of most state-of-the-art recommender systems and is increasingly becoming more relevant with the advent of big data.

References

1. Kim, K.; Ahn, H.: A recommender system using GA K-means clustering in an online shopping market. Expert Syst. Appl. **34**, 1200–1209 (2008). https://doi.org/10.1016/j.eswa.2006.12.025
2. Meteren, R. Van; Someren, M. Van: Using content-based filtering for recommendation. ECML/MLNET Work. Mach. Learn. New Inf. Age. (2000). https://doi.org/10.1125/5743
3. Vekariya, V.; Kulkarni, G.R.: Hybrid recommender systems: survey and experiments. In: 2012 2nd International Conference on Digital Information and Communications Technology its APPL. DICTAP 2012, pp. 469–473 (2012). https://doi.org/10.1109/DICTAP.2012.6215409
4. Zhang, X.; Liu, H.; Chen, X.; Zhong, J.; Wang, D.: A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness. Inf. Sci. (Ny) **519**, 306–316 (2020). https://doi.org/10.1016/j.ins.2020.01.044
5. Peterson J.J.; Yahyah M.; Lief K.; H.N.: Predictive distributions for constructing the ICH Q8 design space. In: Comprehensive Quality by Design for Pharmaceutical Product Development and Manufacture, pp. 55–70 (2017). https://doi.org/10.1145/1143844.1143865
6. Zhu, X.: Semi-supervised learning literature survey contents. Sci. York. **10**, 10 (2008)
7. Grant, P.: Assessment and selection. Bus. Giv. (2014). https://doi.org/10.1057/9780230355033.0018
8. Liu, Y.H.; Wang, X.K.; Liu, P.X.; Zheng, J.P.; Shu, C.Y.; Pan, G.S.; Luo, J.: Bin: Modification on the tribological properties of ceramics lubricated by water using fullerenol as a lubricating additive. Sci. China Technol. Sci. **55**, 2656–2661 (2012). https://doi.org/10.1007/s11431-012-4938-y
9. Zhang, Y.; Ai, Q.; Chen, X.; Croft, W.B.: Joint representation learning for top-N recommendation with heterogeneous information sources. In: International Conference on Information and Knowledge Management. Proceedings of Part F1318, 1449–1458 (2017). https://doi.org/10.1145/3132847.3132892
10. Almahairi, A.; Kastner, K.; Cho, K.; Courville, A.: Learning distributed representations from reviews for collaborative filtering. In: RecSys 2015 – Proceedings of 9th ACM Conference on Recommendation Systems, pp. 147–154 (2015). https://doi.org/10.1145/2792838.2800192
11. Elkahky, A.M.; Song, Y.; He, X.: A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: Proceedings of 24th International Conference on World Wide Web - WWW'15, pp. 278–288 (2015). https://doi.org/10.1145/2736277.2741667
12. Seo, S.; Huang, J.; Yang, H.; Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of Elsevier ACM Conference on Recommendation Systems - RecSys'17, pp. 297–305 (2017). https://doi.org/10.1145/3109859.3109890
13. Donkers, T.; Loepp, B.; Ziegler, J.: Sequential user-based recurrent neural network recommendations. In: Proceedings of Elsevier ACM Conference on Recommendation Systems - RecSys'17, pp. 152–160 (2017). https://doi.org/10.1145/3109859.3109877
14. Purkaystha, B.; Datta, T.; Islam, M.S.; Marium-E-Jannat: Product recommendation: a deep learning factorization method using separate learners. In: 20th The International Conference on Information and Computer Technologies ICCIT 2017. 2018-Janua, pp. 1–5 (2018). https://doi.org/10.1109/ICCITECHN.2017.8281852
15. Zheng, L.; Noroozi, V.; Yu, P.S.: Joint Deep Modeling of Users and Items Using Reviews for Recommendation, pp. 425–433 (2017). https://doi.org/10.1145/3018661.3018665
16. Paradarami, T.K.; Bastian, N.D.; Wightman, J.L.: A hybrid recommender system using artificial neural networks. Expert Syst. Appl. **83**, 300–313 (2017). https://doi.org/10.1016/j.eswa.2017.04.046
