Available online at: http://www.ijmtst.com/vol6issue04.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 06, Issue No: 04, April 2020

A Study on Autoencoder based Technique in Modern Movie Recommendation System

Vaibhavi Bhatt[1] | Bhavesh Tanawala[1]| Kirtikumar J. Sharma[1]

[1]Computer Engineering, Birla VishvakarmaMahavidyalaya, Vallabh Vidyanagar, Anand, Gujarat,India

To Cite this Article

Vaibhavi Bhatt, Bhavesh Tanawala and Kirtikumar J. Sharma, "A Study on Autoencoder based Technique in Modern Movie Recommendation System", International Journal for Modern Trends in Science and Technology, Vol. 06, Issue 04, April 2020, pp.:210-216.

Article Info

Received on 07-March-2020, Revised on 03-April-2020, Accepted on 15-April-2020, Published on 18-April-2020.

ABSTRACT

In today's era, people are suffering from some common problems like stress, depression, anxiety, discouragement and pessimism due to busy work life. Thus, a sprinkle of entertainment in life is a basic necessity to boost lives with some fresh energy. From long ago movies are considered as a great source of entertainment. The immense growth of the internet, mobile devices, and e-business lead to the tremendous growth of data. As a result, the searching process becomes very tedious and time consuming, which leads to the development of a system that can filter out information based on the user's requirement and priority. Recommendation systems are an effective information filtering tool, which constantly observes the behavior of users and provides recommendations according to their interests and preferences. This paper presents a preliminary survey on various movie recommendation approaches like content-based filtering, collaborative filtering, and hybrid approach. It also discusses various drawbacks of these approaches and mainly focuses on an autoencoder based deep learning approach to enhance the performance of recommendation systems.

KEYWORDS: *Movie, Recommendation System, Content-Based filtering, Collaborative filtering, Hybrid approach, Autoencoder, Deep learning*

I. INTRODUCTION

Entertainment brings happiness, which is a powerful medicine that adds good health and wellbeing in one's life. In this age of the internet, IT and mobile devices the craze of e-commerce and live streaming is increasing exponentially. The streaming services like Netflix, Amazon Prime, YouTube, Hulu and many more are considered as a vital source of Entertainment. As per the survey, 80% of people and especially the young generation prefer to watch television shows and movies via different applications instead of cable TV due to mainly two advantages: Firstly, it offers VOD (Video on demand) service and secondly, it provides a wide variety of great content. For example, in 2017 Netflix subscribers collectively watched more than 140 million hours per day and roughly 80% of hours were influenced by their proprietary recommendation system [1]. A recommendation engine is an intelligent system which main objective is to analyze the historical behavior of a user and based on their likes and dislikes it provides some suggestions according to the user's taste and preferences. Content based filtering, collaborative filtering, and hybrid approach are basic and widely used techniques of RS. These traditional techniques suffer from several challenges such as cold start, data sparsity,

scalability, and Overspecialization. Thus, these approaches need some improvement that can reduce the challenges and enhance the capabilities for better recommendation quality.

In recent years, deep learning becomes very popular in fields of image processing, data mining and especially machine learning. Due to its state-of-art- performance and high-quality recommendations, deep learning techniques have been gaining momentum in RS. Compare with traditional recommendation techniques, deep learning provides a better understanding of user's demand, items characteristic and historical interactions between them [1]. In this survey, I mainly focus on an autoencoder based deep learning approach to enhance the performance of traditional recommender systems.

II. LITERATURE REVIEW

A.Overview of Recommendation System

Given a set of users U and a set of items V, a recommender system is designed to recommend items to the users according to their purchase history or past ratings. Mostly, a recommendation system recommends items by either predicting ratings or providing a ranked list of items for each user.RS works based on three important stages, data selection, similarity decision and prediction computation [2]. Depending upon the type of input used to make a recommendation, recommendations systems mainly classified into the following categories: content-based recommender systems, collaborative filtering recommender systems, and hybrid recommender systems.

