A NEURAL NETWORK-BASED COLLABORATIVE FILTERING MODEL FOR SOCIAL RECOMMENDATION SYSTEMS

Aadil Alshammari (1), Rakan Alanazi (2)*

(1) Department of Information Systems, Faculty of Computing and Information Technology, Northern Border University; (2) Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University
Saudi Arabia

* Corresponding Author, e-mail: aadil.al-shammari@nbu.edu.sa

Abstract: The exponential rise in the popularity of social networks in recent years has led to a corresponding surge in online data, creating a vast pool of information. This growing dataset has captivated the interest of researchers, compelling them to devise recommendation algorithms designed to navigate through this immense sea of data to establish meaningful friendships. Despite the success of CF, there is a notable gap in the related research, as deep learning's potential to generate friendship recommendations in social networks using CF has been neglected. This paper seeks to fill this gap by introducing a novel, collaborative friendship recommendation system. The proposed system leverages the power of Neural Collaborative Filtering (NCF) to offer users a platform for discovering new friends who are highly likely to engage interactively. To underscore the effectiveness of our model, we present experimental results using a real dataset, showcasing the system's capability to deliver accurate and valuable friend recommendations.

Keywords: neural networks, friend recommendation, collaborative filtering.

1. INTRODUCTION

Over the last two decades, there has been a significant surge in the number of individuals using the internet worldwide. In 2000, the user count stood at approximately 400 million, whereas, in 2016, it surpassed 3.4 billion [1]. This surge in internet users has led to an exponential growth of online data, particularly in the realm of social media [2], where most of the data is generated by users. As a result, locating appropriate information has become a daunting task. According to Statista, the global digital population consisted of five billion users in 2022, with 93% using social media [3]. Therefore, filtering algorithms are necessary to manage and analyze this vast amount of data.

Friendship recommendation algorithms grounded in collaborative and content-based filtering have been thoroughly investigated [4, 5]. However, these algorithms primarily emphasize recommending connections that have a high probability of acceptance but a low probability of becoming interactive. In our collected dataset, which is presented in detail below, we found that the average user engages with only 2.45% of their declared friends using comments and 13.23% using likes.

2. RELATED WORKS

2.1. Friendship recommendation systems/social networks

There are two popular friendship recommendation algorithms: CF and content-based algorithms. A content-based algorithm takes data from users' profiles and calculates the similarities between the users. The dependence on data from users' profiles is one of the disadvantages of such a system. For content-based algorithms to generate good recommendations, the data drawn from users' profiles must be accurate and up to date [6]. This is where a content-based algorithm struggles; these data are provided by the users themselves and may therefore be both inaccurate and outdated [6]. An advantage of the content-based filtering technique is that it does not face the cold start problem since it is dependent on users' profiles [6, 7].

A CF algorithm recommends friends by considering related users' behaviors. This approach has been widely used in well-known social networks [7, 8]. For example, CF has been used to recommend friends based on their locations and personalities [9, 10]. The popularity of social networks has yielded many proposed collaborative friendship algorithms. Studies [11-13] introduced various ways of using CF approaches such as, interaction-driven friending algorithms (IDF) and a clustering interaction-driven friending algorithm (CIDF), to generate interactive friendship connections.

The authors used k-means clustering and MF to recommend friends that were likely to interact. To create explainable recommendation engines, they introduced additional side information into MF [14-16]. In addition, CF has been employed to explore the strength of friendships on social networks with the aim of enhancing item recommendations. In [17], the authors leveraged friendship relations using trust quantification and recommendation models, emphasizing global trust network structures and shared interest preferences to improve item recommendations.

2.2. Deep learning-based collaborative filtering

In recent years, deep neural networks have exhibited tremendous success in different areas [18, 19], achieving state-of-the-art performance in computer vision [20], speech recognition [21], and natural language processing [22], as well as in the identification of fruitful recommendations using recommender systems. Recommendation tasks can be divided into three types based on the output: rating prediction, ranking prediction (top-n recommendation), and classification. The goal of rating prediction is to predict the explicit ratings that users would give to items. This task is important for scenarios where precise numerical feedback is available. Ranking prediction [23] provides users with a ranked list of n items. It focuses on predicting the relative order or preference of items for a user. It is crucial when the exact rating values are less important than the item's position in the recommendation list.

CF algorithms have been widely employed in recommendation systems. As a result of the ongoing research on and use of deep learning technology in different areas, CF based on deep learning has become a popular research topic. Many model-based CF approaches can be extended with neural networks, resulting in more flexible and scalable models capable of accelerated deep learning computations [26].