17. Wang, X.; He, X.; Nie, L.; Chua, T.-S.: Item Silk Road: Recommending Items from Information Domains to Social Users, pp. 185–194 (2017). https://doi.org/10.1145/3077136.3080771
18. Zhu, H.; Li, X.; Zhang, P.; Li, G.; He, J.; Li, H.; Gai, K.: Learning Tree-based Deep Model for Recommender Systems. (2018). https://doi.org/10.1145/3219819.3219826
19. Lu, Y.; Dong, R.; Smyth, B.: Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. Www, pp. 773–782 (2018). https://doi.org/10.1145/3178876.3186158
20. Oh, K.J.; Lee, W.J.; Lim, C.G.; Choi, H.J.: Personalized news recommendation using classified keywords to capture user preference. In: The International Conference on Advance and Communcations Technologies ICACT, pp. 1283–1287 (2014). https://doi.org/10.1109/ICACT.2014.6779166
21. Song, Y.; Elkahky, A.M.; He, X.: Multi-rate deep learning for temporal recommendation. In: SIGIR 2016 - Proceedings of 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 909–912 (2016). https://doi.org/10.1145/2911451.2914726
22. Kumar, V.; Khattar, D.; Gupta, S.; Gupta, M.; Varma, V.: User profiling based deep neural network for temporal news recommendation. IEEE International Conference on Data Mining Work. ICDMW. 2017-Novem, pp. 765–772 (2017). https://doi.org/10.1109/ICDMW.2017.106
23. Cao, S.; Yang, N.; Liu, Z.: Online news recommender based on stacked auto-encoder. In:Proceedings of - 16th IEEE/ACIS International Conference on Computational Science ICIS 2017, pp. 721–726 (2017). https://doi.org/10.1109/ICIS.2017.7960088
24. Park, K.; Lee, J.; Choi, J.: Deep Neural Networks for News Recommendations. In: Proceedings of 2017 ACM Conference on Information and Knowledge Management - CIKM'17, pp. 2255–2258 (2017). https://doi.org/10.1145/3132847.3133154
25. Shani, G.; Heckerman, D.; Brafman, R.I.; Liebman, E.; Saar-Tsechansky, M.; Stone, P.; Zhao, X.; Zhang, L.; Ding, Z.; Yin, D.; Zhao, Y.; Tang, J.; Feng, J.; Li, H.; Huang, M.; Liu, S.; Ou, W.; Wang, Z.; Zhu, X.; Cai, Q.; Filos-Ratsikas, A.; Tang, P.; Zhang, Y.; Zheng, G.; Zhang, F., Zheng, Z.; Xiang, Y.; Yuan, N.J.; Xie, X.; Li, Z.; Mahmood, T.; Ricci, F.; Taghipour, N.; Kardan, A.; Ghidary, S.S.; Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J.; Chapelle, O.; Wu, Q.; Wang, H.; Hong, L.; Shi, Y.; Zhou, M.; Ding, Z.; Tang, J.; Yin, D.; Du, N.; Wang, Y.; He, N.; Sun, J.; Song, L.; Kapoor, K.; Subbian, K.; Srivastava, J.; Schrater, P.; Zhao, X.; Xia, L.; Zhang, L.; Ding, Z.; Yin, D.; Tang, J.; Xia, L.; Tang, J.; Yin, D.; Chen, S.-Y.; Yu, Y.; Da, Q.; Tan, J.; Huang, H.-K.; Tang, H.-H.; Shi, J.-C.; Yu, Y.; Da, Q.; Chen, S.-Y.; Zeng, A.-X.: DRN: A Deep Reinforcement Learning Framework for News Recommendation. In: Proceedings of 2018 World Wide Web Conf. World Wide Web. 6, 113–120 (2018). https://doi.org/10.1145/3178876.3185994
26. Yi, B.; Shen, X.; Zhang, Z.; Shu, J.; Liu, H.: Expanded autoencoder recommendation framework and its application in movie recommendation. In: Ski. 2016 - 2016 10th International Conference on Software, Knowledge, Information Managements Applcations, pp. 298–303 (2017). https://doi.org/10.1109/SKIMA.2016.7916236
27. Vuurens, J.B.P.; Larson, M.; De Vries, A.P.: Exploring deep space: Learning personalized ranking in a semantic space. In: ACM International Conference on Proceeding Series, 15-Septemb, pp. 23–28 (2016). https://doi.org/10.1145/2988450.2988457
28. Zhao, C.; Shi, J.; Jiang, T.; Zhao, J.; Chen, J.: Application of deep belief nets for collaborative filtering. 2016 16th Int. Symp.

