Big Data and Deep Learning Techniques Applied in Intelligent Recommender Systems

Yifeng He[1, *]
[1]University of California at Davis, CA, USA
[*]yfhe@ucdavis.edu

Abstract

With the rapid growth of big data and information all-round the internet, deep learning has become a solution to improve the quality of Recommender Systems (RS). Adapting to various kinds of data formats, modern deep learning methods can extract nontrivial hidden features from data in different fields, including deep semantic features from images and texts or relationships between users and items. However, some recently proposed algorithms have not yet been exploited by RS-related applications. In this paper, we will discuss several trendy deep learning networks with some of their notable variants in RS with respect to different kinds of data. In addition, we will provide an overview of approaches to choosing deep learning algorithms for specific recommendation tasks. Secondly, we will provide some possible research directions related to deep learning applications in RS.

Keywords: Recommender Systems, Deep Learning, Collaborative Filtering, Hybrid Recommendation, Big Data

1 Introduction

Recommender Systems have become an important tool in modern life. In general, RS focus on the problem with *user* and *item*, like consumers and products (Amazon and other shopping platforms) or audience and media content (YouTube, iTunes, and other media platforms), and many other businesses [1, 2, 3]. With its wild application in various fields, RS not only gives suggestions but also provides methods of predicting future trends [4]. Algorithms used in RS usually fall into two categories: content-based recommendation (CB) and collaborative filtering (CF) [5]. Sometimes the combination of the two is also used to improve performance. CB usually recommends by predicting the possibility of a particular user liking a particular item used on that user's history record, profile, item's description, and other metadata. Differ from CB learning on user's taste, CF methods study the relationship of the user and item in a neighborhood of users and a group of items based on similarity.

Deep learning (DL) has been utilized in RS in a hybrid way dealing with big data with various formats. DL methods are known for their ability to find out hidden information in the data via multiple linear combinations and nonlinear activation. In RS, DL methods can learn from user data and item data separately and map them into one single representation space where norm-based comparison is possible. This paper will describe and analyze the different DL approaches in RS, their applications, and future works. Consider the rapid grow of internet and social media, DL is suitable for making recommendation in big data environment for its epoch-based training nature.

Most of the published literature reviews either lack the recently developed algorithms or only focus on one kind of deep learning network but lack an overview of deep learning in RS as a whole with analysis and comparison of different networks [5, 15, 16]. For example, Ong et al. summarized some deep learning methods with application on RS, but missing some new developed popular methods like Graph Neural Network and Transformer [5]. On the other hand, other surveys only focus on one of the recent popular methods instead of providing a comparison to the others. Wu et al. summarize Graph Neural networks (GNN) based RS methods with taxonomy on evaluating GNN-based recommendations [16]. Gao et al. focus on examples of successful application of generative adversarial networks (GAN) for recommendations, showing some benefits of adversarial networks in noise reduction and discriminators in classifying auxiliary information of user-item interaction data while providing some possible research fields on GAN-based RS [15].

With the success of Transformers and Bidirectional Embedding Representation for Transformer (BERT) on natural language processing (NLP), the idea of attention mechanism and pre-training and fine-tuning has also been shown effective in the convolutional network, graph network, and RS [18, 21, 22, 31, 32]. Sun et al. proposed BERT4Rec as a solution for the sequential recommendation which uses the bidirectional self-attention mechanism to model users' preferences based on their past behaviors [21]. Zhang et al. proposed Graph-BERT, which can be used as a pre-training model for a user-item bidirectional graph [22]. Both newly found BERT-based algorithms contributed to the development and advancement of RS directly or indirectly, and few preview surveys covered these methods together with the more traditional learning methods.

This review gives a detailed summarization of the applications of popular DL methods in the recommendation. With the repaid development in DL, we believe a conclusive review would be useful in terms of both taking these methods into a new recommendation scenario or adapting/developing new DL methods for RS in the future. This review can provide an overall understanding of the interaction of the latest research in DL and RS.