3. PROBLEM STATEMENT

In this section, we outline the top-N recommendation task under investigation in this paper. We assume that there are M users and N friends, denoted as $U = \{u_1, \ldots, u_m\}$ and $I = \{i_1, \ldots, i_n\}$, respectively. The interaction label between user u and friend i is denoted as y_{ui}, where "interaction" includes likes and/or comments. A user–friend connection (u, i) is considered interactive when $y_{ui} > 0$; otherwise, the connection is non-interactive. The value of y_{ui} indicates the degree of interactivity between u and i. The primary objective of top-N recommendation is to generate the list of friends who are most likely to engage interactively with their corresponding *target* users. A *target* user is a user for whom the model generates friendship recommendations.

4. PROPOSED APPROACH

CF is currently the most well-known approach to constructing recommendation systems, with recent studies increasingly incorporating deep neural network models to capture intricate user–item interactions [27]. One example is Neural Collaborative Filtering (NCF) [25]. In our research, we aimed to leverage the NCF model to make friends-of-friends (FoF) recommendations.

As illustrated in Figure 1, our approach uses the identity of user u and friend i as input. Through one-hot encoding, the input layer transforms these features into a binarized sparse vector composed of two feature vectors, v^U_u and v^I_i, representing user u and friend i. This sparse representation is then converted into dense vectors pu and qi via an embedding layer, where P and Q denote the latent factor matrices for users and friends, respectively.

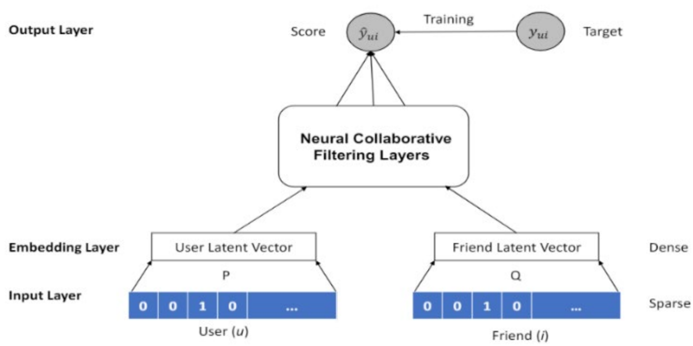

Figure 1. The Neural Collaborative Filtering Architecture

The user and friend embeddings are subsequently fed into the NCF layer to map the latent vectors. Following the approach proposed by He et al. [25], NCF can express and generalize MF through the use of element-wise products to combine users' and friends' representations, as expressed by the following equation:

$$\emptyset(p_u, q_i) = p_u \odot q_i \qquad (1)$$

Here, $\odot$ denotes the element-wise product of vectors. The resulting vector is projected onto the output layers, where $\hat{y}_{u_i}$ signifies the prediction scores, and y_{u_i} represents the interaction between user v^U_u and friend v^I_i. Following the approach in [28], we adopt binary cross-entropy loss as the objective function:

$$L = -\sum_{(u,i)\in Y^+ \cup Y^-} Y_{ui} log\hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \qquad (2)$$

Here, Y^+ denotes the set of observed interactions, and Y^- denotes the sampled unobserved interactions, i.e., negative instances. Our comprehensive model architecture and loss function contribute to the effective learning and prediction of interactive friendship recommendations, paving the way for further exploration and advancements in interactive friendship recommendation systems.

5.EXPERIMENTAL RESULTS

5.1. Dataset

For our research, we collected Facebook data in 2018 using our custom web crawler, ensuring diversity by randomly selecting seeds from various parts of the world. Our dataset included profiles of over 16,000 users, capturing friends' IDs, cities, interests, posts, and favourite media, alongside likes and comments. Focusing on interactive relationships, we collected IDs of users who liked or commented on friends' posts, resulting in over 15 million interactions. To protect privacy, we replaced users' IDs with random numeric IDs and only retrieved publicly available profiles. More details about the dataset in [12]. Interactions via likes and comments offer a more precise reflection of users' preferences than the information users declare about themselves in their profiles. Consequently, we prepared the dataset for our model as shown in Table 1:

Table 1: Our model's data input layout

User ID	*Friend ID*	*# Interaction*

The total number of interactions was determined by summing the likes and/or comments between the user and their friend.

5.2. Parameter settings

We implemented our proposed model using Keras and TensorFlow. The choice of hyper-parameters significantly influences the performance of recommendation systems. Therefore, we conducted thorough experiments to fine-tune these parameters. We randomly withheld 10% of the training set for each user as validation data and conducted parameter tuning on the data. Our chosen hyper-parameters had a batch size of 64 and a learning rate of 0.0001. We used the BinaryCrossentropy loss function to evaluate the model's performance and optimized our method with the Adam optimizer.

5.3. Handling negative samples

We randomly selected negative samples for each *target* user to ensure a balanced representation of interactions and non-interactions. For evaluation, we withheld five positive and five negative samples for each *target* user. The positive samples consisted of five interactive friends, while the negative samples comprised five users who had no prior interaction with the respective *target* user. These samples collectively form the test set, with the remaining samples constituting the training set.