Commun. Inf. Technol. Isc. 2016. 201–205 (2016). https://doi.org/10.1109/ISCIT.2016.7751621
29. Sottocornola, G.; Stella, F.; Zanker, M., Canonaco, F.: Towards a deep learning model for hybrid recommendation. In:Proceedings of Int. Conf. Web Intell. - WI'17. 1260–1264 (2017). https://doi.org/10.1145/3106426.3110321
30. Taheri, S.M.; Irajian, I.: DeepMovRS: A unified framework for deep learning-based movie recommender systems. 2018 6th Iran. Jt. Congr. Fuzzy Intell. Syst. 200–204 (2018). https://doi.org/10.1109/CFIS.2018.8336633
31. Fu, M.; Qu, H.; Yi, Z.; Lu, L.; Liu, Y.: A Novel Deep Learning-Based Collaborative Filtering Model for Recommendation System. IEEE Trans. Cybern. 1–13 (2018). https://doi.org/10.1109/TCYB.2018.2795041
32. Yuan, J.; Shalaby, W.; Korayem, M.; Lin, D.; AlJadda, K.; Luo, J.: Solving Cold-Start Problem in Large-scale Recommendation Engines: {A} Deep Learning Approach. CoRR. abs/1611.0, 1901–1910 (2016)
33. Chen, W.; Zhang, X.; Wang, H.; Xu, H.: Hybrid deep collaborative filtering for job recommendation. 2017 2nd IEEE Int. Conf. Comput. Intell. Appl. ICCIA 2017. 2017-Janua, 275–280 (2017). https://doi.org/10.1109/CIAPP.2017.8167222
34. Nguyen, T.T.; Lauw, H.W.: Collaborative Topic Regression with Denoising AutoEncoder for Content and Community Co-Representation. In:Proceedings of 2017 ACM Conference on Information and Knowledge Management - CIKM'17. 2231–2234 (2017). https://doi.org/10.1145/3132847.3133128
35. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 1339–1351 (2017). https://doi.org/10.1016/j.eswa.2016.09.040
36. Zhang, Y.; Yin, H.; Huang, Z.; Du, X.; Yang, G.; Lian, D.: Discrete Deep Learning for Fast Content-Aware Recommendation. In:Proceedings of Elev. ACM Int. Conf. Web Search Data Min. - WSDM'18. 717–726 (2018). https://doi.org/10.1145/3159652.3159688
37. Tan, Y.K.; Xu, X.; Liu, Y.: Improved Recurrent Neural Networks for Session-based Recommendations (2016). https://doi.org/10.1145/2988450.2988452
38. Hidasi, B.; Quadrana, M.; Karatzoglou, A.; Tikk, D.: Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In:Proceedings of 10th ACM Conference Recommendations Systems - RecSys'16, pp. 241–248 (2016). https://doi.org/10.1145/2959100.2959167
39. Greenstein-Messica, A.; Rokach, L.; Friedman, M.: Session-Based Recommendations Using Item Embedding. In:Proceedings of 22nd Int. Conf. Intell. User Interfaces - IUI'17. 629–633 (2017). https://doi.org/10.1145/3025171.3025197
40. Chatzis, S.P.; Christodoulou, P.; Andreou, A.S.: Recurrent Latent Variable Networks for Session-Based Recommendation. In:Proceedings of 2nd Workshop on Deep Learning based Recommender System - DLRS 2017. 38–45 (2017). https://doi.org/10.1145/3125486.3125493
41. Ruocco, M.; Skrede, O.S.L.; Langseth, H.: Inter-Session Modeling for Session-Based Recommendation (2017). https://doi.org/10.1145/3125486.3125491
42. Wang, X.; Wang, Y.: Improving Content-based and Hybrid Music Recommendation using Deep Learning. In: Proceedings of ACM Int. Conf. Multimed. - MM'14. 627–636 (2014). https://doi.org/10.1145/2647868.2654940
43. Chiliguano, P.; Fazekas, G.: Hybrid music recommender using content-based and social information. ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc. 2016-May, 2618–2622 (2016). https://doi.org/10.1109/ICASSP.2016.7472151
44. Oramas, S.; Nieto, O.; Sordo, M.; Serra, X.: A Deep Multimodal Approach for Cold-start Music Recommendation. In:Proceedings of 2nd Workshop on Deep Learning based Recommender System - DLRS 2017. 32–37 (2017). https://doi.org/10.1145/3125486.3125492
45. Jiang, M.; Yang, Z.; Zhao, C.: What to play next? A RNN-based music recommendation system. Conf. Rec. 51st Asilomar Conf. Signals, Syst. Comput. ACSSC 2017. 2017-Octob, 356–358 (2018). https://doi.org/10.1109/ACSSC.2017.8335200
