Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777238

RECOMMENDATION SYSTEMS USING DEEP LEARNING METHODS: A REVIEW

Manjula A K
Research Scholar, Department of Computer Science and Engineerring, University Visvesvaraya College of Engineering, Bangalore University , Bengaluru-560001
Manjulaak18@gmail.com

Dr H N Champa
Professor, Department of Computer Science and Engineerring, University Visvesvaraya College of Engineering, Bangalore University , Bengaluru-560001
Champahn@uvce.ac.in

Abstract -- Recommendation Algorithms assist users so that they are able to identify intriguing material within a big corpus. Search engines and other recommendation systems have emerged as efficient methods for locating pertinent information in a reduced amount of time, thanks to the meteoric rise in the quantity of digital resources available on the Internet. Users get extremely helpful assistance from smart recommendation systems and sophisticated search engines these days, which is why it is necessary to develop these tools. The capacity of these systems to retrieve relevant information from vast volumes of data is largely responsible for both their widespread adoption and their practical applicability. As a result, massive and successful corporations have algorithms in place to efficiently learn the user's likes and dislikes and offer the user with the products that they would be interested in purchasing or viewing. Traditional methods such as collaborative filtering techniques and content-based techniques provide impressive results. However, deep learning, which is now being steadily used in domains such as NLP, image and audio processing amongst many other fields, may give answers that are both significant and effective to the issue of information overload. Although there is a vast and extensive body of research on Recommender Systems and potential algorithms to achieve optimal performance, there are very few studies that review generative, discriminative, and hybrid deep learning models and their applications in Recommendation Systems. The aim of this study is to review the various deep learning methods that exist in peer-reviewed literature, analyse their effectiveness, and also look at the growth in Industrial recommender systems over the years.
Index Terms— Recommendation Systems, Deep Learning, Content-Based Techniques, CF, Auto Encoders, Boltzmann Machines, CNN, RNN, Neural Networks, BaRT

I. INTRODUCTION

In 2006, Netflix put up a reward of one million dollars for anybody who could solve the challenge of anticipating a user's likes and dislikes by analysing said user's previous activity on the platform [1]. Various industrially-successful global business houses owe their popularity to the fact that they can anticipate the needs and the expectations of every individual customer. The importance of recommender systems, also known as RS for short, cannot be overstated

[2], particularly in industries such as online retailing, academic research, and the entertainment industry. A Recommender system is a means that assists users by providing them with products or services which are highly probable to be of interest to them [3]. The practise of using recommender systems in sales marketing is common among businesses. Tech giants like Amazon, Netflix, and so-forth, rely on Recommender Systems to serve their users the most relevant content [4-7].

In a nutshell, an RS is an automated system [8] that filters entities like products, people, advertisements, movies, or songs on the basis of user preference to predict items that the user might like the most. Netflix provides movie recommendations on the basis of a user’s watching history so as to find movies that the user would like that are comparable in theme, genre, aesthetic etc. Online commercial websites recommend products to their clients based on their previous product viewing or purchase history [9, 10].

Recommender Systems are used to take vague but useful information like user preference and interest, and represent them objectively. With the development in information technology and data science, there is a plethora of information and data available all over the internet, and while it does bring convenience, it also reduces efficiency in client-side applications, thus resulting in information overload. Recommendation Systems help to solve the information overload problem and narrow down the set of choices for the user [11]. Businesses can benefit from using Recommender Systems by selling more relevant items to the user and by better understanding what the user wants [12].

With the assistance of recommender systems, our day-to-day requirements, such as shopping, reading, listening to music, watching movies, and socialising, may all be met in a more satisfactory manner [13-15]. In order to achieve this goal, sophisticated Recommendation Systems are employed to deduce information of interest from significant volumes of data [16-18].

Recommender Systems function by either asking for feedback from the user regarding the relevance of the content, or by looking at other users who have similar behaviour, or by studying the user’s activity. User reviews, popularity, and relevance, all play a part in what the user gets recommended [18, 19]. The preferences of users may be graphically depicted (see Fig. 1) as links between individuals on one side and items or entities on the other, with the thickness of the lines between the people and products or things representing the strength of the connection.

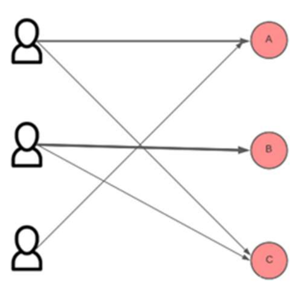


Fig 1. An example of a User Preference Graph

User preferences are messy and unpredictable, and hence it is difficult to perfectly predict what they might like. To estimate these values, the problem can be described as a matrix. Matrix Factorization Models [20] work by making a matrix of all the users against all the entities and mapping their preference and rating within the matrix. There will be unknown values that the model must predict. Fig 2 represents the User Preference Matrix of the graph in figure 1. Collaborative filtering techniques [21] and content-based methods [22] are the two primary ways that traditional recommender systems carry out their functions [23]. Content-Based Filtering uses the available information or features of the user or the entity to calculate the estimated strength of connection between them. Collaborative Filtering works with latent features, on the notion that a user will probably like things that people with similar activity also like. The features are extracted from patterns which exist across the users. Traditional RSs often face cold start and data sparsity problems [24]. Cold start means that they have no profile for completely new users.

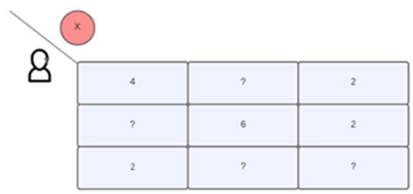


Fig 2. An example of a User Preference Matrix

There have been remarkable progress made in DL, and it has been found that it shows potential in the domains of voice control, image processing, NLP etc. Researchers have been working on ways to improve the efficiency of RS by using deep learning techniques [25], and they have made some progress. This study aims to explore existing deep learning methods and analyse their performance.

In Section 2, the study discusses the traditional methods that Recommender Systems employ to provide users with relevant entities of their interest. Section 3 gives an overview of the various categories of deep learning models that exist and can be applied in Recommender Systems. Section 4, 5 and 6 explore Generative, Discriminative and Hybrid Deep Learning approaches respectively, and the existing models available are discussed in detail. Section 7 gives an insight into the development of Recommender Systems for tech giants such as Amazon, Netflix, and other well-known companies. Section 8 concludes the paper and Section 9 contains the list of references.

II. TRADITIONAL RECOMMENDATION SYSTEMS

RSs are the software strategies that are typically considered to be the ones that make recommendations for products which are highly probable to be of the user's liking. The proposals include a wide range of decision-making procedures that are used in real-world services such as online shopping, social networking, and entertainment. RSs alleviate the issue of information overload as they filter information based on the target user's connection to the entity in question. There have been many different ways of classifying RS methods, and each of these ways is dependent on the information that was utilised and the kind of recommendation that was provided. Collaborative Filtering approaches, content-based methods, and hybrid recommendation algorithms are utilised the most often [26].

A. Collaborative Filtering Techniques

It has been shown that CF approaches are the most often employed procedures for RS. In most cases, they base their ability to predict a user's likes and dislikes on the history of the user's interaction with those products. As a result, in order to produce a recommendation, they take advantage of the collaborative capabilities offered by the ratings made by various users who are connected to one another. In practise, CF approaches arrive at recommendations by capitalising on either the link between users or the link between items [27]. There are methods that take advantage of both sorts of connections. In addition, some approaches generate optimized training models with methods that are analogous to the manner in which classifiers utilise the marked data to create training models. In their most basic forms, the collaborative filtering (CF) techniques may be broken down into two distinct categories, one based on users, and the other on items [28].