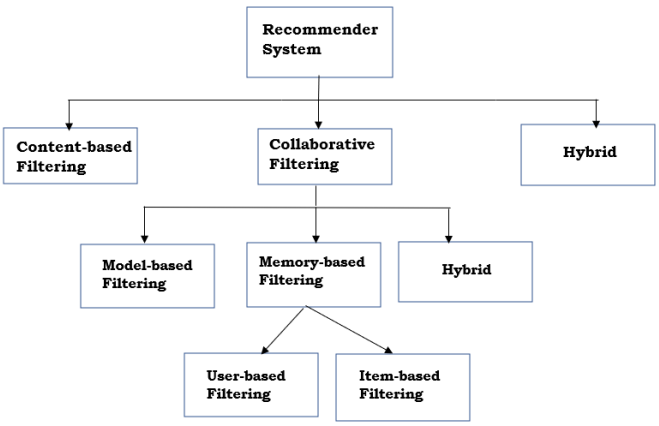

Figure 1: Classification of recommendation system

a.) Content Based Filtering

RS is based on item description. It identifies a new item for a particular user based on the user's interest in the items in the past. CB works in two steps: Firstly, it stores a user profile based on item features that are most commonly liked by the user. Now, these features are used to map the similarity of one item with another by using a different similarity measure. This model does not require any feedback from the user and sometimes a single preference is enough to recommend many items to the user [3-4]. For a movie RS user's profile is analyzed based on different features of items like genre, actor, director, singer, age and location of a user, etc. This technique makes use of K-means algorithm through Euclidean distance or cosine similarity to find how similar two movies from each other [5].

Limitations:

• Over-Specialization: The Overspecialization is that when the user is prohibited in getting recommendations that are similar to their profile.

• Limited Content Analysis: This is only a few keywords are not sufficient to provide accurate recommendation, it requires a rich description of items and well-organized user profile.

b.) Collaborative Filtering

It is the most popularly used recommendation technique. CF makes use of a dataset of user ratings given by user i for item(movie) j to predict ratings for a movie which is not yet seen by a user and based on predicted ratings it provides recommendations that a user might like. The model can be expressed as rating matrix of m × n, where m is number of users (U1, U2, U3,...., Un) and n is number of items (I1, I2, I3,...., In) this is movies in our case [3].

CF is divided into two categories, Memory-based and Model-based filtering.

i). Memory-Based Collaborative Filtering: This model requires the user to explicitly rate the movies and based on that, similarity is calculated between either users or items. To calculate similarity different similarity measures like Euclidean distance [6], Cosine similarity [6], City-Block distance [6], Pearson correlation [6] are used. This technique is mainly classified into two types:

• User-based CF: In this, the similarity between different users is determined to provide recommendations. The idea is to find top k similar users of user A and compute a weighted average of ratings for each group [7].

• Item-based CF: In this, the similarity between two items is determined. The idea is to find top k similar items of item A based on user ratings and if

user X like item A then similar items to A will be recommended to X [7].

ii). Model-Based Collaborative Filtering: In this, different machine learning and data mining methods like decision tree, Bayesian model, association rule, latent factor, clustering, artificial neural network, matrix factorization, SVD [3] are used to predict the user ratings for unknown movies.

iii). Hybrid: It is a combination of memory-based and model-based CF. It can overcome the limitation of CF like sparsity [3].