2 Background

Deep learning is a subset of machine learning which utilize multiple linear combinations together with non-linear activation functions to extract hidden information in the dataset. Each linear combination with its activation is called a neuron, and multiple connected neurons form an artificial neuron network (ANN) with a shape similar to the human brain. With the development and utilization are different subfields of computer science and technology, numbers of variance of ANN appeared and getting the state-of-art performance in the sub-direction. Viewing the ANN as a connected graph, a layer of the ANN is the subset of neurons with the same depth, and the dimension of the layer denotes the number of neurons in that layer. Multi-layer perception is one of the simple forms of ANN, which consists of at least three layers: an input layer with a dimension same as the number of features in the dataset, a hidden layer with a hyper-parameter dimension, and an output layer with a dimension same as the ground truth. The linear combination coefficients are referred to as weights, which are updated by error back-propagating [7] guided by the loss function which measures the difference between the current result and the ground truth (expected results). The value updated by back-propagating is based on Stochastic Gradient Descent (SGD) and its optimized variants [8].

In general, the idea of deep learning is the employment of the described "deep" ANN with multiple layers that can be trained by gradient descent and back-propagation. The form of deep learning is not limited to linear combinations of neurons, meaning that the weights can be utilized in other formats. With different weight formats, neural networks are commonly classified into the following forms.

2.1 Datasets

Datasets are necessary for training the DL model and providing an intuition of the effectiveness of the method. We will provide some noteworthy examples and reveal their relationship to different models and different RS contexts. Datasets that will be discussed in this section are MovieLens, Amazon, and Ciao, representing data used in different kinds of recommendation situations.

MovieLens dataset contains user-item wise ratings from 1-5. The dataset also contains side information, including the demographic information of the user, the descriptions of the item, and the timestamp when the rating was made. The dataset was provided in three different sizes of user-item pairs: 100K, 1M, and 20M. Therefore, the MovieLens dataset can be used for all-scale collaborative filtering recommendations and graph-based learning methods.

Amazon dataset contains information about the products from three categories: Books, Instant Video, Beauty, and Electronics. The data is about all kinds of user-item ratings (number rating, helpfulness rating, and comment), product description (brand, price, description, photo, and category), and historical purchase/view records. Thus, the Amazon dataset is suitable for CF methods, CNN-based recommendation embeddings for product images, and sequential recommendations.

Ciao dataset is published by Tang et al. [34] for social recommendations. The dataset contains user-item ratings from 1-5, user reviews of the items, and directed trust relationships between users.

There are more useful datasets in the field corresponding to various recommendation tasks. We choose the above three datasets to provide a fair comparison of the DL-based recommendation methods. These datasets (and some others we do not include in this review) are often considered the most common shared datasets for DL-based RS models. Real world application of RS is often trained with big data from business that are enormous in volume and can be expressed in various form depending on the business type. Some common forms are tabular, image, text, audio, and graph data.

2.2 Metrics

All machine learning algorithms use evaluation metrics to measure their performance on a certain dataset. In this section, we will introduce some commonly used evaluation matrices that can be used in DL-based recommendations so that future researchers can adopt the most suitable one into their model. In RS, we usually calculate the metrics based on the top K recommendations, denoted by K or $@K$ in the following equations. For other notation, we use $\mathcal{U}$ To denote the set of all users, u to denote a single user from the users set, and $R(u)$ to denote the recommendation that the model made for u, and $T(u)$ to denote the ground truth that the user actually likes/buys/views.

Accuracy, **Precision**, **Recall, and F1** all measure the performance of top-K recommendations with a different focus. Accuracy measures the proportion of correction prediction/recommendation discarding any bias or impact of the incorrect guess. Precision and Recall, on the other hand, consider that either the predicted positive or actual may have a tremendous on the effectiveness of the model. F1, as a function of Precision and Recall, would consider both biases at the same time. Due to the user-facing nature of RS, accuracy is not usually used for evaluating an RS model [19].

$$\text{Precision}@K = \frac{1}{|\mathcal{U}|}\sum_{\mathcal{U}} \frac{|R^K(u) \cap T(u)|}{K}$$

$$\text{Recall}@K = \frac{1}{|\mathcal{U}|}\sum_{\mathcal{U}} \frac{|R^K(u) \cap T(u)|}{|T(u)|}$$

$$\text{F1}@K = \frac{1}{|\mathcal{U}|}\sum_{\mathcal{U}} \frac{2 \times \text{Precision}@K(u) \times \text{Recall}@K(u)}{\text{Precision}@K(u) + \text{Recall}@K(u)}$$

where $@K(u)$ in F1 means the metric for one particular user, with the averaging calculation.