1) User-Based Collaborative Filtering

The first recommendation system ever developed was a user-based filtering method called collaborative filtering [29]. The algorithm will first determine a user set that is comparable to the user interest of the target user, and then it will propose an entity from the set of entities that are in the preference set of the users that the user is similar to [30]. The user is then recommended this entity if they have not yet interacted with it. The user-based CF has two steps.

Step 1: Finding a user collection that is comparable to the interest of the target user.

Step 2: Locate the entity in the collection that the users in the collection enjoy, and ensure that the target has never heard of said entity.

Before one computes the similarity among users, one must first transform the data of entities the user bought into index data which indicates that the product was bought by the aimed user; in further arguments, the overturned directory of the item using the data of the items bought by the aimed user needs to be created [31]. Once the overturned directory of the item is determined, the resemblance among users may be estimated using the following equation for similarity:

$$W_{a,b} = \frac{|N(a) \cap N(b)|}{\sqrt{|N(a)| * |N(b)|}} \tag{1}$$

This measure is known as Cosine Similarity [32]. N(a) designates the number of products that user a bought, while N(b) designates the number of products that user b bought. Together, these two numbers indicate the number of identical stuffs bought by users a and b. The K most comparable users are chosen for a user U, and the things that the user U has not yet interacted with in any shape or form are suggested to that user using the set of products that have already been bought by similar users.

2) Item-Based Collaborative Filtering

The item-based collaborative filtering algorithm, which suggests users the goods that are identical to the items they have previously enjoyed, is now the most extensively used algorithm in the industry [33]. While computing the measure of resemblance between two stuffs, the Item-CF method does not consider the individualities of the stuffs being compared. It does most of its work by determining the degree of similarity between things by analysing the user's behaviour record. Because people who like item A tend to also like item B, the algorithm feels that item A and B have a significant amount of similarity with one another. There is a theory that each user's interests are restricted to a small number of aspects; hence, if two things appear on the interest list of the same user, then it is possible that the two items fall in the same small number of categories [34]. And if two different products exist in the interest lists of many different people simultaneously, it's possible that they are from the same category and thus have a lot in common.

3) Pros of Collaborative Filtering

In comparison to the CB approaches, the CF techniques are methods that are employed for RS the most often. The ability of CF models to grind in a field in which the information correlated with the items is inadequately obtainable and in a state in which the information is hard to develop, such as subjective user opinions [35, 36], proves to be a benefit of CF models in comparison to CB approaches. Another benefit is that CF models are more flexible. The capacity of the collaborative method to generate intriguing and not very evident suggestions is another benefit of using this approach. In spite of the fact that the CF filtering techniques offer a multitude of benefits over the CB filtering methods, they do have some possible downsides, which will be discussed next.

4) Cons of Collaborative Filtering

a) The cold-start issue, which happens in most cases when insufficient information is available about an object or a user to make pertinent predictions about them [37]. The efficiency of the RS will normally suffer as a result of this. In practice, the record of a new person will be blank since there will be no ratings associated with them in the system. As a consequence of this, the system will not be able to detect the user's preferences.

i. User cold-start: When new users who don't have any preferences on any products come to the recommendation system, we need to figure out how to provide tailored suggestions for them. This presents a difficulty that we need to address. Because the system has not had access to the user's interaction data, it is unable to make any inferences on the user's interests based on his or her past actions. Because of this, the system is unable to provide the user with tailored recommendations [38].

ii. Item's cold start takes place when it is added to a recommender system and it is the first time any community member has encountered it. The challenge of how to propose new products without previous users having any kind of engagement with the system is brought up by the item cold-start [39].

iii. System cold-start primarily indicates how to create a personalised recommender system on a recently developed official site (no user, no user behaviour), consumers to experience a personalized recommender service [37].

b) Data sparsity is an issue that, in general, impacts CF approaches and occurs as a direct consequence of insufficient information incorporated within the system [40]. This occurs when there are not many things in the database that have been reviewed by the user overall. As a consequence, the resulting user-item matrix is often rather sparse. Additionally, it results in an inability to discover effective neighbours and ultimately a weak recommendation procedure. The recommender system has a sparsity issue, which means there are a huge number of null ratings in the user-item matrix [41]. This makes it very difficult to locate users who have similar preferences using the CF technique, and it also drives up the cost of storing space [42]. It is necessary for the user to have sufficient interactions with the objects in order for the profile to be useful and to contribute to the calculation of similarities. That is often not the case since many websites, including e-commerce companies, only have a tiny percentage of repeat customers [43]. For instance, the vast majority of users only make one visit or transaction.

c) Scalability is an additional concern that comes along with RS [44]. In general, the difficulty of the task of the recommender technique, which was originally created to deal with only a limited volume of a dataset, will increase as the volume of the dataset increases. As a result, it is of the utmost importance to make use of recommendation strategies that are capable of successfully scaling up in response to an increasing amount of datasets inside a database.

B. Content-Based Methods

When it comes to delivering item recommendations, CB filtering strategies almost exclusively depend on user and item descriptions [45, 46]. In most cases, web mining or information retrieval is used in order to accomplish the task of producing the relevant user/item data. In general, the CB approach will filter things based on how similarly they are related to the contents that the user is intrigued by. Vector Space Model and Latent semantic indexing are two ways that are often used to portray these terms as vectors in an n-dimensional space [47]. Learning a user profile often involves using a number of distinct learning methods, like Bayesian classifiers, SVM, and neural networks.

1) Pros of Content-Based Methods

Many of the problems that are seen while using CF techniques may be addressed by using CB filtering methods instead. The following provides an overview of some of the benefits that the CB approaches provide in comparison to the CF method.

In the CB approach, the ratings that are supplied by the user profile are often utilised purely for the purpose of building the user's own profile, in contrast to the CF method, which is based on the similar users' assessments of the entities. Thus, the target user is independent in CB methods. CB methods work best when content features are clearly stated. These features serve as a parameter for determining whether or not a suggestion may be trusted. This is in contrast to collaborative filtering mechanisms because the only parameter for recommending an item there is that anonymous users with similar likes and dislikes happened to like that item [48]. Unlike CF methods, CB methods are operational in producing a reference for items that have not been rated by any user in the past. As a result of this, the CB filtering techniques do not have an issue with first-raters. This issue is often encountered while using the CF approaches.

2) Cons of Content-Based Methods

Because they rely on content description, CB filtering strategies almost always result in suggestions that are very simplistic and obvious [26]. For instance, there is no possibility for an item to be recommended if the people who have voted on it have never used products that include specified terms. This is due to the fact that the intended approach is always specific to the target user. As a result, the collective experience of users with similar needs is rarely drawn upon. This has the undesirable effect of reducing the variety of products that may be suggested, which is not the best outcome. Another significant drawback of the CB approaches is perhaps that they are hardly useful for recommending products to new users, despite the fact that they are useful for recommending newly available items to existing users [49]. This is because rating history is necessary to be included in the training of the model for the user at hand. As a result, it is essential, in most cases, for an accurate result to have a prior experience with the target user.

C. Hybrid Recommendation Approach

The generation of improved recommendations may be facilitated by the use of hybrid RS techniques, which include the combination of different traditional RS methods (for instance, CF and CB) [25]. The fundamental concept that drives the development of hybrid recommendation strategies is the notion that merging many RS approaches will result in improved and enhanced recommendations in comparison to the use of a single strategy. The use of many approaches in conjunction with one another may mitigate the shortcomings of a single methodology. The hybridization of RS may be accomplished via the use of a variety of approaches. They can be cascading, feature augmentation, switching, and feature combination methods [50].

III. DEEP LEARNING – BASED APPROACHES

Deep learning is a branch of machine learning that may be used for demonstration learning by taking advantage of several layers of processing information in hierarchical structures [28]. The development of the deep learning model has led to amazing advancements in a variety of application areas, including computer visions, natural language processing, image processing, and remote sensing, to name a few. Generally speaking, deep learning architectures may be widely grouped into three basic groups, which are known as generative, discriminative, and hybrid architectures [51]. These classifications are determined by the manner in which the architectures are implemented.