Table 1. Advantages and disadvantages of various collaborative filtering approaches

CF technique	Algoused	Advantages	Limitations
Memory based Collaborative filtering	• User basedCF • Item based CF (based on nearest neighbor eg. KNN)	• Simple to implement • Easy to add new items and update database	• Suffers from cold start for new user, problem of sparsity and scalability for large dataset
Model based Collaborative filtering	• Data mining and machine learning algorithms (mostly based on dimensionality reduction eg. SVD)	• Convert large high dimensional dataset into low dimension • Can handle the issue of sparsity and scalability especially in dealing with large sparse dataset	• Sometimes information may lose due to dimensionality reduction • Difficult to implement
Hybrid Collaborative filtering	• Combine both memory-based and model-based CF algorithm	• Overcome the limitations of traditional CF such as sparsity and cold start • Provides better prediction performance compare to traditional CF approaches	• Increase complexity • Expensive to implement

Limitations:

• Cold Start: Initially, for a new user it is difficult for a system to provide a good recommendation because model does not have sufficient information about user's preferences, which results as a cold start for a new user.

• Sparsity: When for large-item set the majority of the users do not rate most of the item, the user-item matrix becomes spars, so that such items cannot be recommended until some users rate them, which results in poor recommendation.

• Scalability: This issue may arise due to a large real-world dataset. Because for extremely large dataset it is difficult to produce satisfactory results.

c.) Hybrid Approach

This technique combines the advantages of two or more filtering techniques to overcomes their limitations [3]. Hybrid filtering is commonly centered on probabilistic or bio-inspired methods such as fuzzy genetic, Bayesian networks, clustering, neural networks and latent features [8]

B. Dataset for Movie Recommender System

Various sources provide movie dataset which may useful in movie recommendation system. One can also generate their own customize dataset. Here, I have mentation the most popular and commonly used dataset by a majority of researchers in the area of movie recommendation

MovieLens dataset: This dataset was published by the GroupLens research group at the University of Minnesota. MovieLens is a collection of movie ratings and comes in different sizes where size refers to the number of ratings contained by dataset, for eg. MovieLens 100K, MovieLens 1M, MovieLens 10M, MovieLens 20M, etc. MovieLens is the most commonly used dataset in the research of movie recommendations. The data provided by movieLens has been collected over the past 20 years from students of university and people on the internet. MovieLens has a website where you can sign up, contribute your ratings and can receive recommendations.

Netflix dataset: This is the user-movie rating data used in the Netflix Prize competition. The dataset consists of about 100 million movie ratings provided by 480,000 people for 17,000 movies.

C. Related Work

A lot of work has been done especially in the movie recommendation field. Social networking sites like Google, Facebook, Amazon, Netflix gain a lot of attention and becomes so successful due to their extraordinary recommendation system. Anuranjan Kumar, et al [6] provides a comparative analysis of various Metrics Used in collaborative filtering for recommendation systems such as Euclidean Distance, City-Block Distance, Tanimoto

Coefficient, Pearson's Correlation and Cosine Similarity. Ashish Pal, et al [2] provides an improved content-based collaborative filtering algorithm for movie recommendations which uses a hybrid approach, where hybrid content recommendation produced better MAE values and improved the sparsity of the dataset between 12, the results may vary when tested with a larger dataset. S. Rajaraj eswari, et al. [5] proposed a system in which they have performed a comparative study on different techniques in movie recommendation systems and came up with a hybrid recommendation algorithm to overcome to the problem of cold start. Tapestry was the first person who built a recommendation system based on CF, in which he first collected the user's rating given by different users implicitly or explicitly. And then provided ratings for an active user using CF [9]. Mukesh Kumar, et al. [9] proposed a model for item-based collaborative filtering in movie recommendation, which is quite dynamic and can regenerate fresh recommendation based on the up-dates done by the subscribers in real-time. This system is more suitable for large dataset but model accuracy is somewhat low as compared to contemporary recommendation models. Yassine Afoudi, et al. [10] provides a study on the impact of feature selection on content-based recommendation system. This paper used different feature selection methods with different similarity measurements.