HR is a special metric to measure the quality of the RS model by counting the proportion of users who actually click on the recommendation.

$$\text{HR}@K = \frac{1}{\mathcal{U}}\sum_{\mathcal{U}} I\left(|R^K(u) \cap T(u)| > 0\right)$$

Discount Cumulative Gain (**nDCG**) also measures the gain of the corrected recommendations. Normalized DCG (nDCG) is a normalized quality of (DCG)

$$\begin{aligned}
\text{DCG}@K &= \sum_{i=0}^{K} \frac{rel_i}{\log_2(i+1)} \\
\text{IDCG}@K &= \sum_{i=0}^{rel_K} \frac{rel_i}{\log_2(i+1)} \\
\text{nDCG}@K &= \frac{\text{DCG}@K}{\text{IDCG}@K}
\end{aligned}$$

where rel is a numerical score of the importance/relevance of item i.

MAE, **MSE,** and **RMSE** are metrics used for regression models by measuring the errors between the predicted rating and the real ratings. Let $P(u,i)$ denote the predicted rating for user u and item i, and $T(u,i)$ is the true rating. For any item i ,

$$\begin{aligned}
\text{MAE} &= \frac{1}{|\mathcal{U}|}\sum_{\mathcal{U}} |P(u,i) - T(u,i)| \\
\text{MSE} &= \frac{1}{|\mathcal{U}|}\sum_{\mathcal{U}} \left(P(u,i) - T(u,i)\right)^2 \\
\text{RMSE} &= \sqrt{MSE}
\end{aligned}$$

3 Neural Network Methods for RS

3.1 Convolutional Neural Network

A convolutional neural network (CNN) is a combination of convolution and deep learning. Convolving filters onto matrix-like format data is a method to extract hidden features [5, 6]. Most applications of CNN are related to computer vision and fields studying image data. In RS, CNN often provides a feature-wise information extraction of images or videos as data preprocessing for CB methods. Some researchers also use the local-concentration feature of the convolution operator to solve the sparsity problem in the CB methods.

The sparsity of the rating matrix has been a problem for recommendation with CF for a long time. Some text-feature extraction based have been proposed to help the accuracy of recommendation with sparse data but are limited by the fixed-position nature of work-embedding algorithms used for document language processing. This limitation leads to the inefficient utilization of context information for RS. Kim et al. [23] proposed a context-aware model of convolutional matrix factorization (ConvMF) to capture the contextual information that can be future used for rating prediction. ConvMF uses CNN to generate a latent vector of the document from text embedding vectors of both user and item description. Let $\mathcal{U}, \mathcal{I}$ denote the set of users and the set of items, the rating of the user-item pair is predicted based on a probabilistic model

$$p(R|U,V,\sigma^2) = \prod_{i}^{|\mathcal{U}|}\prod_{j}^{|\mathcal{I}|} N\left(r_{ij}|u_i^T v_j, \sigma^2\right)^{I_{ij}} \quad (1)$$

where $R \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, U, V are the document latent vectors for user and item as the output of the CNN, $N(x|\mu,\sigma^2)$ is the probability density for normal distribution, and I_{ij} is the indicator function of whether user i rated item j.

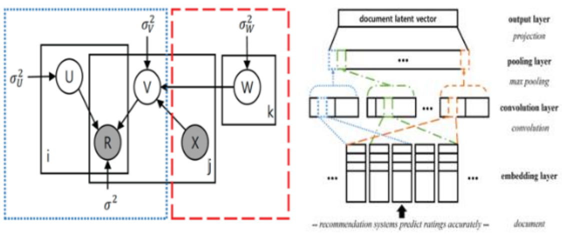

Figure 1. Convolutional Matrix Factorization [23]. The subfigure on the left is an illustration of how the CNN extract latent vectors from the text-embeddings. The same CNN is represented in the red dash box in the left subfigure, representing a preprocess of the text data. The blue dash box represents the PMF process.