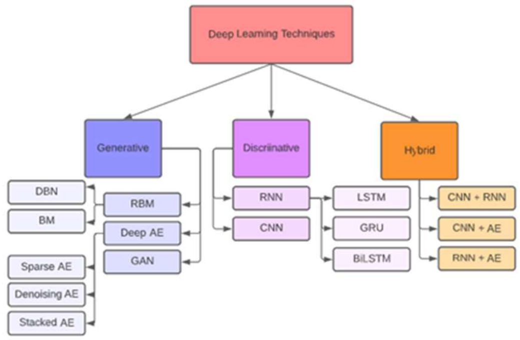


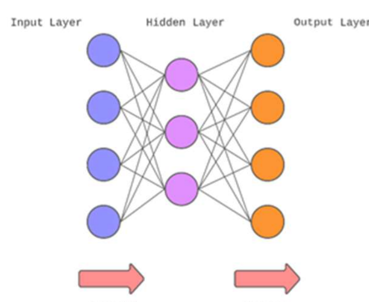
Fig 3. Overview of Deep Learning Techniques for Recommender Systems

IV. GENERATIVE DEEP LEARNING APPROACHES

For the purpose of pattern synthesis, generative architectures make an effort to describe the correlation features of the higher order for the data that is visible [52]. Primarily, unsupervised learning strategies are where they are utilised. During the actual process of learning, one often does not make use of some supervisory variables like the target class labels. The majority of the deep neural networks in this class generate samples, and as a result, they are considered to be generative models. Autoencoders, RBMs, and the relatively newer GAN, are all examples of models that fall under this category.

A. Deep Auto-Encoders

Deep AE is an unsupervised learning model. It is normally trained to copy inputs to outputs [52]. In this sense, an AE is a feed-forward model, which has a propensity to learn a data representation that is distributed. Reducing dimensions of the set is the only purpose for which it is used. Between the input and the output layers, the AE has one hidden layer.



.

Fig 4. An Auto Encoder

The coding feature y is obtained by encoding the sample data x of the Auto Encoder using the encoder function, which may be expressed by the given formula:

$$y = f_\theta(x) = s(W x + b) \quad (2)$$

where s is an excitation function for a neural network; in most cases, a function that is non-linear is used for this purpose; is a list of parameters. Then proceed with the equation that follows:

$$\hat{x} = g\theta(y) = s(W'y + b') \quad (3)$$

The objective of the optimization process for the AE is to transform the value of y into a recreation of the value of the initial input, which is denoted by the equation (3).

$$L = \|x - \hat{x}\|^2 \quad (4)$$

L, the average reconstruction error, is minimized and this is done by constantly changing the values of the parameters θ and θ', and it can be deemed that the output y hence obtained is adequately reflective of the original sample x [53].

1) Sparsed Auto- Encoders

Sparse autoencoder was primarily designed for the purpose of representing sparse features from input-data. This was accomplished by deliberately bringing the term-model-loss function as near to zero as possible. This paradigm is frequently used in situations that demand for the analysis of very complicated data, such as that which is included in multimedia. In essence, the use of sparse autoencoder models guarantees the reliability of the learning applications and feature representation. Thus, sparse-autoencoder approaches are quite useful for the low-dimensional extraction of features from given input data when the learning method is supervised [54].

2) Denoising Auto Encoders

When just the information from the initial input data is kept, the AE cannot ensure that it will learn a feature representation that is legitimate. For instance, the AE is able to readily learn any identity function in an extreme scenario; hence, it is important to provide the AE with a certain restriction in order for the input ratio to be learnt. The Auto Encoder (AE) serves as the basis for the Denoising Auto Encoder (DAE) [55]. Random noise is introduced into the input data in order to avoid the issue of over-fitting, and then the noise is introduced into the data in order to recreate the input data. In addition, denoising-AE may be layered in order to lessen the number of processing mistakes, and it can be used for developing complicated models such as RSs.

3) Stacked Denoising Auto Encoders

A cascade of numerous DAEs makes up the SDAE, which is a kind of network structure [28]. It can extract deeper aspects of the samples and analyse bigger data sets than other systems [56]. The SDAE model utilises the output of one layer as the feed for the subsequent layer, with the result of the final layer of DAE serving as the recreated data of the data that was first entered into the model. When training SDAE networks, a layer-wise unsupervised greedy training approach is used [57]. This algorithm is based on Deep Belief Network. When the training of the SDAE network is finished, the output of the nth layer is utilised as a valid feature of the initial input sample.

Collaborative Denoising AE is a new approach for top-N recommendation that was presented in reference [58]. This method makes use of the technique of Denoising Auto-Encoder. It is a flexible framework for top-N recommendation as well as an extension of latent component models. It generalises numerous current approaches that are quite popular. The framework that has been presented is naturally consistent with the denoising technique, which has the potential to significantly enhance the outcomes of suggestion.

Provided with a list of the top N recommendations, $C_{N,rec}$,

$$Precision@N = \frac{\left|C_{N,rec} \cap C_{adopted}\right|}{N} \quad (5)$$

$$Recall@N = \frac{\left|C_{N,rec} \cap C_{adopted}\right|}{C_{adopted}} \quad (6)$$

$C_{adopted}$ is the set of items that a user has accepted.

AP@N is the mean of all the computed precision scores with the adopted item and is mathematically represented as: $$AP@N = \frac{\sum_{k=1}^{N} Precision@k \times rel(k)}{min\{N, \ |C_{adopted}|\}} \quad (7)$$

Precision (*k*) here is the precision at cut-off *k* in $C_{N,rec}$, while $rel(k)$ is a function that is equal to 1 if the rank k item in adopted by the user, and 0 otherwise. The mean of these AP scores is the Mean Average Position, MAP@N.

When compared to the Matrix Factorization model, CDAE has higher scores for both MAP@10 and Recall@10 on the basis of the Yelp dataset. On the Netflix dataset, CDAE is around 10 percent more accurate than the second-best method, ITEMCF, as measured by the MAP@10 and Recall@10 criteria.

The information included in reference [59] demonstrates how stacking auto encoders makes it possible for the lower levels of the network to discover low-dimensional representations. It does so experimentally in order to improve the overall quality of the network. However, traditional autoencoders often degrade into identity networks, and therefore are unable to acquire an understanding of the connections between data points. As a result, one alternate approach to solving this problem is to corrupt the inputs, which will push the system to denoise the end results.

A deep learning-based framework for contextual recommendations is proposed in reference [60], which discusses the point that contextual information, which is frequently and widely available for recommendation tasks, can be of benefit if it is taken into account. A DAE neural network that has been reinforced with an attention mechanism that is context-driven is used for the proposed architecture. The model is referred to as an attentive contextual denoising auto-encoder (ACDA) [60-62]. To encode the contextual qualities in a hidden representation of the user's preference, the attention mechanism is used. This links tailored context with each user's choice in order to deliver suggestion that is targeted to that particular user. The performance of ACDA in movie and event recommendation tasks is then shown to be superior, much more so than that of CDAE, which was previously addressed.

A Bayesian generative model referred to as the Collaborative Variational Autoencoder (CVAE) is proposed in reference [63]. This model takes into account rating as well as content when it comes to recommendation and multimedia applications. The model is split into two parts. One part is about processing content information and is based on a self-encoder. The other part is about making Bayesian predictions and is based on a probability model. Using the user-item pair as an example, this variational self-encoder encodes the product's content to get the probability of the product's content. The probability matrix decomposition and the information about the item-rating are similar, and a normal distribution is used to hide each user and item. Finally, the user hidden representation as well as the item hidden representation are used to generate feedback.

