International Journal of Engineering & Technology, 7 (4.44) (2018) 137-142

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Increase Robustness SDAE with Imputing Missing Value To Eliminate Users Sparse Data in Case E-Commerce Recommender System

Hanafi[1]*, Nanna Suryana[2], Abd Samad Hasan Basari[3]

[1]*Department of Computer Science, University of Amikom Yogyakarta, Yogyakarta, Indonesia*
[1,2,3]*Faculty of Information and Communication Technology, University Teknikal Malaysia Melaka, Melaka, Malaysia.*
**Corresponding author E-mail: hanafi@amikom.ac.id*

Abstract

Online shopping needs a computer machine to serve product information sale for customer or buyer candidate. Relevant information served by ecommerce system famous called recommender system. The successful to applied, it will have impact to increase of marketing target achievement. The character of information served by recommender system have to be special, personalized, relevant and fit according customer profiling. There are four kind of recommender system model, however there is one model that was successful to be applied in real ecommerce industry that popular named collaborative filtering. Collaborative filtering approach need a record users or customers activity in the past to generate recommendation for example rating record, purchasing record, testimony about product. The majority collaborative filtering approaches rely on rating as fundamental computation to calculate product recommendation. However, just a little number of consumers who willing give rating for products less than a percent, according to several convince datasets such MovieLens. This problem causes of sparse product rating that will impact to product recommendation accuracy level. Sometime, in extreme condition, it is impossible to generate product recommendation. Several efforts have been conducting to handle product sparse rating, however they fail to generate product recommendation accurately when face extreme sparse data, such as matrix factorization family include SVD, NMF, SVD++. This research aims to develop a model to handle users sparse rating involving deep SDAE. One of the efforts to produce better output in handling this data sparse, our strategy is to imputing missing value by statistical method so that the input in SDAE is closer to the feasibility of data that is not too sparse. According to our experiment involve deep learning, TensorFlow, MovieLens datasets, evaluation method by root mean square error (RMSE), our approach involves reducing input missing value could address users sparse rating and increase robustness over several existing approach.

Keywords: *recommender system, collaborative, deep SDAE, deep learning, sparse data, users sparse*

1. Introduction

E-commerce has moved the way in many companies do transaction. To them, e-commerce is no longer an alternative but an imperative. Many companies are struggling with the most basic problem: what is the best approach for developing and doing business in the digital economy? Some companies are moving their businesses entirely to the Web (e.g.egghead.com). Some are establishing subsidiaries, then spinning them off as separate online business entities (e.g. barnesandnoble.com) [1]. E-commerce need a system to serve relevant information about product to deliver through web sites portal or mobile phone, the system namely recommender system. Recommendation system is an essential part in industry area [2]. It is a critical tool to promote sales and services for many online websites and mobile applications. For instances, 80 percent of movies watched on Netflix came from recommendations [3], 60 percent of video clicks came from web sites recommendation in YouTube [4]. According [5] found that sales agents with recommendations by the NetPerceptions system achieved 60% higher average cross-sell value and 50% higher cross-sell success rate than agents using traditional cross-sell techniques based on experiments conducted at a U.K.-based retail and business group, GUS pls (www.gus.co.uk).

Recommender system divided into 4 types based on technical approach [6], 1). Content based: the mechanism of recommendation involves product classification approach. It is tending information retrieval to generate product recommendation 2). Knowledge based: this method develops for specific necessary recommendation, the specific character is to provide product information rarely needed for individuals for example house, loan, insurance, car. 3). Demographic based: product recommendation in which established to provide product recommendation based demographic information. 4). Collaborative Filtering: product recommendation based on user behavior in the past for example the term of behavior is rating, comment, testimony, purchasing and etc. Example display of e-commerce portal shown on figure 1 as Amazon e-commerce portal.

Recommender system based on content-based relies on the classification of items or products. This technique is more likely to use information retrieval, but this method is not liked by users because it tends to be boring. for example, people who buy sugar also buy coffee, people who buy coffee also buy milk. different from the approach taken on the Collaborative filtering approach that relies on user behavior [7].