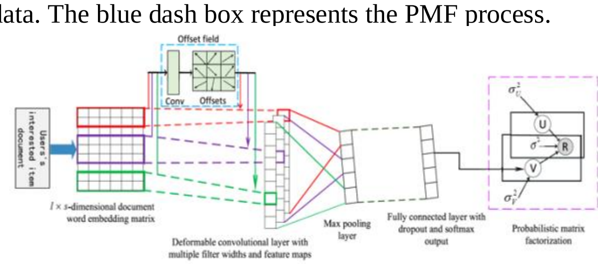

Figure 2. DCMF [24]. From left to right are the four layers of DCMF: Embedding Layer, Deformable Convolution Layer, Pooling Layer, and Full Connected Layer.

To deal with the same problem, Chen et al. [24] proposed Deformable Convolutional Matrix Factorization (DCMF). Deformable Convolutional Network (DCN) is a position-sensitive CNN variant with an offset layer [25], allowing better contextual information extraction. The outputs of the DCN, the document latent vectors, are then used for probability matrix factorization following equation 1.

Beyond using CNN as the processor for document latent vectors, Huang et al. further used two parallel CNN on user latent vectors and item latent vectors to extract hidden features for similarity comparison [26]. This module ensures the distance from users is closer to the items they like than the items they dislike, which provides a more accurate recommendation in their Hybrid recommendation model.

3.2 Generative Adversarial Network

Generative Adversarial Networks (GAN) contain two separate networks: generator and discriminator, where the generator turns a sample from random noise into fake data, and the discriminator decides if the data is from the real sample dataset or fake [14]. For RS application, GAN analyzes the robustness of recommendation models in addition to their accuracy [15, 27, 28, 29, 30]. Studies of GAN in the RS context mainly focus on the discriminator for noise reduction and informative classification from new data.

A basic GAN model consists of two neural networks: the generator G and the discriminator D . The generator maps a random noise to the same distribution of a given real dataset as close as possible, and the discriminator identifies if the input data is from the real sample or the generator. These two networks play a minimax game between each other and improve their ability through iterations. To train these two networks together via backpropagation, the loss function can be defined as

$$\min_G \max_D \mathcal{L}(D,G)$$
$$= \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z}\left[\log\left(1 - D\big(G(z)\big)\right)\right]$$

where p_{data} and p_z are distributions of the real sample and the random noise [14].

In RS, the most common applications of GAN are focusing on the ability of the discriminator and the generator separately. The Discriminator can be used to mitigate casual and malicious noise in the data. Adversarial Multimedia Recommendation (AMR) and Collaborative GAN (CGAM) are notable developments in this direction [27, 28]. Both models' discriminators therefore can improve the performance of recommendation. AMR design used CNN to extract features from the content as a real sample and add both item embedding and user embedding vector to the content embedding vector, whereas CGAN uses a Variational Auto-encoder (VAE) to replace the generator to capture latent information in item-user relations.

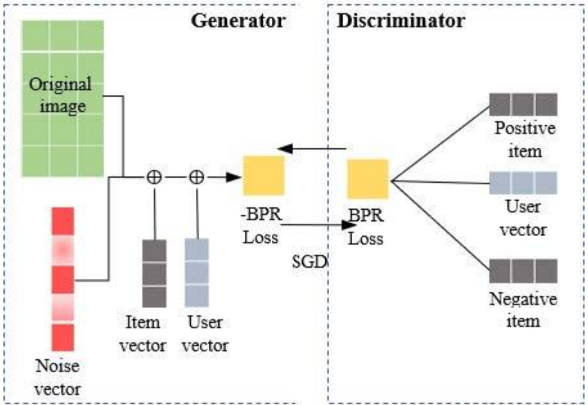

Figure 3. AMR [27]. One the left is how AMR learns hidden feature embeddings from image data and then adds user and item information to the embeddings.