46. Florez, O.U.; Nachman, L.: Deep Learning of Semantic Word Representations to Implement a Content-based Recommender for the RecSys Challenge' 14. 1–5
47. Kim, D.; Park, C.; Oh, J.; Yu, H.: Deep hybrid recommender systems via exploiting document context and statistics of items. Inf. Sci. (Ny) **417**, 72–87 (2017). https://doi.org/10.1016/j.ins.2017.06.026
48. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H.: Convolutional Matrix Factorization for Document Context-Aware Recommendation. In:Proceedings of 10th ACM Conference Recommendations Systems. - RecSys'16. 233–240 (2016). https://doi.org/10.1145/2959100.2959165
49. Bansal, T.; Belanger, D.; McCallum, A.: *Ask the GRU*. In:Proceedings of 10th ACM Conference Recommendations Systems. - RecSys'16. 107–114 (2016). https://doi.org/10.1145/2959100.2959180
50. Wang, X.; Yu, L.; Ren, K.; Tao, G.; Zhang, W.; Yu, Y.; Wang, J.: Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors' Demonstration. In:Proceedings of 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD'17, pp. 2051–2059 (2017). https://doi.org/10.1145/3097983.3098096
51. Lei, C.; Liu, D.; Li, W.; Zha, Z.-J.; Li, H.: Comparative Deep Learning of Hybrid Representations for Image Recommendations, pp. 2545–2553 (2016). https://doi.org/10.1109/CVPR.2016.279
52. Wang, J.; Kawagoe, K.: Ukiyo-e Recommendation based on Deep Learning For Learning Japanese Art and Culture. In:Proceedings of 2017 International Conference on Information Syst. Data Min. - ICISDM'17. 119–123 (2017). https://doi.org/10.1145/3077584.3077612
53. Peska, L.; Trojanova, H.: Towards Recommender Systems for Police Photo Lineup. In:Proceedings of 2nd Workshop on Deep Learning based Recommender System - DLRS 2017. 19–23 (2017). https://doi.org/10.1145/3125486.3125490
54. Deng, S.; Huang, L.; Xu, G.; Wu, X.; Wu, Z.: On Deep Learning for Trust-Aware Recommendations in Social Networks. IEEE Trans. Neural Networks Learn. Syst. **28**, 1164–1177 (2017). https://doi.org/10.1109/TNNLS.2016.2514368
55. Dang, Q.V.; Ignat, C.L.: DTrust: A Simple Deep Learning Approach for Social Recommendation. In:Proceedings of - 2017 IEEE 3rd Int. Conf. Collab. Internet Comput. CIC 2017. 2017-Janua, 209–218 (2017). https://doi.org/10.1109/CIC.2017.00036
56. Rafailidis, D.; Crestani, F.: Recommendation with Social Relationships via Deep Learning. In:Proceedings of ACM SIGIR Int. Conf. Theory Inf. Retr. - ICTIR'17. 151–158 (2017). https://doi.org/10.1145/3121050.3121057
57. Tomar, A.; Godin, F.; Vandersmissen, B.; De Neve, W.; Van De Walle, R.: Towards Twitter hashtag recommendation using distributed word representations and a deep feed forward neural network. In:Proceedings of 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014. 362–368 (2014). https://doi.org/10.1109/ICACCI.2014.6968557
58. Zuo, Y.; Zeng, J.; Gong, M.; Jiao, L.: Tag-aware recommender systems based on deep neural networks. Neurocomputing. **204**, 51–60 (2016). https://doi.org/10.1016/j.neucom.2015.10.134
59. Xu, Z.; Chen, C.; Lukasiewicz, T.; Miao, Y.; Meng, X.: Tag-Aware Personalized Recommendation Using a Deep-Semantic Similarity Model with Negative Sampling. In:Proceedings of 25th ACM

International Conference on Information and Knowledge Management - CIKM'16. 1921–1924 (2016). https://doi.org/10.1145/2983323.2983874
60. Wang, F.; Qu, Y.; Zheng, L.; Lu, C.T.; Yu, P.S.: Deep and Broad Learning on Content-Aware POI Recommendation. In:Proceedings of - 2017 IEEE 3rd Int. Conf. Collab. Internet Comput. CIC 2017. 2017-Janua, 369–378 (2017). https://doi.org/10.1109/CIC.2017.00054
61. Yin, H.; Wang, W.; Wang, H.; Chen, L.; Zhou, X.: Spatial-Aware Hierarchical Collaborative Deep Learning for POI Recommendation. IEEE Trans. Knowl. Data Eng. **29**, 2537–2551 (2017). https://doi.org/10.1109/TKDE.2017.2741484