Fig 1. Example of e-commerce portal

This method is the most successful approach that applied in many large e-commerce companies, due they have ability to provide recommendation character as follow; provide product fit, serve relevant information, accurate, serendipity. In common collaborative filtering adopted rating as explicit feedback to calculate similarity user for product using rating matrix to generate product recommendation. The biggest challenge in collaborative filtering is just only a little user who gave rating for product approximately only less than 1 percent. This problem popular called sparse data also in extreme condition sparse famous called cold start. When cold start happens, there is no recommendation possible generated by system. Refer to figure 2 is example of unrating product by customer. It is main reason why sparse rating happened.

Fig 2. Example of movie rating

In the first time established in the mid 90s, applied of recommender system based collaborative filtering involve statistical approach such as Cosine similarity, Spearman rank, and etc. popular mention as memory based. This method having advantage in simplicity, justify. However, the method having shortcoming in sparse data, cold start, scalability issue, accuracy.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Equation 1. Memory based approach to calculate similarity

$$\cos ine\_sim(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

Equation 2. Cosine similarity approach

The major challenge in recommender system problem is how to handle extreme sparse data accurately [8]. Some of the efforts that have been done by researchers to overcome this are enhancing auxiliary information resource and enhance machine learning approach [6].
The one of the major problems in collaborative filtering is sparse data [6], various efforts have been made by researchers, but the issue in this research is still opened considering that sparse data in the case of collaborative filtering is very extreme, namely where the explicit number of ratings is less than 1 percent, according to reliable dataset reference [9]. There are two kinds of sparse data classification, the first namely user's sparse data which means lack of information about rating on users and the second is item sparse rating that is the lack of rating on an item. Several efforts that have been made to eliminate this problem, for example in the case of items sparse data is to extract content feature, in the case of music recommender systems is classified base audio feature [10], online fashion shop by classification use color feature extraction [11], recommender for news base text content classifier [12], video recommendation by YouTube researcher involve key word, video description, demographic information use deep learning [13][4].

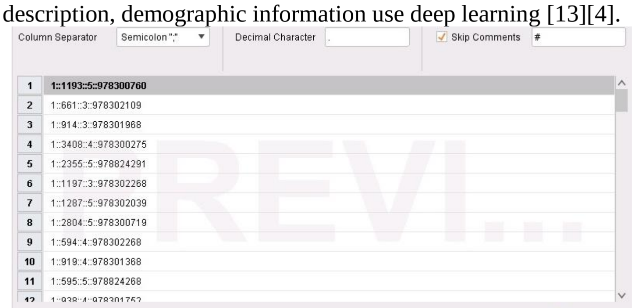

Fig 3. Dataset contain user and item information

Sparse data in recommender system research field is divided into two kind: cold start items and cold start users, the illustration fig. 3 on above shows one example of the structure of the dataset in the movie recommender system which contains, user id in the form of numbers, movie id also in the form of numbers and rating values in the form number between range 1-5, then the timestamp which shows exactly the time was given by system when users giving the rating.

Table 1. Characteristic of MovieLens datasets

Datasets (ML)	#users	#movie	#rating	sparse	years
100k	943	1.682	100,000	6.3%	4/1998
1M	6.040	3.900	1.000.209	4.2%	2/2003
10M	71.567	10.681	10.000.054	1.3%	1/2009
20M	138.492	27.278	20.000.263	0.52%	10/2007

2. Previous Work

Auto encoder is a part of neural network variant. Autoencoders are categorical of unsupervised networks where the output of the Network plan to reconstruct the first input. The Network is constrained to use narrow hidden layers, forcing a dimensionality reduction on the data. In this research SDAE will be used to find the latent factor user and rating. Some of the previous studies were classical method such Nearest Neighbor, Non-Negative Matrix Factorization, Singular Value Decomposition (SVD). SVD is popular algorithm that popularized by Simon Funk since the Netflix competition, according prediction approach as follow $\hat{r}_{u,i} = \mu + b_u + b_i + q_i^T p_u$. Then there is the development of the SVD variant, SVD ++ by Koren [14], which is by using a formula $\hat{r}_{u,i} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$, indeed, the use of variant matrix factorization can produce an accurate recommender product, but it will produce inaccurate recommendation when face with sparse rating data, and several experts are still arguing on SVD linearity in generating user rating latent factor.

