Indonesian Journal of Electrical Engineering and Computer Science
Vol. 30, No. 3, June 2023, pp. 1805~1812
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v30.i3.pp1805-1812

Cluster-based denoising autoencoders for rate prediction recommender systems

Ammar Abdulsalam Al-Asadi[1], Mahdi Nsaif Jasim[2]
[1]Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq
[2]College of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq

Article Info

Article history:

Received Sep 19, 2022
Revised Feb 10, 2023
Accepted Feb 18, 2023

Keywords:

Clustering
Deep learning
Denoising autoencoder
K-means optimization
Recommender system

ABSTRACT

Recommender system (RS) is a suitable tool for filtering out items and providing the most relevant and suitable items to each user, based on their individual preferences. Deep learning algorithms achieve great success in several fields including RS. The issue with deep learning-based RS models is that, they ignore the differences of users' preferences, and they build a model based on all the users' rates. This paper proposed an optimized clustering-based denoising autoencoder model (OCB-DAE) which trains multiple models instead of one, based on users' preferences using k-means algorithm combined with a nature-inspired algorithm (NIA) such as artificial fish swarm algorithm to determine the optimal initial centroids to cluster the users based on their similar preferences, and each cluster trains its own denoising autoencoder (DAE) model. The results proved that combining NIA with k-means gives better clustering results comparing with using k-means alone. OCB-DAE was trained and tested with MovieLens 1M dataset where 80% of it is used for training and 20% for testing. Root mean squared error (RMSE) score was used to evaluate the performance of the proposed model which was 0.618. It outperformed the other models that use autoencoder and denoising autoencoder without clustering with 38.5% and 29.5% respectively.

Corresponding Author:

Ammar Abdulsalam Al-Asadi
Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics
Baghdad, Iraq
Email: phd202010553@iips.icci.edu.iq

1. INTRODUCTION

Due to the rapid growth of internet applications and services, a vast amount of information is generated daily. This results in a phenomenon called information overload, making it challenging for users to find relevant information [1]. To address this problem, researchers have developed recommender systems (RSs), which filter out irrelevant services and items and provide personalized recommendations to users based on their historical preferences [2]. RSs can also incorporate additional information such as user profiles or item features [3]. They are mainly classified into three categories: collaborative filtering (CF), content-based (CB), and hybrid filtering [4]. CF is the most efficient and simple method among them. It has been used in many real-world systems such as Amazon. CF is divided into user-based and item-based filtering depending on the adopted prediction technique [5].

Clustering is one of the unsupervised learning techniques, it is widely used in recommender system (RS) to group the users or items based on their similarity. Each group is called a cluster and each cluster contains very similar members by a given data properties [6]. Despite being the most commonly used clustering method, k-means is known to have several disadvantages, such as being influenced by the initial centroids and being sensitive to the initial parameter settings [7]. However, nature-inspired algorithms have been shown to be more effective in overcoming these weaknesses, as they have demonstrated their superiority over traditional clustering techniques through their swarm behavior, which enables them to achieve optimal solutions in a cooperative and organized manner [8]. The particle swarm optimization (PSO) algorithm is a well-known algorithm that was inspired by natural phenomena. It was introduced in 1995 by Kennedy and uses a population and probability approach to solve optimization problems [9]. There are several other nature-inspired algorithms, including the artificial fish swarm algorithm (AFSA) and ant colony optimization (ACO). AFSA was developed based on the social behavior of fish in swarms [10], while ACO was inspired by the foraging behavior of ants searching for food [11].

Deep learning achieves great success in several fields of applications such as speech recognition, computer vision, and natural language processing [12]. Academia and industry are becoming interested in applying deep learning to different applications because it is able to solve many problems and achieve high-quality results [13]. Recently, deep learning-based recommender systems have been actively investigated [14], where each user and item features are combined (or averaged, concatenated) to make predictions by following several perceptron layers. Deep structured models [15] look into users' textual behaviors (search queries and browsing histories) and textual content, then maps the users and items into a latent representation where the similarity among the users' preferences is maximized. Several research works try to combine collaborative filtering and deep learning into a collaborative deep learning-based recommendation in recent years [16]–[19]. Autoencoder is an approach recently introduced into the recommender system where non-linear matrix factorization is computed by the autoencoder framework with user-item ratings [20], [21].