Deep AEs are hence capable of learning complicated feature representation with the help of their unsupervised learning algorithm. They are however not as scalable to high dimensional data as they are to lower dimensional data.

B. Boltzmann Machines

The Deep Boltzmann Machine (DBM) is yet a different classic instance of a deep unsupervised model that exhibits a generative architecture [64]. The DBM is made up of multiple layers of concealed variables, and there are no direct links between the variables that are included inside the same layer. DBM is a specialised form of the original BM (Boltzmann-Machine). It is a structure that is linked symmetrically, based on stochastic mechanisms. In general, Boltzmann Machines have a very slow training speed and are also highly difficult to understand, despite the fact that their learning method seems to be relatively straightforward [65]. DBM has the benefit of learning complicated internal representations, which are highly significant in the process of finding solutions to issues involving object and voice recognition. An instance of a simple DBM model is shown in Figure 5.

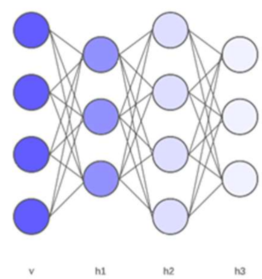


Fig 5. Deep Boltzmann Machine

Reducing the number of hidden layers in a DBM to one results in a Restrictive Boltzmann Machine [65]. RBM is a kind of undirected generative model that may be used as a component in greedy-layer-by-layer feature modelling. Typically, it is made up of two distinct layers: one that is visible, and another that is hidden, and both of these layers are built of input and hidden variables, respectively. Figure 6 demonstrates that the nodes are restricted to take the shape of a bipartite graph. There is a complete link between the units that are visible and the ones that are hidden, but there is no relationship between the units that are on the same layer. In order to get accurate estimations of maximum likelihoods learning, the models are trained using contrastive divergences, abbreviated as CD.

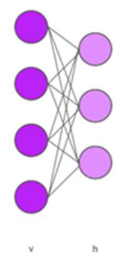


Fig 6. Restricted Boltzmann Machine

RBMs can be used in scenarios that require group-based recommendations.

Traditional neural networks often have issues with optimization, which in turn leads to the network's poor performance because of its inability to fully use its resources. In addition to this, they often waste what is available in plenty. DBN networks were established so that these problems might be solved [65]. Because it employs a deep architecture, DBN is unable to learn a feature representation using either labelled or unlabeled input. In most cases, DBN will combine supervised learning with unsupervised learning processes in order to construct an optimization that is both more resilient and more effective. The unsupervised stage may be used in the process of learning the distribution of the data without the need for any previous information, whereas the supervised step can be utilised to carry out a local search in order to get an optimum outcome.

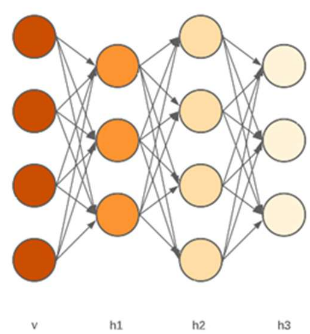


Fig 7. Deep Belief Network

A recommender system is described in Reference [66], which is based on the database maintained by the Ritsumeikan Art Research Center (ARC). This database holds a significant number of digitised copies of historic Japanese books, pamphlets, and artworks. This system does collaborative filtering by using an RBM, which helps it to predict which items will be the most interesting. By using the trained model, RBM is able to determine the new user's preferences. A number that denotes the number of times each picture has been seen is represented by each node. Reconstructing the inputs allows the RBM to learn effectively about the characteristics of those inputs. The system seems to provide satisfactory outcomes.

The RBM is also used for collaborative filtering in the reference [67], which suggests the neighbourhood-conditional Restricted Boltzmann Machine (N-CRBM) model. The activation functions, learning rates, and hidden units are the factors that are used for training and evaluating the model. The suggested N-CRBM model has 0.46 average RMSE and 78.5 percent accuracy in predicting consumers' choices of recommended ads, according to the results of simulations run on a dataset that contains 22 million records.

C. Generative Adversarial Network

Generative Adversarial Networks is a DL technique that was very recently developed [68]. It employs both an unsupervised learning approach and a supervised learning method, and it consists of two neural networks competing against one another in a zero-sum game. It makes a concerted effort to train a model, the purpose of which is to evaluate the distribution of the target data based on the training data. In addition to this, it makes use of a discriminative model, which estimates the likelihood that a given sample of data was derived from actual training data instead of the outcome. The learning rate as well as a variety of other factors, including the model structure are important for the training method of GAN [69]. When trying to achieve successful convergence, it is frequently necessary to use a large number of ad hoc strategies for the purpose of enhancing the integrity of the data that is created [70]. The GAN structure is shown pictorially in Figure 8. Several enhancements of the GAN approaches were created in order to reduce the difficulties and obtain greater training process convergence. The Loss Sensitive GAN (LSGAN) and the Wasserstein GAN (WGAN) [71] are two examples of these types of networks. Despite this, there hasn't been much research done on the GAN. GAN has recently been the subject of a number of research that suggested it may be used for supervised learning. It seems that making use of the unsupervised learning power of GAN might be beneficial for RS and IR systems.

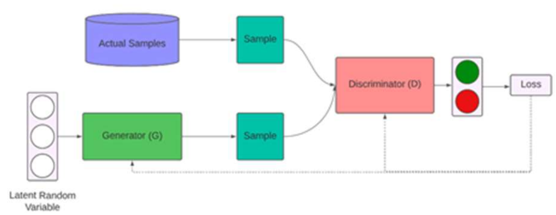


Fig 8. GAN Model Architecture

The authors of Reference [72] offer a novel optimization framework that, via the use of adversarial training, improves the pairwise ranking approach known as BPR (Bayesian Personalized Ranking). It is possible to interpret it as a minimax game, wherein the BPR must be minimised while an adversary changes model parameters so that the BPR function is maximized. They develop APR on MF in order to demonstrate how it works by imposing adversarial perturbations on the vectors of people and things. Extensive experiments conducted on three publicly available datasets have shown that APR is successful. Furthermore, when MF is optimised using APR, it surpasses BPR with an average improvement of 11.2 percent.

Because of their limited capacity to capture users' long-term stable interests and the fact that they only have short-term memories, most recommender models can only be applied to scenarios that are session-based. A few sequential or temporal recommender models have been developed relatively recently. The authors of [73] propose a streaming recommendation system model that is developed using neural memory networks that have external memory storage to collect and record both long-term consistent interests as well as relatively brief dynamic interests in a unified manner. This model is designed to capture and store both types of interests. For the purpose of optimising the streaming recommendation model, an adaptive negative sampling approach that is built using Generative Adversarial Networks has been developed.

This framework is able to effectively overcome the limitations of traditional approaches to negative sampling and enhancing the efficiency of the model parameter inference.

V. DISCRIMINATIVE LEARNING APPROACHES

This group of deep learning models provides a discriminative function for classifying patterns [74]. Most of the time, they try to describe the posterior class distribution based on the data that can be seen. For supervised deep learning, discriminative architectures are used vastly. CNN, MLP and RNN are all well-known examples of these kinds of architectures.