Deep learning, one of the machine learning approaches used to solve data sparse problems and overcome some of the weaknesses in several approaches described above. Deep learning is a derivative of a resurgent neural network because it can produces outperform result in image processing, natural language processing. Some experts attempt to empower deep learning in dealing with problems that exist in the recommender system. Several study was use deep learning to extract content feature aims to handle sparse data in rating product, such as [15] enhance Deep Convolutional Neural Network to create audio music classifier aim to develop auto music recommendation, due no data product record, the recommendation difficult to generate.

According [16] where author conducting approach to eliminate sparse data by combining between multi-layer neural network and non-negative matrix factorization, even the result of hybridization could be work, however the results are less accurate than some studies that use similar technique.

Table 2. Previous work

No	Method	Description	Ref.
1	PMF	Probabilistic Matrix Factorization (PMF) is standard rating prediction approach that only involve rating for collaborative filtering	[17]
2	CTR	Collaborative Topic Regression (CTR) is a state-of the art recommendation model, which combines collaborative filtering (PMF) and topic modeling (LDA) to use both ratings and documents.	[18]
3	NMF	Recommender system based on collaborative filtering applied use non-negative matrix factorization (NMF) to generate recommendation	[19]
4	CDL	Collaborative Deep Learning (CDL) is another state-of- the-art recommendation model, which enhances rating prediction accuracy by analyzing documents using Deep Learning approach.	[20]
5	SVD	Recommender system based on collaborative filtering applied use Singular value decomposition (SVD) as low rank dimensional factorization.	[21]

3. Our Approach Model

This research focus to develop a method to handle user rating sparse data extremely in case collaborative filtering recommender system involve deep learning machine to find relationship latent factor between user rating from a product. According our best knowledge, in state of the art by use matrix factorization has failure to addressing user rating sparse data extremely. So, the result of recommendation sometime getting the mistake [14]. This research is categorical lab scale research involves public datasets. We consider MovieLens dataset as convince datasets were applied in so many study in recommender system research field, detail specification refer to [9].

3.1. SDAE Neural Network

SDAE is based on feed forward neural network also a part of neural network family. Basic concept of this approach as follow, the form of an Autoencoder is a feedforward neural network consist an input layer, one hidden layer and an output layer as follow Fig. 4. The output layer has the equal number of neurons as the input layer for the purpose of reconstructing its own inputs. This makes an autoencoders a form of unsupervised learning, which means no labelled data are necessary just a set of input data instead of input-output connection.

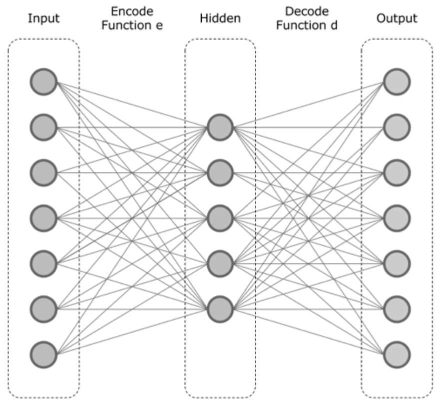

Fig 4. SDAE in shallow model

It is useful that an autoencoders has a smaller hidden layer than the input layer. This impact forces the model to make a compressed description of the data in the hidden layer by learning relationship of the data. The switch over from the input to the hidden layer is named the encoding stage and the transform from the hidden to the output layer is called the decoding step. We can also describe these transitions mathematically as a mapping.

Basic SDAE formula, the Denoising Autoencoder (DAE) loss function is modified to emphasize the denoising aspect of the network. It refers on two main hyper parameters a, b. They balance whether the network would focus on denoising the input (a) or reconstructing the input (b).

$$L_{2,a,b}(x,x) = a\left(\sum_{j\in t(x)} [nn(x)_j - x_j]^2\right) + b\left(\sum_{j\notin t(x)} [nn(x)_j - x_j]^2\right)$$

Equation 3. SDAE formula approach

Where $nn(x)_k$ is the k^{th} output of the network, $\hat{x}$ is the corrupted input x, J are the indexes of the corrupted element of x.