Table 2. Comparative study on related works and algorithms

Technique	Algorithm	Drawback	Remarks
Content based and Collaborative filtering	K means	For a larger dataset, if an initial partition is not proper then efficiency will decrease	
Memory based CF	User based	It increases computational cost as the number of users are generally higher than the number of items in the system, easy to game	Compare to content-based technique it is more accurate
Model based CF	KNN	Not suitable for a highly sparse dataset	
Model based CF	Decision tree	For item evaluation, recommendation list needs to examine each time, Not work well if dataset consists of highly correlated items	Provides better result compare to other classifiers like SVM
Model based CF	Bayesian classifiers	Not suitable for rapidly updated system	Works well for noisy data
Model based CF	Matrix Factorization (eg. SVD)	Not work well for a nonlinear dataset, the model building process is computationally expensive	Solves the issue of scalability and sparsity

III. AUTOENCODER FOR RECOMMENDATION SYSTEM

A.Autoencoder

Autoencoder is feed forward, non-recurrent deep neural network based on un- supervised learning, which only needs unlabeled data this is just a set of input data rather than input-output pair. AE is trained to copy its input to its out- put, with the typical purpose of dimension reduction [1]. Data denoising and dimensionality reduction for data visualization are considered as two main interesting practical applications of autoencoders [11]. The basic structure of an autoencoder consists of three layers: input layer, hidden layer, and output layer.

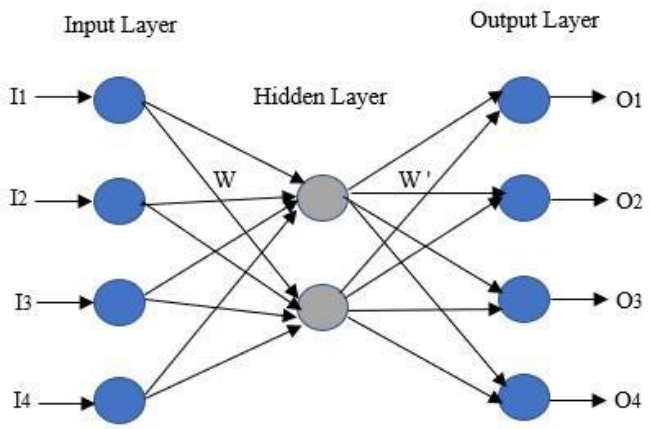

Figure 2: Basic autoencoder model

The transition from the input to the hidden layer is known as the encoding step and transition from hidden to the output layer is known as decoding step. The Encoder part of the network compresses the input into a latent-space representation. It can be represented by an encoding function h=f(x). Decoder part of the network aims to reconstruct the input from the latent space representation. It

can be represented by a decoding function y=g(h). The number of neurons in output layer is equal to the number of neurons in input layer. With multi-layer AE, the most middle layer is known as bottleneck layer(code). This bottleneck is a hidden layer that has a much smaller dimensionality than the input. For input vector x, encoding and decoding part can represent as:

Encoding : $f(x) = s(W x + b)$ (1)

Decoding : $g'(z) = l(W' z + b')$ (2)

Where W is a weight matrix, b is a bias vector, s is any activation function and l is typically an identity function [12]. Autoencoder uses a backpropagation algorithm to apply training. The purpose of network learning is to make the output signal and input signal as similar as possible. This similarity is represented by the reconstruction error. The main objective of AE is to minimize the reconstruction error between the input and output. Reconstruction Error is typically a loss function that calculates the difference between original input and produced output.

Algorithm 1: Basic Learning algorithm for AE

Step 1:Take the input vector x
Step 2:Encode the input vector x into another vector h which is a lower dimension vector thanthe input h=f(Wx+b)
Step 3:Decode the vector *h* to recreate the input y=g(W'h+b')
Step 4: **Calculate the Reconstruction error**
Step 5:Back propagate from output layer to the input layer to update the weights
Step 6: Continue until desired threshold for loss is obtained

B. Variants of Autoencoder

In recent years, various forms of AE have appeared in deep learning. Meanwhile, many variants of AE are used in RS. This section briefly introduces common variants of AE used in recommender systems.