The problem with deep learning-based CF models is that, they use all users (or items) in dataset to build the latent space which will be used later to predict the missing rates of each user, where the users with different interests will contribute to generating the predictions. As a result, the prediction accuracy will be low, because they are generated by users with different preferences.

This research proposes an optimized clustering-based denoising autoencoder model. This model is different from other deep learning-based models as it trains multiple models instead of one, based on users' preferences using k-means algorithm combined with a nature-inspired algorithm to determine the optimal initial centroids to cluster users based on their similar interests, and each cluster trains its own denoising autoencoder model. The rest of this paper is organized as follows, section 2 discusses the techniques that are used in the proposed system, section 3 presents the proposed method, the results are presented in section 4, and section 5 discuss the conclusions.

2. THE COMPREHENSIVE THEORETICAL BASIS

The research proposes an optimized clustering-based denoising autoencoder model (OCB-DAE). It is based on clustering optimization and autoencoders techniques. These topics will be briefly discussed in the following sub-sections.

2.1. Clustering optimization

K-means is one of the clustering algorithms in partitioning methods. It is the simplest, most used, and computationally efficient clustering algorithm [22]. K-Means divides data points into clusters (K) based on clusters' centers or centroids. The centroid of each cluster is computed as the mean of all data points in that cluster. Before training the clustering model the users need to specify the number of clusters (K) [23].

Silhouette is one of the methods that are used to find the optimal number of clusters (K). It validates the consistency within clusters of data points. The silhouette method measures how much the data point is similar to its cluster compared to other clusters by computing its silhouette coefficients [24]:

$$coefficients = \frac{(b-a)}{max(a,b)} \tag{1}$$

To overcome the limitation of k-means algorithm specially the influencing by the initial centroid, nature-inspired algorithms (NIA) can be combined with k-means algorithm to determine the optimal initial centroids that give the best clustering results. NIA can be utilized as optimization technique to identify the optimal initial centroids for each cluster. They begin by randomly generating centroids and assigning them as the initial centroids in k-means algorithm. The sum of squared errors (SSE) of the k-means output is then calculated using (2), which adds up the distance each data point and its corresponding centroid. The SSE is used as a measure of fitness, and NIA endeavor to minimize it to find the best centroids [10].

$$SSE = \sum_{k=1}^{k} \sum_{\forall x_i \in C_k} \|x_i - \mu_k\|^2 \tag{2}$$

2.2. Autoencoders

Autoencoder (AE) was first presented in 1991 by Kramer [25]. It is a deep learning algorithm which obtains high-level representation of original features. Autoencoders uses feed forward neural networks to learn the input representation with a com-pact dimension. The output of AE network attempts to reconstruct the input. It back-propagates the loss to train the network through the reconstruction process, by using the two parts:

– Encoder: $x \rightarrow z$
– Decoder: $z \rightarrow x$

there is only one hidden layer in the simplest case, where the encoder takes input x and maps it to z, then the decoder maps z into reconstructed x [26].

To discover more robust features through autoencoding and learning the identity function, Vincent presented the denoising autoencoder (DAE). DAE utilizes corrupted input x as $\tilde{x}$, and trains the network to denoise and reconstruct input x. Many corruption options can be used including the additive gaussian noise and multiplicative mask-out/drop-out noise [21].

3. METHOD

The general steps of the proposed system are extracting user-genre matrix, users clustering, and build DAE model, as they are illustrated in Figure 1. These steps are applied to MovieLens 1M dataset, which contains 1,000,209 ratings of approximately 3,900 movies made by 6,040 users. Two tables are used out of this dataset:

– Movies table contains all the movies with their features. Features represent the movies' genres.
– Ratings table contains the rates (in a scale 1 and 5) that given by users to movies.