A. Convolutional Neural Networks

The Convolutional Neural Network (CNN) is a discriminative model that makes use of perceptrons to process high-dimensional input [54]. The structure of the visual cortex in many animals was the original motivation for the conception of CNN. Each module of a CNN consists of a pooling and a convolutional layer, and these layers are often layered one atop the other to build a deep architecture. Some invariance qualities, such as translation invariances, are provided to CNNs by the pooling layers, which, along with the convolutional layer's shared weight, makes up the convolutional neural networks (CNNs). The implementation of CNN models in a variety of applications, such as NLP and RSs, has included the use of a variety of pooling methods, including maximum, average, and stochastic pooling, among others [75, 76]. The overall outline of the CNN architecture is seen in figure 9. The capacity to perform pooling operations to reduce the dimensionality of training data, as well as the model's ability to be transitionally resilient against changes and distortions, are two of the primary benefits of the CNN model [77]. In order to enhance the representation of items and users in RS via improved modelling using CNN models, feature extraction was performed using those models. It has also been shown that the CNN model is particularly useful in image processing. As a result, a great number of techniques have taken use of the concept to construct RSs by using picture descriptions. In spite of the fact that the CNN model has achieved a lot of success, it does have a significant flaw, which is that it calls for a great deal of hyperparameter adjustment in order to provide the best possible features. In addition to this, effectively supporting complex activity details is difficult.

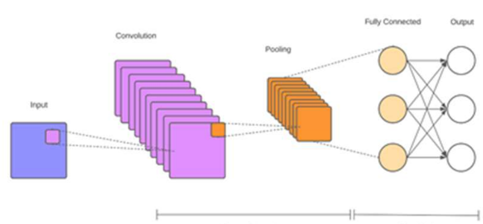


Fig 9. CNN Architecture

The majority of CNN-based RS techniques make extensive use of the CNN for feature extraction [78]. For example, [79] used two parallel CNN architectures to model the latent features of the user/item from the user's textual content. This strategy solves the cold start problem and makes the model easier to understand by using the user's review of the text to figure out what each word means. In particular, the model makes use of the word embedding approach in order to reduce the textual material into a low dimensional space while preserving the contextual information associated with each individual word. After the features have been retrieved, they are passed on to the convolutional layer, where they are convolved using a variety of kernels. Both the polling layer and the connected layer then process these features. The output of the two parallel networks is then eventually concatenated and utilised as input for the prediction layer. This is the layer where the user ratings on items are estimated using factorization machine.

Several of the studies that were investigated for this review made use of CNN methods for image processing. For instance, [80] constructed a visual content enhanced Point of Interest system by using a CNN model to extract picture characteristics. The framework for making recommendations was developed using the probabilistic MF method. This method took advantage of how the visual content and the latent user factors, as well as the visual content and the latent location, interacted.

In order to deal with the lack of data and boost the effectiveness of RS, [81] combined the CNN model with MF to make the convolutional MF model. The ConvMF model employs a CNN model to learn the feature representation. The Convolutional Neural Network is able to collect contextual information via the use of convolutional filters and word embeddings. This strategy was improved upon in the future by enabling the CNN model to more effectively understand the context of the users and items in order to achieve improved prediction performance [82].

[83] developed a visual BPR (VBPR)s model by integrating into MF visual features that were learnt using CNN. This model was further expanded upon in [84] by investigating the user's awareness as well as the visual characteristics that the user takes into consideration while making a selection.

B. Recurrent Neural Networks

RNN is used particularly for modelling sequential data by only inserting temporal layers in order to gather sequential information [54]. As a result, it typically becomes an appropriate technique to handle the temporal changes in user behaviour. This is because of the manner that it is structured. RNNs, as opposed to feed-forward networks, have memory and loops that allow them to recall previously processed data. In most cases, RNN takes advantage of the hidden units in order to learn complicated changes. The hidden units are able to adjust themselves to accurately reflect the present state of the network based on the information that is found in the network. RNNs perform the processing of the currently hidden state by activating the next hidden state. They, however, have a problem known as the vanishing or exploding gradient, which makes training them challenging. Because of this, its efficiency in modelling activities over extended periods of time and temporal dependence is hindered. When dealing with the problem of vanishing gradients, it is common practise to resort to variants of RNN, such as the GRU and the LSTM (Long Short Term Memory) system. In most cases, they combine a number of distinct memories and gates in order to record consecutive actions. In recent times, other iterations of the LSTM, such as the BiLSTM [85], have been developed to provide improved performance. However, a significant shortcoming of RNN models, particularly LSTM, is the lengthy amount of time needed for calculation because of the large number of parameters that need to be tuned. Several approaches, such as a technique with a high throughput for updating parameters, are available for use in combating the computation time.

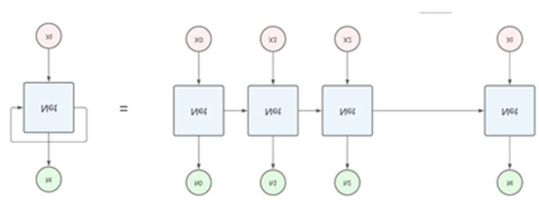


Fig 10. RNN Architecture

The vast majority of RNN-based RS methods, in particular, make use of the RNN model in order to provide session-based recommendations. RNNs have been used as a for feature representation and the construction of RS in several studies [86, 87].

[88] makes use of RNN architecture to enhance session-based recommendation by investigating and modifying a variety of methods. These methods involve data augmentation via the use of sequence pre-processing as well as embedding dropout for the purpose of improving training and lowering the problem of overfitting. To learn from limited datasets, there are many other methods such as model pretraining and distillation.

For the purpose of generating session-aware recommendations, [89] presented an RS that was constructed using the RNN model. This particular model makes use of hierarchical RNN, which means that the hidden state of a lower-level RNN at the conclusion of one user session is sent as an input to a higher-level RNN in order to make a prediction for the user's subsequent session.

Behaviour-Intensive Neural Network (BINN), which was developed by [90], is a model that uses LSTM and was given the name Behaviour-Intensive Neural Network. Its purpose is to improve the learning of item embedding and discriminative behaviour for sequential recommendation by looking at user preferences from the past along with the current consumption motivation. In particular, the model is made up of two primary components: one of these components is known as neural item embedding, and the other is known as discriminative behaviour learning. An item embedding approach that is dependent on user involvement has been created in order to achieve the goal of obtaining a unified item representation.

C. Multilayer Perceptron

The MLP is a type of feed-forward neural network which may have one or more than one layers and deals in non-linearities [51]. MLP can generally be considered to be the simplest architecture for deep learning. At least one of its layers is hidden, and all of those layers are linked to one another, like they are in feed forward networks. It is used in a large number of the available deep learning models. Linear method of RSs can be changed to non-linear methods using MLP. As a result, they are utilized in several applications, one of which is in Recommendation Algorithms.

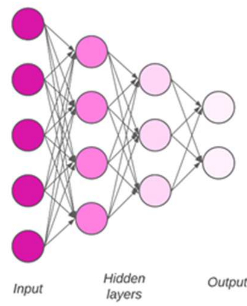


Fig 11. MLP with 2 hidden layers

In [91], one of the most fundamental strategies for using MLP for Recommender Systems was presented. In order to represent the information about the object and the user in terms of their connection, the model employs the utilisation of a dual network. The authors demonstrated that MF may be presented as an form of Neural Collaborative FIlter, and they made use of a multi-layer perceptron in order to enhance it with a non-linearities. In order to facilitate the complete neural coupling with CF, the authors made use of multilayer perceptrons for modelling the latent properties of both users and items. The user and object feature vectors are what make up the bottom input layer. These vectors can be changed to fit a broad range of user-item modelling situations, including context-aware and context-based modelling. Using user and item identities as input characteristics, the authors generated a binary sparse vector using one hot encoding. This was done because the model is entirely predicated on the collaborative settings. Using content characteristics as representations of users and objects efficiently helps to remedy the issue of a cold start. This technique was subsequently expanded to inter-domain recommendations by [92] and [93].

VI. HYBRID DEEP LEARNING APPROACHES

Several strategies have been presented that employ deep hybrid methods by integrating several deep learning strategies so as to further enhance the results of RSs. The following is a list of some of the hybrid techniques that are currently in use for recommender systems.