62. Xia, B.; Li, Y.; Li, Q.; Li, T.: Attention-based recurrent neural network for location recommendation. In:Proceedings of 2017 12th Int. Conf. Intell. Syst. Knowl. Eng. ISKE 2017. 2018-Janua, 1–6 (2018). https://doi.org/10.1109/ISKE.2017.8258747
63. Ebesu, T.; Fang, Y.: Neural Citation Network for Context-Aware Citation Recommendation. In:Proceedings of 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'17. 1093–1096 (2017). https://doi.org/10.1145/3077136.3080730
64. Huck-Fries, V.; Wiegand, F.; Klinker, K.; Wiesche, M.; Krcmar, H.: Reranking-based Recommender System with Deep Learning. Inform. 2017. 585–596 (2017). https://doi.org/10.18420/in2017
65. Hassan, H.A.M.: Personalized Research Paper Recommendation using Deep Learning. In:Proceedings of 25th Conf. User Model. Adapt. Pers. - UMAP'17. 327–330 (2017). https://doi.org/10.1145/3079628.3079708
66. Suglia, A.; Greco, C.; Musto, C.; de Gemmis, M.; Lops, P.; Semeraro, G.: A Deep Architecture for Content-based Recommendations Exploiting Recurrent Neural Networks. In:Proceedings of 25th Conf. User Model. Adapt. Pers. - UMAP'17. 202–211 (2017). https://doi.org/10.1145/3079628.3079684
67. Smirnova, E.; Vasile, F.: Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. (2017). https://doi.org/10.1145/3125486.3125488
68. Verma, M.; Ganguly, D.: LiRME: Locally interpretable ranking model explanation. SIGIR 2019 - Proc. 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval 1281–1284 (2019). https://doi.org/10.1145/nnnnnnn.nnnnnnn
69. Hongliang, C.; Xiaona, Q.: The video recommendation system based on DBN. In:Proceedings of - 15th IEEE The International Conference on Information and Computer Technologies CIT 2015, 14th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2015, 13th IEEE Int. Conf. Dependable, Auton. Se. 1016–1021 (2015). https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.154
70. Covington, P.; Adams, J.; Sargin, E.: Deep Neural Networks for YouTube Recommendations. ACM Conference Recommendations Systems. 191–198 (2016). https://doi.org/10.1145/2959100.2959190
71. Lee, H.; Ahn, Y.; Lee, H.; Ha, S.; Lee, S.: Quote Recommendation in Dialogue using Deep Neural Network. In:Proceedings of 39th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'16. 957–960 (2016). https://doi.org/10.1145/2911451.2914734
72. Tan, J.; Wan, X.; Xiao, J.: A Neural Network Approach to Quote Recommendation in Writings. In:Proceedings of 25th ACM International Conference on Information and Knowledge Management - CIKM'16. 65–74 (2016). https://doi.org/10.1145/2983323.2983788
73. Zhang, H.; Yang, H.; Huang, T.; Zhan, G.: DBNCF: Personalized courses recommendation system based on DBN in MOOC environment. In:Proceedings of - 2017 Int. Symp. Educ. Technol. ISET 2017. 106–108 (2017). https://doi.org/10.1109/ISET.2017.33
74. Wang, X.; Zhang, Y.; Yu, S.; Liu, X.; Yuan, Y.; Wang, F.Y.: E-learning recommendation framework based on deep learning. 2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017. 2017-Janua, 455–460 (2017). https://doi.org/10.1109/SMC.2017.8122647
75. Li, P.; Wang, Z.; Ren, Z.; Bing, L.; Lam, W.: Neural Rating Regression with Abstractive Tips Generation for Recommendation. 345–354 (2017). https://doi.org/10.1145/3077136.3080822
76. Maheshwary, S.; Misra, H.: Matching Resumes to Jobs via Deep Siamese Network. Companion Proc. Web Conf. **2018**, 87–88 (2018). https://doi.org/10.1145/3184558.3186942
77. Jaradat, S.: Deep Cross-Domain Fashion Recommendation. In:Proceedings of Elev. ACM Conference Recommendations Systems. - RecSys'17. 407–410 (2017). https://doi.org/10.1145/3109859.3109861
78. Jiang, S.; Wu, Y.; Fu, Y.: 5 Deep Bidirectional Cross-Triplet Embedding for Online Clothing Shopping. ACM Trans. Multimed. Comput. Commun. Appl. Artic. 14, 1–22 (2018). https://doi.org/10.1145/3152114
79. Webb, T.; Harnden, D.G.: The transformation by simian virus 40 of cells from patients with mucopolysaccharidosis and from normal controls. Cancer Res. **36**, 203–212 (1976)