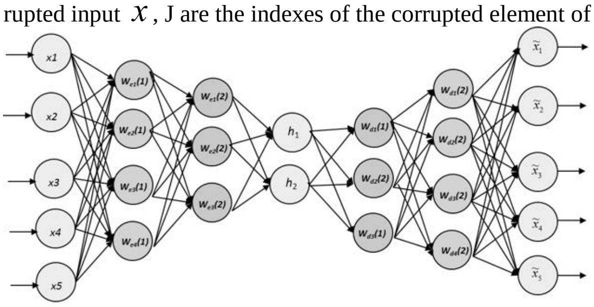

Fig 5. Deep SDAE architecture

The important stage before data training is done, it is necessary to do the initial data processing step, which is handling sparse input. We consider involve SGD (Stochastic Gradient Descent) to optimization learning for commonly use in neural network.

$$u_i = u_i - \eta \frac{\partial}{\partial u_i} L(U,V),$$

$$v_j = v_j - \eta \frac{\partial}{\partial v_i} L(U,V)$$

Equation 4. SGD optimization learning

The optimization approach by SGD consist two correspondents between users rating optimization and product rating optimization, the detailed SGD formula show in equation as follow equation 5 and equation 6.

$$\frac{\partial}{\partial u_i} L(U,V) = \alpha \sum_i (s_i^{(u)} - \hat{s}_i^{(u)}) \frac{\partial \hat{s}_i^{(u)}}{\partial u_i} +$$
$$\lambda u_i + (1-\alpha) \sum_i (x_i - \hat{x}_i) \frac{\partial \hat{x}_i}{\partial \hat{u}_i} -$$
$$\sum_{i,j \in I} (R_{ij} - u_i v_j^T) v_j$$

Equation 5. Optimization learning for V

$$\frac{\partial}{\partial u_j} L(U,V) = \alpha \sum_j (s_j^{(i)} - \hat{s}_j^{(i)}) \frac{\partial \hat{s}_j^{(i)}}{\partial v_j} +$$
$$\lambda v_j + (1-\alpha) \sum_j (y_j - \hat{y}_j) \frac{\partial \hat{y}_j}{\partial v_j} -$$
$$\sum_{i,j \in I} (R_{ij} - u_i v_j^T) u_j$$

Equation 6. Optimization learning for U

Reference to neural network architecture in general, it requires a activation function in this approach we involving to use tanh. Example function show in figure 6.

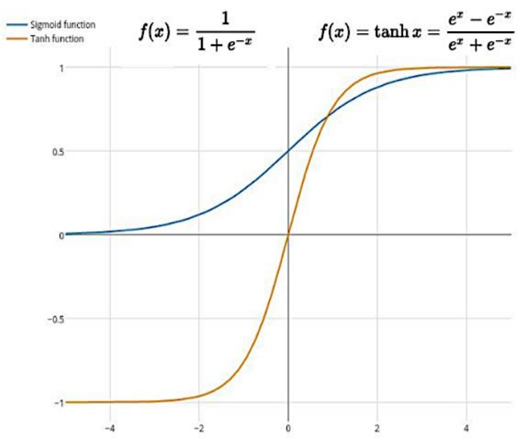

Fig 6. Activation function

3.2. Strategy to Handle Imputing Missing Value

We consider involving alternative approach to inputting missing value strategy that adopted from [22] We propose another approach in contrast to the past methodologies: by decide to consider imputing missing rating to item we attempt to enhance the SDAE performance without independently handling cold start and finish precedents. By utilizing this methodology, we attempt to "repair" the input matrix aim to eliminate the expectation recommendation problem. The thought for our methodology is likewise identified with discovering that utilizes unlabeled integrate with the marked data to develop more precise classifiers.
Our methodology derives missing rating from different users and totals them. This idea is identified with referred by [23], who demonstrated that the incorporation of a user method shows with trust and doubt systems to recognize dependable users is proficient exposed to the cold start issue. Growing user profiles with items that are equal items that have just been rated has additionally been appeared to be valuable.
We involve a methodology that takes a solitary user exposed to the cold start state (we mean this user with cold start users; take note of that the cold start can be either total or halfway) and continues with the accompanying four stages (outlined in figure 4):

1. Discover certain % of user that are like a given cold start user.
2. Select applicable traits (items) for the attribution procedure.
3. Total rating from comparative users and impute the outcome into rating for the Cold start user.
4. The final completeness matrix applied as first layer in SDAE input.