A. CNN + RNN

This combination captures the semantic information as well as the sequential information included in words. [94] merged CNN and RNN to create the quote RS. It uses CNN in learning semantics from the information extracted from tweets and to represent them as vectors. The target quotations relevant to a debate are selected using LSTM.

A deep hybrid model was suggested by the authors of [95] for the purpose of hashtag recommendation on the basis of a tweet along with accompanying photos. [96] addressed the sparsity issue utilising regularised matrix factorization and merged CNN and RNN to solve the problem.

B. CNN + AE

It is possible to create models that integrate CNN and AE in order to improve the learning of feature representations and to enable probabilistic treatment in order to enhance the RSs's overall performance. [97] merged AE and CNN to obtain textual and visual content utilizing word embedding technique for improved representation learning to enhance the prediction accuracy of RS.

C. RNN + DBN

It is possible to mine key comments using a combination of DBN and RNN algorithms. This combination is quite successful in solving the issue of the cold start. [1] suggested a parallel DNN for users and objects modelling for creating RS. This was done in order to create a hybrid recommendation system using a combination of DBN and RNN. Rating prediction is done with the assistance of a shared layer. The authors demonstrated that the model performs rating prediction much better than the conventional approach [98].

VII. INDUSTRIAL RECOMMENDER SYSTEMS

Nearly all global companies and tech giants employ Recommender Systems to assist their clients. This section discusses some of the most well-known companies, their robust Recommendation Systems, and a deeper insight into how they work.

A. Netflix

Netflix, since its 2006 contest incentivising development of Recommendation Systems, has expanded its use of data beyond the realm of rating prediction to include customised ranking, page creation, search, image selection, message, marketing, and other areas. The Netflix Recommendation Engine [99], which is the company's most effective algorithm, is made up of other algorithms that filter material based on the specific user profiles of each person. The engine processes approximately 3,000 titles concurrently while making use of 1,300 suggestion clusters that are determined by the preferences of the user. Netflix records data points such as the date and time when a customer watched a movie, the age, sex, region, chosen preferred content after signing up, the streaming device, pauses, fast forwards, rewinds, whether or not the viewer restarted viewing after pausing, whether or not a tv series or film was finished, the duration the user watched it for, feedback from users (Netflix has a thumbs up feature), repetitively viewed scenes, the user's search history, and scrolling and browsing behaviour on the site or application.

After some early difficulties with deep learning methods for Recommendation Systems, further tests revealed that deep learning algorithms, in particular, tended to shine when presented with extra information sources, context, and heterogeneous features.

B. Amazon

The very first recommendation engine that Amazon built using Collaborative Filtering was considered revolutionary twenty years ago. The item-based Collaborative Filtering technique, as discussed earlier, helped customers by showing them related items (for example, if one happens to buy a camera, the algorithm will recommend them a camera stand next, or a lens, or a camera bag). Amazon still uses this model, and also later combined collaborative filtering with heuristics to give customers personalized recommendations rather than just showing them the popular items.

Improvements in the computing power of Amazon Web Services and more sophisticated machine learning algorithms later led to development in these recommendation systems. This did not happen overnight. It started in 2012, when they first began with basic state-of-the-art graph clustering techniques, and then moved from matrix completion methods to deep learning methods because they could take advantage of non-linearities. However, these somehow performed worse than the best collaborative filtering algorithms. Amazon later started using the Sparse Auto Encoder, but it still did not improve on the performance shown by Collaborative Filtering. Various approaches like singular value decomposition, bilinear regression, and Restricted Boltzmann Machines were tried, but to very little improvement in results.

It was later found, in the context of Amazon Prime Video that matrix-completion methods learned that classic or Oscar-winning would be of interest to many customers. However, customers often tend to prefer movies that are newly-released or have mass appeal. A new algorithm for Recommendations was developed, using Multilayer Neural Networks for classification. The model is trained with a loss function to predict what the customers might want to watch in the next week. This simple approach outperformed other much more sophisticated applications and made it incredibly scalable. This deep-learning based technique perform two times better than Collaborative Filtering [100].

C. Spotify

The algorithm that determines which songs Spotify users should listen to next is known as BaRT, which stands for "Bandits for Recommendations as Treatments." Exploitation and exploration are the two modes that may be used with the BaRT model [101].

The term "exploitation" refers to the process through which the system leverages the information that it has obtained about the user, such as the songs that you skip or your favourites. Exploration, on the other hand, is when the system proposes songs to you based on all of the other data that the system is able to use, such as what other similar users like, what playlists exist containing songs that the target user liked, trending music, recent releases, etc. Exploration is when the system makes suggestions to you based on all of the data that the algorithm can use [102].

CF-based Recommendation Systems almost exclusively use the Exploitation mode [103]. This option is ideal in situations in which there is sufficient historical information about the individuals and the content that the users listened to. The exploitation mode takes advantage of all of the data that is accessible about the users and the songs, such as the songs that have been skipped, the number of times a song has been listened to, songs that have been shared, playlists, and so on. One of the most significant issues that arises with systems that are based only on Exploitation is the issue of item relevance. If the system has very little information about a person or an object (in this instance a particular song), for example, the song has not really been heard very frequently, then it is unsure whether to recommend the song or not (that is, whether to exploit or ignore).

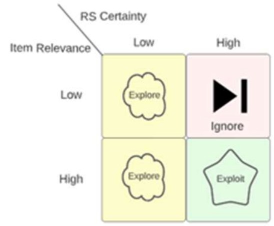


Fig 12. Bandit Method Modes

If a new track has not yet been listened to very often, the model will need statistics to determine whether or not the song has promise. Therefore, the BaRT feature of Spotify will propose these tracks to you through exploration while simultaneously gathering information on the new song. A song is considered as a negative if it is listened to for a duration of less than half a minute. Therefore, in other words, BaRT is founded on the concept of reinforcement learning and seeks feedback in an effort to enhance user satisfaction and rectify incorrectly anticipated suggestions. When it comes to persons and music that have previously participated in the system by either uploading or listening to a song, collaborative filtering works perfectly. Figure 12 illustrates how Bandit approaches are used by Spotify in order to explore, exploit, and explain its suggestions [104].

VIII. CONCLUSION

This study presented a detailed review on Recommendation Systems and the deep learning techniques that may be used to develop them. Traditional methods such as collaborative filtering and content-based procedures were touched upon. The problem of cold starting and the problem of data scarcity were dissected in great detail. Generative Deep Learning algorithms like Auto Encoders, Boltzmann Machines, and Generative Adversarial Networks were examined, followed by Discriminative approaches like RNN, CNN, and MLP. Some hybrid methods were also explored and it was seen how a few of them partially address the issues of the model they have been combined with. Finally, a review of the Recommender Systems of Netflix, Amazon, and Spotify was presented and their working was explained. In conclusion, although Recommender Systems have come a long way, there is still a significant amount of room for growth in RSs that are based on deep learning.

IX. REFERENCES

[1] J. P. White, "Lessons on Innovation and Collaboration from the Netflix Competition," 2009.

[2] N. Polatidis, C. K. J. I. J. o. E.-E. Georgiadis, and Innovation, "Recommender systems: The Importance of personalization in E-business environments," vol. 4, no. 4, pp. 32-46, 2013.

[3] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in Proceedings of the 1st ACM conference on Electronic commerce, 1999, pp. 158-166.

[4] R. Burke, A. Felfernig, and M. H. J. A. M. Göker, "Recommender systems: An overview," vol. 32, no. 3, pp. 13-18, 2011.

[5] S. M. Al-Ghuribi and S. A. M. J. I. A. Noah, "Multi-criteria review-based recommender system–the state of the art," vol. 7, pp. 169446-169468, 2019.

[6] D. Jannach, S. Naveed, and M. Jugovac, "User control in recommender systems: Overview and interaction challenges," in International Conference on Electronic Commerce and Web Technologies, 2016, pp. 21-33: Springer.