80. Zahalka, J.; Rudinac, S.; Worring, M.: Interactive multimodal learning for venue recommendation. IEEE Trans. Multimed. **17**, 2235–2244 (2015). https://doi.org/10.1109/TMM.2015.2480007
81. Gao, T.; Li, X.; Chai, Y.; Tang, Y.: Deep learning with consumer preferences for recommender system. 2016 IEEE International Conference on Information Autom. IEEE ICIA 2016. 1556–1561 (2017). https://doi.org/10.1109/ICInfA.2016.7832066
82. Dai, H.; Wang, Y.; Trivedi, R.; Song, L.: Recurrent coevolutionary latent feature processes for continuous-time recommendation. ACM Int. Conf. Proceeding Ser. 15-Septemb, 29–34 (2016). https://doi.org/10.1145/2988450.2988451
83. Dominguez, V.; Messina, P.; Parra, D.; Mery, D.; Trattner, C.; Soto, A.: Comparing Neural and Attractiveness-based Visual Features for Artwork Recommendation. In:Proceedings of 2nd Workshop on Deep Learning based Recommender System - DLRS 2017. 55–59 (2017). https://doi.org/10.1145/3125486.3125495
84. Soh, H.; Sanner, S.; White, M.; Jamieson, G.: Deep Sequential Recommendation for Personalized Adaptive User Interfaces. In:Proceedings of 22nd Int. Conf. Intell. User Interfaces - IUI'17. 589–593 (2017). https://doi.org/10.1145/3025171.3025207
85. Jishan, S.T.; Wang, Y.: Audience Activity Recommendation Using Stacked-LSTM Based Sequence Learning. In:Proceedings of 9th Int. Conf. Mach. Learn. Comput. - ICMLC 2017. 98–106 (2017). https://doi.org/10.1145/3055635.3056606
86. Wu, H.; Zhang, Z.; Yue, K.; Zhang, B.; He, J.; Sun, L.: Dual-regularized matrix factorization with deep neural networks for recommender systems. Knowledge-Based Syst. **145**, 1–14 (2018). https://doi.org/10.1016/j.knosys.2018.01.003
87. Yuan, W.; Li, C.; Guan, D.; Han, G.; Khattak, A.M.: Socialized healthcare service recommendation using deep learning. Neural Comput. Appl. **30**, 2071–2082 (2018). https://doi.org/10.1007/s00521-018-3394-4
88. Katzman, J.L.; Shaham, U.; Cloninger, A.; Bates, J.; Jiang, T.; Kluger, Y.: DeepSurv: Personalized Treatment Recommender System Using A Cox Proportional Hazards Deep Neural Network. 1–12 (2018). https://doi.org/10.1186/s12874-018-0482-1
89. Wang, H.; Yeung, D.: IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING Towards Bayesian Deep Learning: A Framework and Some Existing Methods. 1–14
90. Bai, T.; Wen, J.-R.; Zhang, J.; Zhao, W.X.: A Neural Collaborative Filtering Model with Interaction-based Neighborhood. In:Proceedings of 2017 ACM Conference on Information and

Knowledge Management - CIKM'17. 1979–1982 (2017). https://doi.org/10.1145/3132847.3133083
91. He, X.; Chua, T.-S.: Neural Factorization Machines for Sparse Predictive Analytics. 355–364 (2017). https://doi.org/10.1145/3077136.3080777
92. Lee, W.; Song, K.; Moon, I.-C.: Augmented Variational Autoencoders for Collaborative Filtering with Auxiliary Information. In:Proceedings of 2017 ACM Conference on Information and Knowledge Management - CIKM'17. 1139–1148 (2017). https://doi.org/10.1145/3132847.3132972
93. Zhang, W.; Liu, F.; Jiang, L.; Xu, D.: Recommendation based on collaborative filtering by convolution deep learning model based on label weight nearest neighbor. In:Proceedings of - 2017 10th Int. Symp. Comput. Intell. Des. Isc. 2017. 2, 504–507 (2018). https://doi.org/10.1109/ISCID.2017.235
94. Liu, J.; Wang, D.: PHD : A Probabilistic Model of Hybrid Deep Collaborative Filtering for Recommender Systems. Acml. 1–16 (2017)
95. Xue, H.J.; Dai, X.Y.; Zhang, J.; Huang, S.; Chen, J.: Deep matrix factorization models for recommender systems. IJCAI Int. Jt. Conf. Artif. Intell. 3203–3209 (2017). https://doi.org/10.24963/ijcai.2017/447
96. Tay, Y.; Luu, A.T.; Hui, S.C.: Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. 729–739 (2017). https://doi.org/10.1145/3178876.3186154