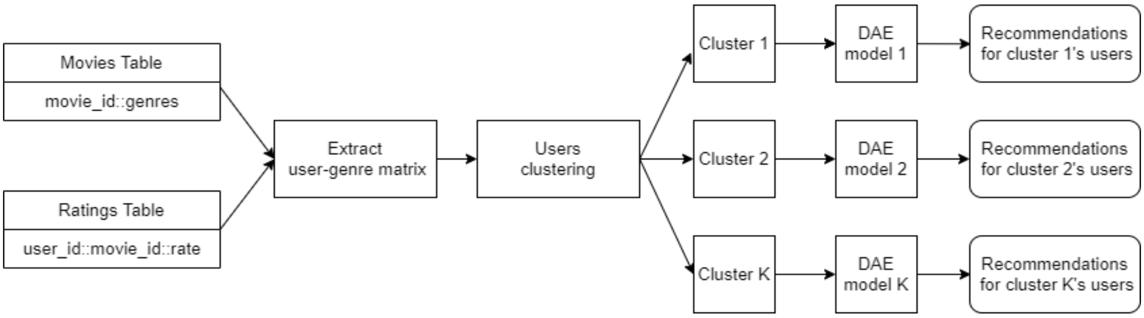

Figure 1. General diagram of the proposed system

3.1. Extract user-genre matrix

The first step of the proposed system is loading the necessary data such as movies and ratings tables to be used for extracting the users' preferences. Movies and ratings tables are merged to produce a single table that contains all the users as rows and all the movies' genres (18 genres) as columns. The intersection them represents the average rate that given by a user to a movie's genre. Part of the users-genres matrix is shown in Figure 2.

user_id	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance
1	4.200000	4.000000	4.111111	4.250000	4.142857	4.000000	0.000000	4.428571	4.00	0.000000	0.000000	4.285714	0.000000	3.666667
2	3.500000	3.736842	0.000000	0.000000	3.560000	3.583333	0.000000	3.898734	3.00	4.000000	3.000000	0.000000	3.333333	3.708333
3	3.956522	4.000000	4.000000	4.000000	3.766667	0.000000	0.000000	4.000000	4.50	0.000000	2.666667	4.000000	3.000000	3.800000
4	4.157895	3.833333	0.000000	4.000000	0.000000	5.000000	0.000000	4.166667	4.50	0.000000	4.333333	0.000000	0.000000	4.000000
5	2.612903	3.000000	4.000000	3.833333	3.410714	3.285714	3.666667	3.096154	0.00	4.000000	2.800000	3.333333	3.125000	3.100000

Figure 2. Part of the user-genre matrix

3.2. Users clustering

The silhouette method is applied to the user-genre matrix to determine the best k value by using (1). AFSA is combined with k-means (AFSA-KM) to determine the optimal initial centroids for clustering. In this step, the users with similar interests are grouped together. The similarities among users are computed based on the average rates that each user gave to each movie's genre in user-genre matrix. The output of this step is the clusters that users belong to, this information is added as a new column to the ratings table to be used for training the model. The AFSA-KM algorithm steps are presented in Algorithm 1.

Algorithm 1. AFSA-KM algorithm

```
Input: user-genre matrix, number of clusters (K), and population size (i)
Output: users' clusters
Generate random initial K centroids X_ij for each fish AF_i
Execute k-means on user-genre matrix using X_ij as initial centroids
Compute SSE(X_ij) for each AF_i using (2)
Best=min SSE(X_ij)
While (t<Max iteration) do
     For each AF_i do
          Execute Follow Behavior on X_ij^(t)
          Execute Swarm Behavior on X_ij^(t)
          If F(X_ij^(t), follow) <F(X_ij^(t), swarm)
               X_ij^(t+1) =X_ij^(t), follow
          Else
               X_ij^(t+1) =X_ij^(t), swarm
          End if
     End for
     If F(X_AF) <F (Best)
          Best=X_AF
     End if
 End while
Execute k-means on user-genre matrix using Best as initial centroids
```