[7] A. Das, C. Mathieu, and D. J. a. p. a. Ricketts, "Maximizing profit using recommender systems," 2009.

[8] O. Moselhi, H. Bardareh, and Z. J. A. S. Zhu, "Automated data acquisition in construction with remote sensing technologies," vol. 10, no. 8, p. 2846, 2020.

[9] T. Ngo, J. Kunkel, and J. Ziegler, "Exploring mental models for transparent and controllable recommender systems: a qualitative study," in Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, 2020, pp. 183-191.

[10] J. Das, P. Mukherjee, S. Majumder, and P. Gupta, "Clustering-based recommender system using principles of voting theory," in 2014 International conference on contemporary computing and informatics (IC3I), 2014, pp. 230-235: IEEE.

[11] Y. Wang, Y. Liu, and X. Yu, "Collaborative filtering with aspect-based opinion mining: A tensor factorization approach," in 2012 IEEE 12th International Conference on Data Mining, 2012, pp. 1152-1157: IEEE.

[12] N. Ghanem, S. Leitner, and D. J. a. p. a. Jannach, "Balancing consumer and business value of recommender systems: A simulation-based analysis," 2022.

[13] L. M. Gladence, V. M. Anu, R. Rathna, E. J. J. o. A. I. Brumancia, and H. Computing, "Recommender system for home automation using IoT and artificial intelligence," pp. 1-9, 2020.

[14] U. A. Bhatti, M. Huang, D. Wu, Y. Zhang, A. Mehmood, and H. J. E. i. s. Han, "Recommendation system using feature extraction and pattern recognition in clinical care systems," vol. 13, no. 3, pp. 329-351, 2019.

[15] A. Gyrard and A. J. S. H. Sheth, "IAMHAPPY: Towards an IoT knowledge-based cross-domain well-being recommendation system for everyday happiness," vol. 15, p. 100083, 2020.

[16] K. Dahdouh, A. Dakkak, L. Oughdir, and A. J. J. o. B. D. Ibriz, "Large-scale e-learning recommender system based on Spark and Hadoop," vol. 6, no. 1, pp. 1-23, 2019.

[17] L. A. G. Camacho, S. N. J. I. P. Alves-Souza, and Management, "Social network data to alleviate cold-start in recommender system: A systematic review," vol. 54, no. 4, pp. 529-544, 2018.

[18] X. Su, G. Sperlì, V. Moscato, A. Picariello, C. Esposito, and C. J. I. T. o. I. I. Choi, "An edge intelligence empowered recommender system enabling cultural heritage applications," vol. 15, no. 7, pp. 4266-4275, 2019.

[19] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing popularity bias in recommender systems with personalized re-ranking," in The thirty-second international flairs conference, 2019.

[20] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 301-304.

[21] Z. Cui et al., "Personalized recommendation system based on collaborative filtering for IoT scenarios," vol. 13, no. 4, pp. 685-695, 2020.

[22] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, and S. J. I. J. o. E. T. i. L. Luo, "A review of content-based and context-based recommendation systems," vol. 16, no. 3, pp. 274-306, 2021.
[23] X. Su and T. M. J. A. i. a. i. Khoshgoftaar, "A survey of collaborative filtering techniques," vol. 2009, 2009.
[24] Y.-D. Seo, Y.-G. Kim, E. Lee, and D.-K. J. E. S. w. A. Baik, "Personalized recommender system based on friendship strength in social network services," vol. 69, pp. 135-148, 2017.
[25] M. Kunaver and T. J. K.-b. s. Požrl, "Diversity in recommender systems–A survey," vol. 123, pp. 154-162, 2017.
[26] J. Lu, D. Wu, M. Mao, W. Wang, and G. J. D. S. S. Zhang, "Recommender system application developments: a survey," vol. 74, pp. 12-32, 2015.
[27] X. Yang, Y. Guo, Y. Liu, and H. J. C. c. Steck, "A survey of collaborative filtering based social recommender systems," vol. 41, pp. 1-10, 2014.
[28] Z. Batmaz, A. Yurekli, A. Bilge, and C. J. A. I. R. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," vol. 52, no. 1, pp. 1-37, 2019.
[29] L. H. J. E. S. w. A. A. I. J. Son, "HU-FCF: a hybrid user-based fuzzy collaborative filtering method in recommender systems," vol. 41, no. 15, pp. 6861-6870, 2014.
[30] N. Ghasemi, S. J. E. C. R. Momtazi, and Applications, "Neural text similarity of user reviews for improving collaborative filtering recommender systems," vol. 45, p. 101019, 2021.
[31] T. Gao, L. Jiang, and X. Wang, "Recommendation System Based on Deep Learning," in International Conference on Broadband and Wireless Computing, Communication and Applications, 2019, pp. 535-543: Springer.
[32] M. Gupta, A. Thakkar, V. Gupta, and D. P. S. Rathore, "Movie recommender system using collaborative filtering," in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 415-420: IEEE.
[33] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A hybrid approach using collaborative filtering and content based filtering for recommender system," in Journal of Physics: Conference Series, 2018, vol. 1000, no. 1, p. 012101: IOP Publishing.
[34] J. Herlocker, J. A. Konstan, and J. J. I. r. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," vol. 5, no. 4, pp. 287-310, 2002.
[35] F. O. Isinkaye, Y. O. Folajimi, and B. A. J. E. i. j. Ojokoh, "Recommendation systems: Principles, methods and evaluation," vol. 16, no. 3, pp. 261-273, 2015.
[36] R. Frey, D. Wörner, and A. Ilic, "Collaborative filtering on the blockchain: a secure recommender system for e-commerce," 2016.
[37] B. Lika, K. Kolomvatsos, and S. J. E. s. w. a. Hadjiefthymiades, "Facing the cold start problem in recommender systems," vol. 41, no. 4, pp. 2065-2073, 2014.
[38] J. Bobadilla, F. Ortega, A. Hernando, and J. J. K.-b. s. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," vol. 26, pp. 225-238, 2012.
[39] M. Hasan, F. J. B. D. Roy, and C. Computing, "An item–item collaborative filtering recommender system using trust and genre to address the cold-start problem," vol. 3, no. 3, p. 39, 2019.

[40] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. J. E. S. w. A. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," vol. 149, p. 113248, 2020.
[41] J. F. G. da Silva, N. N. de Moura Junior, and L. P. Caloba, "Effects of data sparsity on recommender systems based on collaborative filtering," in 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-8: IEEE.
[42] Z. Zhang and S. Qian, "The research of e-commerce recommendation system based on collaborative filtering technology," in Advances in Computer Science and Information Engineering: Springer, 2012, pp. 507-512.
[43] H. Jinming, "Application and research of collaborative filtering in e-commerce recommendation system," in 2010 3rd International Conference on Computer Science and Information Technology, 2010, vol. 4, pp. 686-689: IEEE.
[44] D. Kotkov, S. Wang, and J. J. K.-B. S. Veijalainen, "A survey of serendipity in recommender systems," vol. 111, pp. 180-192, 2016.
[45] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 627-636.
[46] C. C. Aggarwal, "Content-based recommender systems," in Recommender systems: Springer, 2016, pp. 139-166.
[47] B. Krishnamurthy, N. Puri, and R. J. P. C. S. Goel, "Learning vector-space representations of items for recommendations using word embedding models," vol. 80, pp. 2205-2210, 2016.
[48] S. Jain, A. Grover, P. S. Thakur, and S. K. Choudhary, "Trends, problems and solutions of recommender system," in International Conference on Computing, Communication & Automation, 2015, pp. 955-958: IEEE.
[49] G. Adomavicius, A. J. I. t. o. k. Tuzhilin, and d. engineering, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," vol. 17, no. 6, pp. 734-749, 2005.
[50] S. Bostandjiev, J. O'Donovan, and T. Höllerer, "TasteWeights: a visual interactive hybrid recommender system," in Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 35-42.
[51] L. Deng, D. J. F. Yu, and t. i. s. processing, "Deep learning: methods and applications," vol. 7, no. 3–4, pp. 197-387, 2014.
[52] L. J. A. t. o. S. Deng and I. Processing, "A tutorial survey of architectures, algorithms, and applications for deep learning," vol. 3, 2014.
[53] Y. J. F. Bengio and t. i. M. Learning, "Learning deep architectures for AI," vol. 2, no. 1, pp. 1-127, 2009.
[54] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. J. N. Alsaadi, "A survey of deep neural network architectures and their applications," vol. 234, pp. 11-26, 2017.
[55] H. Wang, X. Shi, and D.-Y. Yeung, "Relational stacked denoising autoencoder for tag recommendation," in Twenty-ninth AAAI conference on artificial intelligence, 2015.
[56] A. Van den Oord, S. Dieleman, and B. J. A. i. n. i. p. s. Schrauwen, "Deep content-based music recommendation," vol. 26, 2013.