3.3. Datasets

As previously mentioned you will learn to estimate the rating a user would give a movie. For that matter we will use the popu*lar MovieLens* dataset [9]. *MovieLens* is a web-based recommender system and online community that preference movies for its users to watch. More especially we will apply the *ml-1m.zip* dataset that contains 1,000,209 anonymous ratings of about 3,900 movies created by 6,040 *MovieLens* users. The import file we need is *ratings.dat.* This file contains 1,000,209 lines all having the following format: "user_id::movie_id::rating:time_stamp". The explanation contain data table is user_id is number identity of user, movie_id is identity of movie id, rating is value of rating start that given by user for movie, time_stamp is detailed of the time when user give the rating.

3.4. Pre-processing

The authors considers two kinds of datasets ml-100k and ml-10M and also datasets contains divides into training data and test data within 50/50, 40/60, 30/70, 20/80, 10/90. This is very important to determine the level of accuracy in the condition of the amount of data capacity and the proportion of data ration between data testing and data training in testing using RMSE.

3.5. Evaluation Metric

RMSE is frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed [24]. The RMSE represents the sample standard deviation of the differences between predicted values and observed values. Root Mean Squared Error (RMSE) is might the most famous metric used in evaluating accuracy of predicting rating. The system generates predicted ratings $\bar{r}_{ui}$ for test set t of user-item pairs (u,i) for which the true rating r_{ui} are known. Typically, r_{ui} are known because they are hidden in an offline experiment. The RMSE between the predicted and actual rating is given by:

$$RMSE = \sqrt{\frac{1}{|t|} \sum_{(u,i)\in t} (\bar{r}_{ui} - r_{ui})^2}$$

Equation 7. Evaluation metric by RMSE

3.6. Device and Tools

Our experiment involves several tools include software and hardware. There are several tools and software that involve includes Anaconda as web interface for Python utilities with several libraries such as tensor flow for deep learning implementation and GeForce GTX 1001 for running convolutional neural networks supported by processors that we use Intel Xeon 2.4 Ghz quad Core.

Table 3. List of device and library use

No	Device/tools/ library	Specification
1	Processor	Intel Xeon Quad core, 2.4 Ghz
2	Memory	16 Gb
3	GPU	GeForce GTX 1001
4	Tensor Flow	Deep learning tools
5	Keras	Deep learning tools
6	Anaconda	Web interface
7	Python	Tool programming
8	Scikit-learn	Handle sparse module
9	Surface	Recsys data analytic
10	Torch	Missing value

4. Result and Analysis

According to our experiment, the result of training test consists four ratio training datasets, show in table 4. Refer to our observations use table 4 on below, the performance of NN-SDAE is outperform than the NMF (Non-negative Matrix Factorization). Refer our best knowledge, they use preprocessing with reduce input missing value by random approach to predict value of input. However, when compared to the other methods, NN-SDAE is still losing, but the important notes are NN-SDAE only relies on user and rating information, so that it can work maximum to predict rating accurately. We expected when NN-SDAE applied additional auxiliary information, the performance will be work better.

According to our best knowledge, current research, many involve additional information to increase effectiveness users latent factor representation in predicting rating values. SVD still have good performing over new approach. One of the approaches is possible to imbedded with additional information. In the observation based on the table data, it can be concluded that this system can work as generally existing models for example by increasing the training data ratio, the level of accuracy also increases. we will try to apply this model to a larger amount of data so that the phenomenon will be more proven.

Table 4. Result and comparison experiment

	Ratio of training set (MovieLens)			
Model	20%	40%	60%	80%
NMF	0.90481	0.88182	0.87413	0.86912
PMF	0.93842	0.88161	0.85072	0.82703
SVD	0.87920	0.83712	0.81283	0.79521
SDAE-NN	**0.90093**	**0.89526**	**0.88581**	**0.87145**

5. Conclusion

NN-SDAE obtains good result in recommendation compare another existing approach, even though without additional auxiliary information can produce rating predictions that are quite accurate compared to NMF approaches, the NMF approach has similarities with the NN-SDAE approach which only relies on user data and rating components and improve strategy to reduces missing values in the input data before pre-processing. It is different with some other approaches that involve additional information and produce a rating prediction that is more accurate in evaluation using standard deviations using RMSE. The benefit of unless additional information is they no need collect the other information which is sometime not easy to collect. The second benefit is not need high computation to much more training the data before produce the product recommendation.