3.3. Build DAE model

Based on the clustering information extracted from the previous step, the ratings data is divided into K sets. Each set trains a DAE model by following same model structure. The proposed model is designed as an item-based model, where the input data represented in form of item-users matrix (r_i). Figure 3 shows the structure of the proposed model, which consist of the following layers:

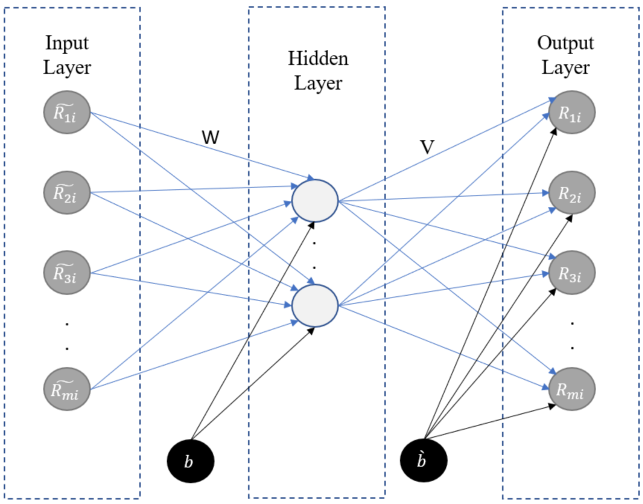

Figure 3. The structure of proposed model

3.3.1. Input layer

The number of nodes in the input layer equals the number of users within a cluster C_k. Where the input values r_i represent the rates R the are given by C_k users to an item i, r_i = {R_{1i}, R_{2i}, $R_{3i,}$ R_{mi}} where m is the number of C_k users. The input data is further corrupted $\widetilde{r}_i$ by applying dropout noise. It sets the input data to zero randomly based on a noise ratio.

3.3.2. Hidden layer

The hidden layer represents the latent space of the input data. It has a smaller number of nods, and they are fully connected to the input data $\tilde{r_i}$. The output of the hidden layer is z which is computed by:

$$z = f(W * \tilde{r_i} + b) \tag{3}$$

where, f is a non-linear activation function, W is the wights of the hidden nodes, and b is biases.

3.3.3. Output layer

The output layer has the same number of nods in the input layer. They are fully connected with the hidden layer's nods. The output of this layer is the predicted rates $\hat{r_i}$ which are computed by:

$$\hat{r_i} = f(V * z + \grave{b}) \tag{4}$$

where, f is a linear activation function, V is the wights of the output nodes, and $\grave{b}$ is the biases.

During the training, the model keeps trying to minimize the error (loss) the actual rating r_i and the predicted rating $\hat{r_i}$ through several epochs. Adam optimization algorithm [27] is used to update the model's weights to reduce the mean squared error (MSE) which computed using (5). The steps of the proposed model are illustrated in Algorithm 2.

Algorithm 2. Cluster-based denoising autoencoder RS

```
Input: Ratings data
Output: Predicted rates;
k← number of clusters
i←0
While i<=k do
     Load the ratings of cluster_i
     m←number of items rated by cluster_i users
     n←number of cluster_i users
     Generate m×n matrix r_i
     Split r_i into training and testing sets
     r̃_i ←dropout_noise (r_i, ratio)
     Build DAE model with input data r̃_i
     While epoch<max do
        Compute z using (3)
        Compute r̂_i using (4)
        Compute MSE between r_i and r̂_i using (5)
        Update parameters using Adam optimizer
     End while
     Generate predictions
     i←i+1
End while
```

$$\text{MSE}=\frac{\sum_{i=1}^{n}(r_i-\hat{r_i})^2}{n} \tag{5}$$

3.3.4. Model evaluation

To evaluate the performance of the proposed model, root mean squared error (RMSE) is used. It has a straightforward relation with MSE which is computed by:

$$\text{RMSE}=\sqrt{MMSE} \tag{6}$$

4. RESULTS AND DISCUSSION

This section presents the results of users clustering and the proposed model evaluation. Moreover, it provides the needed comprehensive discussion. Results are presented in charts and tables to help the reader understand the whole principle. Therefore, the discussion will be illustrated in the following sub-sections.