[57] A. A. Asroni, "Handling Sparse Rating Matrix for E-commerce Recommender System Using Hybrid Deep Learning Based on LSTM, SDAE and Latent Factor."

[58] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in Proceedings of the ninth ACM international conference on web search and data mining, 2016, pp. 153-162.

[59] F. Strub and J. Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs," in NIPS workshop on machine learning for eCommerce, 2015.

[60] Y. Jhamb, T. Ebesu, and Y. Fang, "Attentive contextual denoising autoencoder for recommendation," in Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, 2018, pp. 27-34.

[61] T. A. Ebesu, "Deep learning for recommender systems," Santa Clara University, 2019.

[62] G. Yin, F. Chen, Y. Dong, and G. J. A. I. Li, "Attentive convolutional neural network with the representation of document and sentence for rating prediction," pp. 1-18, 2022.

[63] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 305-314.

[64] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. J. E. S. w. A. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," vol. 105, pp. 233-261, 2018.

[65] A. G. Pacheco, R. A. Krohling, and C. A. J. E. S. w. A. da Silva, "Restricted Boltzmann machine to determine the input weights for extreme learning machines," vol. 96, pp. 77-85, 2018.

[66] J. Wang and K. Kawagoe, "A recommender system for ancient books, pamphlets and paintings in ritsumeikan art research center database," in Proceedings of the 2018 10th International Conference on Computer and Automation Engineering, 2018, pp. 53-57.

[67] H. B. Yedder, U. Zakia, A. Ahmed, and L. Trajković, "Modeling prediction in recommender systems using restricted boltzmann machine," in 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2017, pp. 2063-2068: IEEE.

[68] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. J. a. p. a. Frey, "Adversarial autoencoders," 2015.

[69] H. Chen, S. Wang, N. Jiang, Z. Li, N. Yan, and L. J. I. J. o. I. S. Shi, "Trust-aware generative adversarial network with recurrent neural network for recommender systems," vol. 36, no. 2, pp. 778-795, 2021.

[70] D. Yang, Z. Guo, Z. Wang, J. Jiang, Y. Xiao, and W. Wang, "A knowledge-enhanced deep recommendation framework incorporating gan-based models," in 2018 IEEE International Conference on Data Mining (ICDM), 2018, pp. 1368-1373: IEEE.

[71] G.-J. J. I. J. o. C. V. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," vol. 128, no. 5, pp. 1118-1140, 2020.

[72] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 355-364.

[73] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, "Neural memory streaming recommender networks with adversarial training," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2467-2475.

[74] H. Wang and B. J. a. p. a. Raj, "A survey: Time travel in deep learning space: An introduction to deep learning models and how deep learning models evolved from the initial ideas," 2015.

[75] L. Antony Rosewelt, J. J. J. o. I. Arokia Renjit, and F. Systems, "A content recommendation system for effective e-learning using embedded feature selection and fuzzy DT based CNN," vol. 39, no. 1, pp. 795-808, 2020.

[76] V. Suma, R. A. Shetty, R. F. Tated, S. Rohan, and T. S. Pujar, "CNN based leaf disease identification and remedy recommendation system," in 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 395-399: IEEE.

[77] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in Proceedings of the eleventh ACM conference on recommender systems, 2017, pp. 288-296.

[78] J. Wang, H. Xie, O. T. S. Au, D. Zou, and F. L. Wang, "Attention-based CNN for personalized course recommendations for MOOC learners," in 2020 International symposium on educational technology (ISET), 2020, pp. 180-184: IEEE.

[79] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in Proceedings of the tenth ACM international conference on web search and data mining, 2017, pp. 425-434.

[80] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in Proceedings of the 26th international conference on world wide web, 2017, pp. 391-400.

[81] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 233-240.

[82] D. Kim, C. Park, J. Oh, and H. J. I. S. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," vol. 417, pp. 72-87, 2017.

[83] R. He and J. McAuley, "VBPR: visual bayesian personalized ranking from implicit feedback," in Proceedings of the AAAI conference on artificial intelligence, 2016, vol. 30, no. 1.

[84] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in proceedings of the 25th international conference on world wide web, 2016, pp. 507-517.

[85] J. Yoon and H. Kim, "Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis," in Proceedings of the 29th conference on computational linguistics and speech processing (ROCLING 2017), 2017, pp. 244-253.

[86] T. Singh, A. Nayyar, and A. Solanki, "Multilingual opinion mining movie recommendation system using RNN," in Proceedings of first international conference on computing, communications, and cyber-security (IC4S 2019), 2020, pp. 589-605: Springer.

[87] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," in Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 727-735.

[88] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in Proceedings of the 1st workshop on deep learning for recommender systems, 2016, pp. 17-22.
[89] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 130-137.
[90] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen, "Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1734-1743.
[91] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in Proceedings of the 26th international conference on world wide web, 2017, pp. 173-182.
[92] J. Lian, F. Zhang, X. Xie, and G. Sun, "CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems," in Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 817-818.
[93] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, 2017, pp. 185-194.
[94] H. Lee, Y. Ahn, H. Lee, S. Ha, and S.-g. Lee, "Quote recommendation in dialogue using deep neural network," in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 957-960.
[95] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 353-362.
[96] T. Tran, K. Lee, Y. Liao, and D. Lee, "Regularizing matrix factorization with user and item embeddings for recommendation," in Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 687-696.
[97] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Twenty-ninth AAAI conference on artificial intelligence, 2015.
[98] J. Y. Liu, "A survey of deep learning approaches for recommendation systems," in Journal of Physics: Conference Series, 2018, vol. 1087, no. 6, p. 062022: IOP Publishing.
[99] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, and J. J. A. M. Basilico, "Deep learning for recommender systems: A Netflix case study," vol. 42, no. 3, pp. 7-18, 2021.
[100] M. Alfarhood and J. Cheng, "DeepHCF: a deep learning based hybrid collaborative filtering approach for recommendation systems," in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 89-96: IEEE.
[101] T. Kang, H. Lee, B. Choe, and K. Jung, "Entangled bidirectional encoder to autoregressive decoder for sequential recommendation," in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1657-1661.

[102] R. Logesh and V. Subramaniyaswamy, "Exploring hybrid recommender systems for personalized travel applications," in Cognitive informatics and soft computing: Springer, 2019, pp. 535-544.

[103] C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. J. A. O. Chua, "Advances and challenges in conversational recommender systems: A survey," vol. 2, pp. 100-126, 2021.

[104] J. McInerney et al., "Explore, exploit, and explain: personalizing explainable recommendations with bandits," in Proceedings of the 12th ACM conference on recommender systems, 2018, pp. 31-39.