97. Dong, X.; Yu, L.; Wu, Z.; Sun, Y.; Yuan, L.; Zhang, F.: A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems. In:Proceedings of 31st AAAI Conf. Artif. Intell. 1309–1315 (2017). https://doi.org/10.1103/PhysRevLett.93.077207
98. Catherine, R.; Cohen, W.: TransNets: Learning to Transform for Recommendation. 288–296 (2017). https://doi.org/10.1145/3109859.3109878
99. Preethi, G.; Krishna, P.V.; Obaidat, M.S.; Saritha, V.; Yenduri, S.: Application of deep learning to sentiment analysis for recommender system on cloud. IEEE CITS 2017 - 2017 Int. Conf. Comput. Inf. Telecommun. Syst. 93–97 (2017). https://doi.org/10.1109/CITS.2017.8035341
100. Serrà, J.; Karatzoglou, A.: Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. 279–287 (2017). https://doi.org/10.1145/3109859.3109876
101. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z.: Collaborative Filtering and Deep Learning Based Hybrid Recommendation for Cold Start Problem. In:Proceedings of - 2016 IEEE 14th Int. Conf. Dependable, Auton. Secur. Comput. DASC 2016, 2016 IEEE 14th Int. Conf. Pervasive Intell. Comput. PICom 2016, 2016 IEEE 2nd Int. Conf. Big Data. 874–877 (2016). https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2016.149
102. Wang, H.; Wang, N.; Yeung, D.-Y.: Collaborative Deep Learning for Recommender Systems, pp. 1235–1244 (2014). https://doi.org/10.1145/2783258.2783273
103. Li, Q.; Zheng, X.; Wu, X.: Collaborative Autoencoder for Recommender Systems, pp. 305–314 (2017). https://doi.org/10.1145/3097983.3098077
104. Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; Anil, R.; Haque, Z.; Hong, L.; Jain, V.; Liu, X.; Shah, H.: Wide & Deep Learning for Recommender Systems. (2016). https://doi.org/10.1145/2988450.2988454
105. Chen, C.; Zhao, P.; Li, L.; Zhou, J.; Li, X.; Qiu, M.: Locally Connected Deep Learning Framework for Industrial-scale Recommender Systems. In:Proceedings of 26th International Conference on World Wide Web Companion - WWW'17 Companion, pp. 769–770 (2017). https://doi.org/10.1145/3041021.3054227
106. Da'u, A.; Salim, N.; Rabiu, I.; Osman, A.: Recommendation system exploiting aspect-based opinion mining with deep learning method. Inf. Sci. (Ny). 512, 1279–1292 (2020). https://doi.org/10.1016/j.ins.2019.10.038
107. Zhang, S.; Yao, L.; Sun, A.; Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. **52**, 1–35 (2019). https://doi.org/10.1145/3285029
108. Bobadilla, J.; Alonso, S.; Hernando, A.: Deep learning architecture for collaborative filtering recommender systems. Appl. Sci. (2020). https://doi.org/10.3390/app10072441
109. Zarzour, H.; Al-Sharif, Z.A.; Jararweh, Y.: RecDNNing: A recommender system using deep neural network with user and item embeddings. 2019 10th International Conference on Information Commun. Syst. ICICS 2019. 99–103 (2019). https://doi.org/10.1109/IACS.2019.8809156
110. Fessahaye, F.; Perez, L.; Zhan, T.; Zhang, R.; Fossier, C.; Markarian, R.; Chiu, C.; Zhan, J.; Gewali, L.; Oh, P.: T-RECSYS: a novel music recommendation system using deep learning. In: 2019 IEEE International Conference on Consumer Electronics. ICCE 2019. (2019). https://doi.org/10.1109/ICCE.2019.8662028
111. Lee, H.; Lee, J.: Scalable deep learning-based recommendation systems. ICT Express. **5**, 84–88 (2019). https://doi.org/10.1016/j.icte.2018.05.003
112. Nimirthi, P.; Venkata Krishna, P.; Obaidat, M.S.; Saritha, V.: A framework for sentiment analysis based recommender system for agriculture using deep learning approach. SpringerBriefs Appl. Sci. Technol. (2019). https://doi.org/10.1007/978-981-13-1456-8_5
113. Nweke, H.F.; Teh, Y.W.; Al-garadi, M.A.; Alo, U.R.: Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. Expert Syst. Appl. **105**, 233–261 (2018). https://doi.org/10.1016/j.eswa.2018.03.056
114. Zheng, L.: A survey and critique of deep learning on recommender systems. (2016)