4.1. Users clustering results

The best number of clusters for the applied dataset is determine based on the silhouette coefficients score using (1). Different numbers of clusters (k) are used 2 and 20. As Figure 4 shows, the score kept decreasing until k=11 where it raised before it decreased again. Based on this result the selected number of clusters is 11.

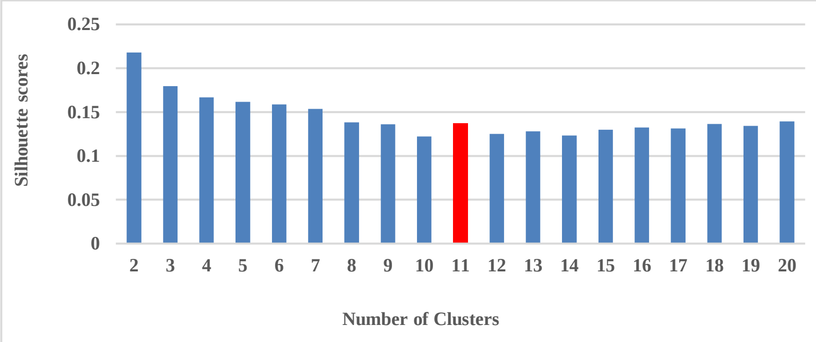

Figure 4. Silhouette scores of different numbers of clusters

For users clustering, we compared different clustering methods to adopt the better method. In comparison we applied k-means alone, ACO with k-means (ACO-KM), PSO with k-means (PSO-KM), and AFSA-KM. The comparison is based on SSE scores using (2) where the lowest score is the better. All methods used the same parameters such as, the number of clusters is 11, the number of dimensions is 18 (number of genres), the number of populations is 8, and the maximum number of iterations is 100. Table 1 shows the comparison results where AFSA-KM achieved the better result in term of SSE. It worth to mention that, using NIA along with k-means gives better results comparing with using k-means alone.

Table 1. SSE cores of different clustering methods

Algorithm	SSE
k-means	97487.79442
ACO-KM	95840.56913
PSO-KM	95840.47005
AFSA-KM	**95840.45985**

4.2. Model evaluation results

The experimental results of the proposed model OCB-DAE are computed by taking the average RMSE of all modeled clusters using (6). The proposed model uses 80% of dataset for training and 20% for testing. The input data is corrupted with dropout noise ratio 50% and the number of nodes in the hidden layer is 256. Sigmoid and linear activation functions are used in hidden and output layers respectively. Where Adam optimizer with 0.0001 learning rate is used to update the wights. The results of all modeled clusters are shown in Table 2. OCB-DAE model is compared with other models such as AE and DAE to evaluate its performance. The comparison of RMSE scores and the models' parameters are illustrated in Table 3. All the models in Table 3 share the same parameter settings, except in DAE and OCB-DAE models the dropout noise ratio is 0.5, for that reason they use a smaller regularization rate such as 0.001. OCB-DAE has a smaller number of nodes in the hidden layer (256) because it has a smaller number of nodes in the input layer as a result of the clustering process, for the same reason it uses a smaller batch size. The sparsity of the training data of AE and DAE models is 3.77%, while in OCB-DAE it varies in each cluster with average 3.37%.