Deep Autoencoder: Deep AE consists of two identical deep belief networks, one network for encoding and another for decoding. It consists of one input layer, one output layer and a minimum 2 to 3 hidden layers, where the output of each hidden layer is connected to the input of the successive hidden layer. By training the input data it acquires learned data. The learned data from the previous layer is used as an input for the next.

Denoising Autoencoder: It is a basic AE consisting of only one hidden layer. DAE is trained to reconstruct a clean 'repaired' input from a corrupted version. In doing so, the learner must capture the structure of the input distribution to reduce the effect of corruption [13]. DAE aims to force the hidden layer to acquire more robust features and to prevent DAE from simply learning the identity function.

Stack denoising autoencoder: SDAE stacks several DAEs together to get higher level representations. The training is conducted layer by layer. Although SDAE has advanced performance, it still has some drawbacks. The main draw- back of SDAE is the high computational cost of training and lack of scalability to high-dimensional features.

Sparse autoencoder: Sparse AE has hidden nodes greater than input nodes. An advancement to sparse autoencoders is the k-sparse autoencoder. Here we choose k neurons with the highest activation functions and adjusting the threshold to find the largest neurons. Sparse autoencoders have a sparsity penalty which is applied on the hidden layer in addition to the reconstruction error. This prevents overfitting [14].

Variational Autoencoder: VAE is same as deep AE where, bottleneck vector is replaced with two different vectors one representing the mean of the distribution and the other representing the standard deviation of the distribution. Loss function in variational autoencoder consists of two terms. First, one represents the reconstruction loss and the second term is a regularize. The encoder and decoder of VAE can be multilayer perceptron (MLP), convolutional network (CNN) or recurrent neural network (RNN) [15].

The figure:3 summarized the paper counts for different variants of AE. we considered 40 most recently published papers of various well-known publications in different fields of the recommendation system which makes use of Autoencoder based deep learning approach.

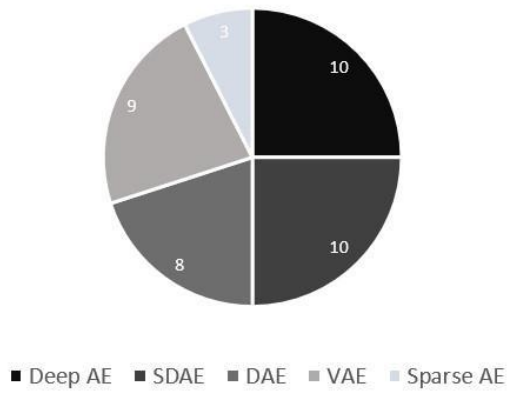

Figure 3: Autoencoder in recommender system paper count

C. Autoencoder based Deep Learning approach in Movie Recommender System

Table 3. Related work on Autoencoder based movie recommendation

Paper	Technique	Dataset	Remarks
Jeffrey Lund, et al, "Movie Recommendations Using the Deep Learning Approach" [1]	Deep AE	Movie-Lens	The experimental results show that this approach outperforms standard CF techniques KNN and matrix-factorization in terms of 0.4209 and 0.3544 RMSE on Training and Testing predicted ratings respectively.
Hanting Chu, et al, "Towards a Deep Learning Autoencoder algorithm for Collaborative Filtering Recommendation" [15]	AE-CF (AE & clustering)	Movie-Lens 1M	MAE: 0.7025 Proposed algorithm outperforms user-based CF and SVD++ algorithms, Solves the issue of cold start and sparsity.
Jian Wei, et al, "Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items" [16]	SDAE, SVD++ (singular value decomposition)	Netflix dataset	RMSE: 1.048 Proposed model outperformed existing baseline approaches for cold start item recommendation
Xiu Wang, et al, "A Recommendation Algorithm for Collaborative Denoising Auto-Encoders Based on User Preference Diffusion" [17]	DAE	Movie-Lens 10M	Precision: 0.1913 Recall: 0.1951 Solves the issue of data sparsity
Dawen Liang, et al, "Variational Autoencoders for Collaborative Filtering" [18]	VAE	Movie-Lens 20M, Netflix	
Bo Pang, et al, "A Novel Top-N Recommendation Approach Based on Conditional Variational Auto-Encoder" [19]	Conditional VAE	Movie-Lens 1M, Movie-Lens 2k	proposes an expanded VAE recommendation framework based on multiple condition labels added with side information. Split-Merge CVAE outperforms other models
Xin Dong, et al, "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems" [20]	SDAE, matrix factorization	Movie-Lens 100K, Movie-Lens 1M	RMSE: 0.5079(100k) RMSE: 0.5023(1M) Hybrid CF model outperforms other basic CF methods in effectively utilizing side information and achieves performance improvement.