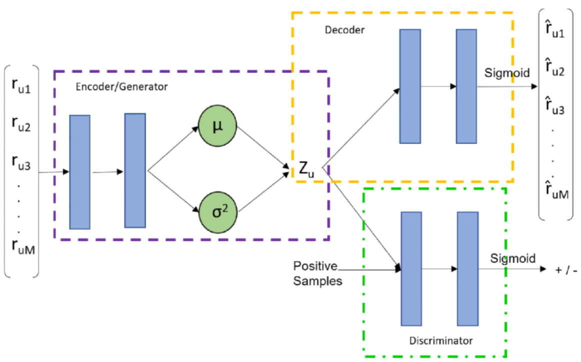

Figure 4. CGAN [28]. The model is composed of three components: Generator, Discriminator, and Decoder. The Decoder reconstructs the output of the Generator to form a vector of predictions along with the traditional GAN framework.

Another popular application of the discriminator is to distinguish informative samples from unseen items [15, 29]. Fan et al. proposed Deep Adversarial Social Recommendation (DASO) which guides the informative negative sampling process dynamically with the minimax game nature of GAN [29]. DASO uses bidirectional mapping between the social domain and item domain to allow the user embedding to contain both social and interactive information. Learning from both user-item interactions and user-user connections, DASO provides sufficient guidance for the optimizer to train the entire network.

Besides the works focusing more on the discriminator, researchers also found GAN is useful in generating and synthesizing user preferences. Recurrent GAN (RecGAN) by Bharadhwaj et al. is a combination of the NLP model to learn the latent features of users and items from description and GAN to make a recommendation [30]. RecGAN uses the generator to produce a list of items for a user, and the discriminator decides if the input list is from the user's real liked items or the generator. In this case, the minimax value function is summing over each user and each item and an addition summation for time T in from the NLP model

$$\min_G \max_D \mathcal{L}(D, G) = \sum_{t=1}^{T} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \mathbb{E}_{r \sim D(r|i,j)_{real|t}} \log D(r|u,i,t) + \mathbb{E}_{r \sim D(r|i,j)_{gen|t}} \log\big(1 - D(r|u,i,t)\big)$$

3.3 Self-Attention Network

A transformer network is also designed for modeling sequential data, but with better parallel computation ability than RNN, GRU, and LSTM [33]. Therefore, the network can be train faster with larger volume of data. Instead of passing information in the layers of a neural network, the Transformer adds positional encoding to the input data, denoting the sequential information and using Self-Attention to calculate a weighted sum of the encoded input itself [33]. The applications of Transformer network are similar to RNN. Based on Transformer, BERT gains its influence in NLP field. In RS, BERT4Rec has been proposed by Sun et al. with similar intuition that can be applied to end-to-end sequential recommendation [18]. Benefiting from the self-attention mechanism, BERT4Rec is able to capture the global interaction information for the users and the items [21]. The model outputs a list of items for given input sequence as shown in Figure 5.

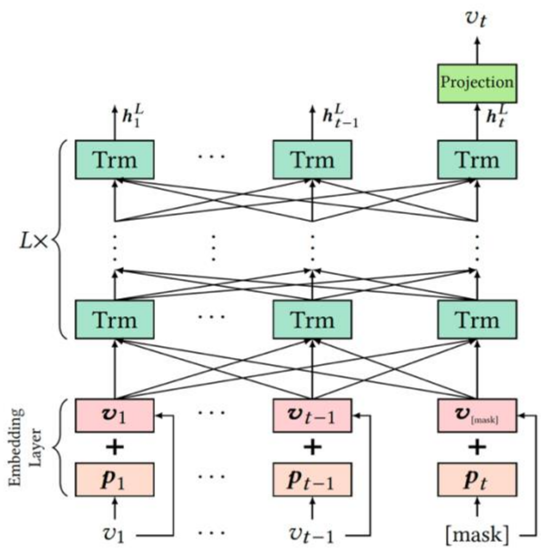

Figure 5. BERT4Rec [21]. The network begins with the Embedding Layer where p_i denotes the positional encoding of the item v_i. The sum of item embeddings and positional encodings will be the input of the following L Transformer Layers. Each Transformer layer (Trm) is a multi-head self-attention [33] sublayer, connected position-wisely to the other sublayers as a Feed-Forward Network.