Our motivation to improvement in the further research ,we has plan to trying involve add additional information derived from user information to increase effectiveness user latent factor representation, so they have impact in recommendation performance, and also will implemented in greater amount of data different character of datasets.

Acknowledgement

we would say to thank for those support our experiment research. The organization that support our research is university Teknikal Malaysia Melaka, Melaka, Malaysia, University of Amikom Yogyakarta, Yogyakarta, Indonesia and also Time Excelindo, Yogyakarta, the Holding Company in Indonesia. I never forget, my deep thank for double blind reviewer who have given critique and suggestion to make improvement for this research paper. May Allah always blessing you all.

References

[1] G. R and G. J, "Get the right mix of bricks and clicks," *Harv. Bus. Rev.*, 2000.

[2] Hanafi, N. Suryana, and A. Sammad, "Evaluation of e-Service Quality, Perceived Value on Customer Satisfaction and Customer Loyalty: A Study in Indonesia." Medwell Journal, pp. 1892–1900, 2017.

[3] C. A. Gomez-uribe and N. Hunt, "The Netflix Recommender System : Algorithms , Business Value ," vol. 6, no. 4, 2015.

[4] J. Davidson and B. Liebald, "The YouTube video recommendation system The YouTube Video Recommendation System," no. August, 2014.

[5] J. Ben Schafer, J. Konstan, and J. Riedl, "E-commerce recommendation applications," *Appl. Data Min. to Electron. …*, pp. 115–153, 2001.

[6] Hanafi, N. Suryana, and A. Sammad, "AN UNDERSTANDING AND APPROACH SOLUTION FOR COLD START PROBLEM ASSOCIATED WITH RECOMMENDER SYSTEM : A LITERATURE REVIEW," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 09, 2018.

[7] L. Candillier, K. Jack, F. Fessant, and F. Meyer, *Recommender systems*, vol. 40, no. 3. 1997.

[8] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intell. Data Anal.*, vol. 21, no. 6, pp. 1487–1524, 2017.

[9] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19:1--19:19, 2015.

[10] X. Wang and Y. Wang, "Improving Content-based and Hybrid Music Recommendation using Deep Learning," pp. 627–636.

[11] S. Jaradat, "Deep Cross-Domain Fashion Recommendation," *Proc. Elev. ACM Conf. Recomm. Syst. - RecSys '17*, pp. 407–410, 2017.

[12] K. Park, J. Lee, and J. Choi, "Deep Neural Networks for News Recommendations," *CIKM'17, Novemb. 6-10, 2017, Singapore*, 2017.

[13] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," *Proc. 10th ACM Conf. Recomm. Syst. - RecSys '16*, pp. 191–198, 2016.

[14] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," pp. 42–49, 2009.

[15] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep Content-Based Music Recommendation," pp. 2643–2651, 2013.

[16] Hanafi, N. Suryana, and abd samad hasan Basari, "HYBRIDIZATION APPROACH TO ELIMINATE SPARSE DATA BASED ON NONNEGATIVE MATRIX FACTORIZATION & DEEP LEARNING," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 14, pp. 4502–4512, 2018.
[17] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization.," *Proc. Adv. Neural Inf. Process. Syst. 20 (NIPS 07)*, pp. 1257–1264, 2007.
[18] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '11*, p. 448, 2011.
[19] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from Incomplete Ratings Using Non-negative Matrix Factorization," *Proc. 2006 SIAM Int. Conf. Data Min.*, pp. 549–553, 2006.
[20] H. Wang and D. Yeung, "Collaborative Deep Learning for Recommender Systems arXiv : 1409 . 2944v1 [cs . LG] 10 Sep 2014," no. July 2015, 2014.
[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems," *Am. Lab.*, vol. 37, no. 10, p. 4, 2005.
[22] U. Ocepek, J. Rugelj, and Z. Bosni??, "Improving matrix factorization recommendations for examples in cold start," *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6784–6794, 2015.
[23] C. C. Chen, Y. H. Wan, M. C. Chung, and Y. C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Inf. Sci. (Ny).*, vol. 224, pp. 19–36, 2013.
[24] A. Gunawardana, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," vol. 10, pp. 2935–2962, 2009.