Table 2. RMSE of each modeled cluster

Cluster	#Items (Samples)	#Users (Input nodes)	Sparsity (%)	RMSE
0	3,194	742	4.58	0.676
1	2,715	377	1.35	0.5033
2	3,268	652	5.07	0.748
3	2,483	361	1.71	0.495
4	2,386	375	1.5	0.5149
5	2,404	370	2	0.5834
6	2,281	326	1.87	0.4925
7	3,563	783	7.47	0.784
8	2,616	510	2.09	0.6034
9	2,839	542	3.01	0.6536
10	3,533	1,002	6.4	0.7441
		Total: 6,040	Average: 3.37	Average: 0.6180

Table 3. Performance compassion the proposed model and other models

Model	Batch size	#Hidden nodes	Regularization rate	RMSE
AE	256	500	0.01	1.0028
DAE	256	500	0.001	0.9129
OCB-DAE	128	256	0.001	**0.6180**

5. CONCLUSION

This research proposed an optimized clustering-based denoising autoencoder recommender system that utilized a nature-inspired algorithm such as artificial fish swarm algorithm to improve the k-means algorithm by determining the optimal initial centroids to divide the users into k clusters based on their similar interests. Each cluster's members will cooperate to extract the latent space of the users' rates by using denoising autoencoder model. Using AFSA with k-means showed a great improvement in clustering results. The proposed model was trained and evaluated with MovieLens 1M dataset where 80% of it is used for training and 20% for testing. RMSE the predicted and the actual test data was (0.618) which outperformed other models that use autoencoder and denoising autoencoder models without clustering.

REFERENCES

[1] A. A. Neamah and A. S. El-Ameer, "Design and evaluation of a course recommender system using content-based approach," in *ICOASE 2018 - International Conference on Advanced Science and Engineering*, Oct. 2018, pp. 1–6, doi: 10.1109/ICOASE.2018.8548789.

[2] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex and Intelligent Systems*, vol. 7, no. 1, pp. 439–457, Feb. 2021, doi: 10.1007/s40747-020-00212-w.

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, Jan. 2019, doi: 10.1145/3285029.

[4] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 430–450, Apr. 2020, doi: 10.1007/s11704-018-8052-6.

[5] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, New York, NY: Springer US, 2022, pp. 91–142.

[6] R. Rashidi, K. Khamforoosh, and A. Sheikhahmadi, "Proposing improved meta-heuristic algorithms for clustering and separating users in the recommender systems," *Electronic Commerce Research*, vol. 22, no. 2, pp. 623–648, Jun. 2022, doi: 10.1007/s10660-021-09478-9.

[7] H. Mittal and M. Saraswat, "CKGSA based fuzzy clustering method for image segmentation of RGB-D images," in *2018 11th International Conference on Contemporary Computing, IC3 2018*, Aug. 2018, pp. 1–6, doi: 10.1109/IC3.2018.8530568.

[8] R. Katarya and O. P. Verma, "Recommender system with grey wolf optimizer and FCM," *Neural Computing and Applications*, vol. 30, no. 5, pp. 1679–1687, Sep. 2018, doi: 10.1007/s00521-016-2817-3.

[9] S. Chourasia, H. Sharma, M. Singh, and J. C. Bansal, "Global and local neighborhood based particle swarm optimization," in *Advances in Intelligent Systems and Computing*, vol. 741, 2019, pp. 449–460.

[10] A. Al-Asadi and M. N. Jasim, "Clustering-based collaborative recommender system using a nature-inspiredalgorithm," in *2022 Iraqi International Conference on Communication and Information Technologies (IICCIT)*, Sep. 2023, pp. 95–100, doi: 10.1109/iiccit55816.2022.10010649.

[11] M. S. Kumar and J. Prabhu, "A hybrid model collaborative movie recommendation system using K-means clustering with ant colony optimisation," *International Journal of Internet Technology and Secured Transactions*, vol. 10, no. 3, pp. 337–354, 2020, doi: 10.1504/IJITST.2020.107079.

[12] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, p. 100134, Dec. 2021, doi: 10.1016/j.mlwa.2021.100134.

[13] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo, "Deep learning for wireless communications: An emerging interdisciplinary paradigm," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 133–139, Aug. 2020, doi: 10.1109/MWC.001.1900491.

[14] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning," *Education and Information Technologies*, vol. 25, no. 4, pp. 2635–2664, Jul. 2020, doi: 10.1007/s10639-019-10063-9.