3.4 Graph Neural Network

Algorithms based on GNN are learning on the graph data structure: directed/undirected graph, homogeneous/heterogeneous graph, or hypergraph [16]. Many different data can be represented in the graph, among all these data, the social network plays an important role in RS. One typical GNN framework is Graph Convolutional Network (GCN), which learns embedding for a target node through the embedding vectors of all the nodes in its adjacent neighbors with the Spectral Graph Convolutional (SGC) operator [17]. Most applications of GNN in RS fall into these categories: CF, sequential recommendation, knowledge graph-based recommendation, and social recommendation. For each task, the algorithm decomposes into two main steps: graph construction, and information propagation on the nodes. Information propagation can be further divided into neighbor aggregation, information update, and final node representation [16]. Each step of the RNN-based recommendation algorithm can be studied independently or jointly.

Task	Model
Content Embedding Understanding	CNN, GAN
Matrix Factorization Collaborative Filtering	CNN, GAN, GNN
Sequential Recommendation	CNN, GNN
Social Recommendation	CNN, GNN
Knowledge Graph based Recommendation	GNN

Table 1. Summarizing of models for specific recommendation task.

4 Conclusion and Future Work

Along with gaining attention and focus, deep learning models become more complex and diverse. Various models and combinations of models are being studied every day. To further improve the effectiveness of deep learning in RS, in addition to understanding the models, cross-domain DL-based RS, and better data quality discussed in previous works, we should also focus on integrating some of the recent works in DL into RS. Unsupervised pre-training on big datasets succussed in most tasks in NLP and CV [18, 31, 32] but is studied slightly in RS. BERT4Rec shows the possibility of a BERT-like structure working for recommendation. Zhang et al. proposed Graph-BERT as a pre-trainable network on the graph data and the fine-tuning experiment results in serval tasks are satisfactory [22]. Integrating Graph-BERT and the existing GNN-based RS algorithm is one of the possible research fields, decreasing the hardness of training task-specified networks. Combining the existing works of pretraining in CV with CNN-based RS algorithm is also legit following the same idea. In conclusion, the power of unsupervised pre-training on huge datasets is clearly shown in other ML-related fields while has not been revealed for recommendation tasks. By applying a similar idea to BERT and MAE in RS, the experiment of recommendation task will become easier for both academia and industry, and the performance will also become more stable.

5 Reference

[1] P. Covington, J. Adams, and E. Sargin, “Deep Neural Networks for YouTube Recommendations,” Sep. 2016. Accessed: May 18, 2022. [Online]. Available: http://dx.doi.org/10.1145/2959100.2959190

[2] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, Jan. 2003, doi: 10.1109/mic.2003.1167344.

[3] L. Barrington, R. Oda, and G. Lackriet RG, “Smarter than Genius? Human Evaluation of Music Recommender Systems.,” ISMIR, vol. 9, pp. 357–362, 2009.

[4] D. Thompson, “The Shazam Effect,” The Atlantic, Nov. 18, 2014. Accessed: May 18, 2022. [Online]. Available: https://www.theatlantic.com/magazine/archive/2014/12/the-shazam-effect/382237/

[5] K. Ong, S.-C. Haw, and K.-W. Ng, “Deep Learning Based-Recommendation System,” Nov. 2019. Accessed: May 18, 2022. [Online]. Available: http://dx.doi.org/10.1145/3372422.3372444

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[8] H. Robbins and S. Monro, “A Stochastic Approximation Method,” The Annals of Mathematical Statistics, vol. 22, no. 3, pp. 400–407, Sep. 1951, doi: 10.1214/aoms/1177729586.

[9] Y. LeCun, Fu Jie Huang, and L. Bottou, “Learning methods for generic object recognition with invariance to pose and lighting.” Accessed: May 18, 2022. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2004.1315150

[10] C. L. Giles, G. M. Kuhn, and R. J. Williams, “Dynamic recurrent neural networks: Theory and applications,” IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 153–156, Mar. 1994, doi: 10.1109/tnn.1994.8753425.