115. Da'u, A.; Salim, N.; Rabiu, I.; Osman, A.: Weighted aspect-based opinion mining using deep learning for recommender system. Expert Syst. Appl. (2020). https://doi.org/10.1016/j.eswa.2019.112871
116. Wang, S.; Huang, C.; Li, J.; Yuan, Y.; Wang, F.Y.: Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts. IEEE Access. **7**, 136951–136961 (2019). https://doi.org/10.1109/ACCESS.2019.2942338
117. Huang, Z.; Tang, J.; Shan, G.; Ni, J.; Chen, Y.; Wang, C.: An efficient passenger-hunting recommendation framework with multitask deep learning. IEEE Internet Things J. **6**, 7713–7721 (2019). https://doi.org/10.1109/JIOT.2019.2901759
118. Vincent, P.; Larochelle, H.: Extracting and Composing Robust Features with Denoising.pdf. 1096–1103 (2008)
119. Zhang, X.; Zhong, J.; Liu, K.: Wasserstein autoencoders for collaborative filtering. Neural Comput. Appl. (2020). https://doi.org/10.1007/s00521-020-05117-w
120. Deng, X.; Huangfu, F.: Collaborative variational deep learning for healthcare recommendation. IEEE Access. **7**, 55679–55688 (2019). https://doi.org/10.1109/ACCESS.2019.2913468
121. Pan, Y.; He, F.; Yu, H.: Learning social representations with deep autoencoder for recommender system. World Wide Web **23**, 2259–2279 (2020). https://doi.org/10.1007/s11280-020-00793-z
122. Saravanan, B.; Mohanraj, V.; Senthilkumar, J.: A fuzzy entropy technique for dimensionality reduction in recommender systems using deep learning. Soft. Comput. **23**, 2575–2583 (2019). https://doi.org/10.1007/s00500-019-03807-9
123. Guan, Y.; Wei, Q.; Chen, G.: Deep learning based personalized recommendation with multi-view information integration. Decis. Support Syst. **118**, 58–69 (2019). https://doi.org/10.1016/j.dss.2019.01.003

124. Wang, K.; Xu, L.; Huang, L.; Wang, C.D.; Lai, J.H.: SDDRS: stacked discriminative denoising auto-encoder based recommender system. Cogn. Syst. Res. **55**, 164–174 (2019). https://doi.org/10.1016/j.cogsys.2019.01.011
125. Zhang, Y.; Yin, C.; Wu, Q.; He, Q.; Zhu, H.: Location-aware deep collaborative filtering for service recommendation. IEEE Trans. Syst. Man, Cybern. Syst. (2019). https://doi.org/10.1109/tsmc.2019.2931723
126. Ahamed, M.T.; Afroge, S.: A Recommender System Based on Deep Neural Network and Matrix Factorization for Collaborative Filtering. 2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019. 1–5 (2019). https://doi.org/10.1109/ECACE.2019.8679125
127. Nassar, N.; Jafar, A.; Rahhal, Y.: A novel deep multi-criteria collaborative filtering model for recommendation system. Knowl.-Based Syst. **187**, 104811 (2020). https://doi.org/10.1016/j.knosys.2019.06.019
128. Feinman, R.: A Deep Belief Network Approach to Learning Depth From Optical Flow, pp. 1–14
129. Pacheco, A.G.C.; Krohling, R.A.; da Silva, C.A.S.: Restricted Boltzmann machine to determine the input weights for extreme learning machines. Expert Syst. Appl. **96**, 77–85 (2018). https://doi.org/10.1016/j.eswa.2017.11.054
130. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015). https://doi.org/10.1016/j.neunet.2014.09.003
131. Hinton, G.E.; Osindero, S.; Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006). https://doi.org/10.1162/neco.2006.18.7.1527
132. Luo, L.; Zhang, S.; Wang, Y.; Peng, H.: An alternate method between generative objective and discriminative objective in training classification Restricted Boltzmann Machine. Knowl.-Based Syst. **144**, 144–152 (2018). https://doi.org/10.1016/j.knosys.2017.12.032
133. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017). https://doi.org/10.1016/j.neucom.2016.12.038
134. Hidasi, B.; Karatzoglou, A.: Recurrent Neural Networks with Top-k Gains for Session-Based Recommendations, pp. 370–371 (2017). https://doi.org/10.1145/3269206.3271761
135. Da'U, A.; Salim, N.: Sentiment-aware deep recommender system with neural attention networks. IEEE Access. **7**, 45472–45484 (2019). https://doi.org/10.1109/ACCESS.2019.2907729