[15] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709–2748, Apr. 2020, doi: 10.1007/s10462-019-09744-1.

[16] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *26th International World Wide Web Conference, WWW 2017*, Apr. 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[17] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 1339–1351, Mar. 2017, doi: 10.1016/j.eswa.2016.09.040.

[18] M. Alfarhood and J. Cheng, "Deep learning-based recommender systems," *Advances in Intelligent Systems and Computing*, vol. 1232, pp. 1–23, 2021, doi: 10.1007/978-981-15-6759-9_1.

[19] S. Lee and D. Kim, "Deep learning based recommender system using cross convolutional filters," *Information Sciences*, vol. 592, pp. 112–122, May 2022, doi: 10.1016/j.ins.2022.01.033.

[20] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, pp. 1309–1315, Feb. 2017, doi: 10.1609/aaai.v31i1.10747.

[21] Y. Pan, F. He, and H. Yu, "A novel enhanced collaborative Autoencoder with knowledge distillation for top-N recommender systems," *Neurocomputing*, vol. 332, pp. 137–148, Mar. 2019, doi: 10.1016/j.neucom.2018.12.025.

[22] X. Ran, X. Zhou, M. Lei, W. Tepsan, and W. Deng, "A novel K-means clustering algorithm with a noise algorithm for capturing urban hotspots," *Applied Sciences (Switzerland)*, vol. 11, no. 23, p. 11202, Nov. 2021, doi: 10.3390/app112311202.

[23] T. Puraram, P. Chaovalit, A. Peethong, P. Tiyanunti, S. Charoensiriwath, and W. Kimpan, "Thai food recommendation system using hybrid of particle swarm optimization and K-means algorithm," in *ACM International Conference Proceeding Series*, Apr. 2021, pp. 90–95, doi: 10.1145/3468891.3468904.

[24] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, no. C, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.
[25] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE Journal*, vol. 37, no. 2, pp. 233–243, Feb. 1991, doi: 10.1002/aic.690370209.
[26] M. Yu, T. Quan, Q. Peng, X. Yu, and L. Liu, “A model-based collaborate filtering algorithm based on stacked AutoEncoder,” *Neural Computing and Applications*, vol. 34, no. 4, pp. 2503–2511, Feb. 2022, doi: 10.1007/s00521-021-05933-8.
[27] D. Yi, J. Ahn, and S. Ji, “An effective optimization method for machine learning based on ADAM,” *Applied Sciences (Switzerland)*, vol. 10, no. 3, p. 1073, Feb. 2020, doi: 10.3390/app10031073.

BIOGRAPHIES OF AUTHORS

Ammar Abdulsalam Al-Asadi received the B.Sc. degree in computer science from the College of Science, Al-Mustansiriyah University, Baghdad, Iraq, the M.Sc. degree in computer science from the Informatics Institute for Postgraduate Studies/Commission for Computers and Informatics Baghdad, Iraq, and preparing to Ph.D. degree in machine learning from the Informatics Institute for Postgraduate Studies/Commission for Computers and Informatics Baghdad, Iraq. He works as senior programmer in Iraqi ministry of health, Department of planning and resource development, IT section. His research interests include recommender systems, machine learning, big data, and artificial intelligent. He can be contacted at email: phd202010553@iips.icci.edu.iq.

Mahdi Nsaif Jasim he works in University of Information Technology and Communications, College of Business Informatics, Dept. of Information Systems Management. Born in Babylon/Iraq, lives in Baghdad. His research Interests include information systems, data and information security, mining in vector data, machine learning, GIS, and database systems. The researcher has interrest in SDN data acquisition and data processing. He also Supervised 6 Ph.D. students in different Iraqi universities. He has been supervised 12 MSc. Students. He taught number os BSc. and MSc. courses a number of Iraqi universities. He has more than 20 years of software systems development in different trends starting from 1988 till now. He can be contacted at email: mahdinsaif@uoitc.edu.iq.