[11] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” arXiv.org, Dec. 11, 2014. https://arxiv.org/abs/1412.3555

[13] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” arXiv.org, Sep. 01, 2014. https://arxiv.org/abs/1409.0473

[14] I. J. Goodfellow et al., “Generative Adversarial Networks,” arXiv.org, Jun. 10, 2014. https://arxiv.org/abs/1406.2661

[15] M. Gao, J. Zhang, J. Yu, J. Li, J. Wen, and Q. Xiong, “Recommender Systems Based on Generative Adversarial Networks: A Problem-Driven Perspective,” arXiv.org, Mar. 05, 2020. https://arxiv.org/abs/2003.02474

[16] Y. Chu, J. Yao, C. Zhou, and H. Yang, “Graph Neural Networks in Modern Recommender Systems,” in Graph Neural Networks: Foundations, Frontiers, and Applications, Singapore: Springer Nature Singapore, 2022, pp. 423–445. Accessed: May 18, 2022. [Online]. Available: http://dx.doi.org/10.1007/978-981-16-6054-2_19

[17] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” arXiv.org, Sep. 09, 2016. https://arxiv.org/abs/1609.02907

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv.org, Oct. 11, 2018. https://arxiv.org/abs/1810.04805

[19] S. M. McNee, J. Riedl, and J. A. Konstan, “Being accurate is not enough,” Apr. 2006. Accessed: May 20, 2022. [Online]. Available: http://dx.doi.org/10.1145/1125451.1125659

[20] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” arXiv.org, Sep. 09, 2016. https://arxiv.org/abs/1609.02907

[21] F. Sun et al., “BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer,” arXiv.org, Apr. 14, 2019. https://arxiv.org/abs/1904.06690 (accessed May 18, 2022).

[22] J. Zhang, H. Zhang, C. Xia, and L. Sun, “Graph-Bert: Only Attention is Needed for Learning Graph Representations,” arXiv.org, Jan. 15, 2020. https://arxiv.org/abs/2001.05140

[23] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional Matrix Factorization for Document Context-Aware Recommendation,”

Sep. 2016. Available: http://dx.doi.org/10.1145/2959100.2959165

[24] H. Chen et al., “Deformable Convolutional Matrix Factorization for Document Context-Aware Recommendation in Social Networks,” IEEE Access, vol. 7, pp. 66347–66357, 2019, doi: 10.1109/access.2019.2917257.

[25] J. Dai et al., “Deformable Convolutional Networks,” arXiv.org, Mar. 17, 2017. https://arxiv.org/abs/1703.06211

[26] Z. Huang, C. Yu, J. Ni, H. Liu, C. Zeng, and Y. Tang, “An Efficient Hybrid Recommendation Model With Deep Neural Networks,” IEEE Access, vol. 7, pp. 137900–137912, 2019, doi: 10.1109/access.2019.2929789.

[27] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua, “Adversarial Training Towards Robust Multimedia Recommender System,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 5, pp. 855–867, May 2020, doi: 10.1109/tkde.2019.2893638.

[28] Y. Tong, Y. Luo, Z. Zhang, S. Sadiq, and P. Cui, “Collaborative Generative Adversarial Network for Recommendation Systems,” Apr. 2019. Accessed: May 24, 2022. [Online]. Available: http://dx.doi.org/10.1109/icdew.2019.00-16

[29] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, “Deep Adversarial Social Recommendation,” Aug. 2019. Accessed: May 24, 2022. [Online]. Available: http://dx.doi.org/10.24963/ijcai.2019/187

[30] H. Bharadhwaj, H. Park, and B. Y. Lim, “RecGAN,” Sep. 2018. Accessed: May 25, 2022. [Online]. Available: http://dx.doi.org/10.1145/3240323.3240383

[31] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” arXiv.org, Nov. 11, 2021. https://arxiv.org/abs/2111.06377

[32] A. Safaya, M. Abdullatif, and D. Yuret, “KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media,” 2020. Accessed: May 25, 2022. [Online]. Available: http://dx.doi.org/10.18653/v1/2020.semeval-1.271

[33] A. Vaswani et al., “Attention Is All You Need,” arXiv.org, Jun. 12, 2017. https://arxiv.org/abs/1706.03762

[34] J. Tang, H. Liu, H. Gao, and A. Das Sarmas, “eTrust,” 2012. Accessed: May 28, 2022. [Online]. Available: http://dx.doi.org/10.1145/2339530.2339574