The above table describes some of the related work in the field of Autoencoder based movie recommendation system. In the table reference, used technique and dataset, and the experimental results are mentioned. From the survey, we can say that autoencoders can be used to solve the issue of cold start and sparsity of CF and to enhance the capabilities of recommendation systems.

The figure:4 summarized the paper counts in each of the three categorize of recommender models. we considered 35 most recently published papers in 2019 of various well-known publications for the movie recommendation system which makes use of Autoencoder based deep learning approach. we find that the majority of the recent publications use deep learning to enhance collaborative filtering capabilities.

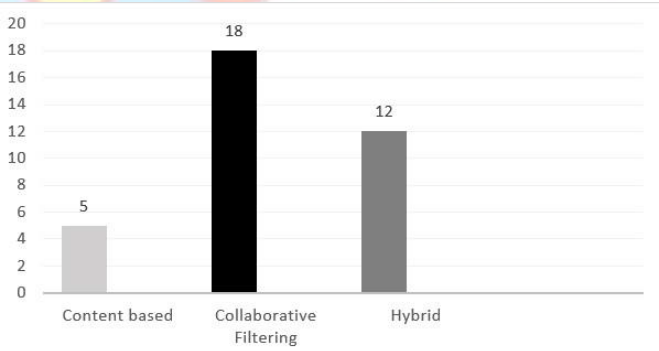

Figure 4: AE based approach in movie recommender system paper count

IV. CONCLUSION

Deep learning and recommender systems are ongoing hot research topics in re- cent decades. This paper provides a basic survey for movie recommendation system using an autoencoder based deep learning approach. In this review, firstly we have provided a basic overview of traditional recommendation systems such as content-based filtering, collaborative filtering and hybrid filtering with their limitations. Then, we have summarized basic supervised and unsupervised deep learning models used in the area of recommendation systems. Most deep

learning efforts have been towards enhancing collaborative filtering approaches. At last, we have discussed the use of autoencoder and its different variants for recommendation systems. Autoencoders can learn the latent representations of users and items from massive data, to construct a more accurate recommendation model by solving the issue of cold start and sparsity to provides an effective recommendation list for the user. As with added advantage of deep learning, this approach helps to improve the performance of RS in terms of predicting ratings. we hope that this review will be helpful for the researcher to understand the development and advantages of deep learning in recommendation systems.

References

[1] Lund, J., Ng, Y.-K, "Movie Recommendations Using the Deep Learning Approach." In: 2018 IEEE International Conference on Information Reuse and Integration (IRI). pp. 47–54 (2018)

[2] Pal, A., Parhi, P. Aggarwal, M, "An improved content based collaborative filtering algorithm for movie recommendations." In: 2017 Tenth International Conference on Contemporary Computing (IC3). pp. 1–3 (2017)

[3] Raghuwanshi, S.K., Pateriya, R.K., "Recommendation Systems: Techniques, Challenges, Application, and Evaluation." In: Bansal, J.C., Das, K.N., Nagar, A., Deep, K., and Ojha, A.K. (eds.) Soft Computing for Problem Solving. pp. 151–164. Springer, Singapore (2019) (a)

[4] S., K., Badre, R.R, "Principles and Methods for Recommendation Framework." In: 2018 4th International Conference on Computing Communication and Automation (ICCCA). pp. 1–6 (2018)

[5] Rajarajeswari, S., Naik, S., Srikant, S., Sai Prakash, M.K., Uday, P., "Movie Recommendation System." In: Shetty, N.R., Patnaik, L.M., Nagaraj, H.C., Hamsavath, P.N., and Nalini, N. (eds.) Emerging Research in Computing, Information, Communication and Applications. pp. 329–340. Springer, Singapore (2019)

[6] Kumar, A., Gupta, S., Singh, S.K., Shukla, K.K., "Comparison of various metrics used in collaborative filtering for recommendation system." In: 2015 Eighth International Conference on Contemporary Computing (IC3). pp. 150–154 (2015)

[7] Raghuwanshi, S.K., Pateriya, R.K., "Collaborative Filtering Techniques in Recom- mendation Systems." In: Shukla, R.K., Agrawal, J., Sharma, S., and Singh Tomer, G. (eds.) Data, Engineering and Applications: Volume 1. pp. 11–21. Springer, Singapore (2019) (b)

[8] Batmaz, Z., Yurekli, A., Bilge, A., Kaleli, C., "A Review on Deep Learning for Recommender Systems: Challenges and Remedies." Artif Intell Rev. 52, 1–37 (2019).

[9] Kharita, M.K., Kumar, A., Singh, P., "Item-Based Collaborative Filtering in Movie Recommendation in Real time." In: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). pp. 340–342 (2018)

[10] Oudi, Y., Lazaar, M., Al Achhab, M. "Impact of Feature selection on content-based recommendation system." In: 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS). pp. 1–6 (2019)

[11] Chu, H., Xing, X., Meng, Z., Jia, Z., "Towards a Deep Learning Autoencoder algorithm for Collaborative Filtering Recommendation." In: 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC). pp. 239–243 (2019)

[12] Badino, L., Canevari, C., Fadiga, L., Metta, G., "An auto-encoder based approach to unsupervised Learning of sub word units." In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7634–7638 (2014)

[13] Deng, J., Zhang, Z., Eyben, F., Schuller, B., "Autoencoder-based Unsupervised Domain Adaptation for Speech Emotion Recognition." IEEE Signal Process. Lett. 21, 1068–1072 (2014). https://doi.org/10.1109/LSP.2014.2324759

[14] Wilkinson, E. (2018). Deep Learning: Sparse Autoencoders. [online] Eric Wilkin- son. Available at:http://www.ericlwilkinson.com/blog/2014/11/19/deep-learning- sparse-autoencoders [Accessed 29 Nov. 2018]

[15] Zhang, G., Liu, Y., Jin, X., "A survey of autoencoder-based recommender systems." Front. Comput. Sci. 14, 430–450 (2020). https://doi.org/10.1007/s11704-018-8052-6

[16] Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z., "Collaborative filtering and deep learning-based Recommendation system for cold start items." Expert Syst. Appl. 69,29–39 (2017). https://doi.org/10.1016/j.eswa.2016.09.040

[17] Wang, X., Liu, X., Xu, X., "A recommendation algorithm for collaborative denoising auto-encoders based on user preference diffusion." In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). pp. 447–450 (2017)

[18] Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T. "Variational Autoencoders for Collaborative Filtering." In: Proceedings of the 2018 World Wide Web Conference. pp. 689–698. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2018)

[19] Pang, B., Yang, M., Wang, C., "A Novel Top-N Recommendation Approach Based on Conditional m Variational Auto-Encoder." In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., and Huang, S.-J. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 357–368. Springer International Publishing, Cham (2019)

[20] Dong, X., Yu, L., Wu, Z., Sun, Y., Yuan, L., Zhang, F., "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems." In: Thirty-First AAAI Conference on Artificial Intelligence (2017)