5.4. Evaluation metric

To evaluate the performance of the recommendation system, we employed the hit ratio at rank k (HR@k) [29], a widely recognized method for assessing recommendation quality. HR@k assesses the presence of positive items within the top-K recommendations, offering valuable insights into the system's ability to accurately recommend interactive friends. Specifically, HR@k calculates the ratio of positive items recommended from the test set as expressed in the following equation:

$$HR@k = \frac{n}{k} \tag{3}$$

Here, n represents the number of positive items recommended from the test set, and K is the total number of recommended items. This evaluation metric provides valuable insights into the system's ability to accurately recommend interactive friends for the *target* user, enhancing our understanding of the recommendation system's performance.

5.5. Results and analysis

We tested our model with 5,638 *target* users. For each *target* user, we calculated the HR metric with a top-K of 5, resulting in an average HR of 67.55. The HR values show clear growth as K increases initially. Figure 2 indicates that our model achieves the best results with a top-K of 3, 4, or 5.

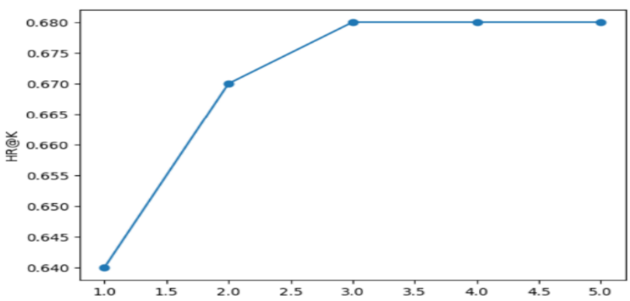

Figure 2. Evaluation of top-K item recommendation, where K ranges from 1 to 5

The performance starts decreasing when the top-K is greater than 5 because many users have only a few interactive friends. Consequently, setting aside users with more than five interactive friends for testing leaves insufficient interactive users for training. Additionally, our dataset is limited to publicly available friends, as many users keep their friend lists private. The results could have been even more successful with more publicly accessible data.

For brevity, Tables 2 to 5 present detailed test cases for only 4 out of the 5,638 cases to showcase the different HR results. The last row of each table contains information about the *target* user for whom friend recommendations were generated. Since the recommended friends were already friends of the *target* user, we have an actual history of interactions between them and the *target* user. The records of those friends are hidden and inaccessible to the model, as mentioned above.

Only the first column was generated by the model, and it represents friends' IDs prioritized as friendship recommendations for the *target* user. The highlighted friend_id is an already declared friend. We added the other four columns for analysis and to better explain the results. The second column shows the total number of interactions between the friend and the *target* user. The users with an interaction number of 0 had not had any interaction with the *target* user since they were not friends.

Table 2: Example of HR=0

Friend_id	***inter-action***	***Friend size***	***Activity %***
32232	*1*	*47*	*66.0*
211996	*0*	*134*	*40.0*
243828	*0*	*78*	*21*
13019	*0*	*324*	*18*
197894	*0*	*69*	*23*
330211	*1*	*64*	*16*
242730	*6*	*89*	*4*
389124	*0*	*94*	*3*
190477	*1*	*158*	*9*
6727	*1*	*518*	*16*
User_id	***66717***	***User_size***	***146***

Table 3: Example of HR=0.4

Friend_id	***inter-action***	***Friend size***	***Activity %***
30747	*0*	*71*	*65.0*
10643	*1*	*71*	*62.0*
535674	*0*	*17*	*35.0*
72162	*0*	*165*	*12.0*
452806	*2*	*120*	*29.0*
468129	*0*	*23*	*9.0*
378720	*2*	*67*	*7.0*
378733	*4*	*46*	*7.0*
378712	*2*	*66*	*11.0*
209442	*0*	*108*	*4.0*
User_id	***457471***	***User_size***	***76***

Table 4: Example of HR=0.6

Friend_id	***inter-action***	***Friend size***	***Activity %***
186662	*5*	*60*	*45.0*
186585	*4*	*74*	*46.0*
32324	*0*	*190*	*20.0*
283747	*2*	*58*	*26.0*
467957	*0*	*25*	*20.0*
282990	*3*	*76*	*33.0*
19648	*0*	*235*	*13.0*
19529	*0*	*213*	*12.0*
295896	*0*	*73*	*12.0*
283752	*8*	*54*	*7.0*
User_id	***283924***	***User_size***	***84***

Table 5: Example of HR=1

Friend_id	***inter-action***	***Friend size***	***Activity %***
726898	*4*	*19*	*68.0*
33083	*2*	*125*	*57.0*
426729	*2*	*57*	*68.0*
39664	*10*	*43*	*53.0*
2854	*3*	*178*	*33.0*
377806	*0*	*46*	*46.0*
709298	*0*	*28*	*7.0*
239850	*0*	*76*	*14.0*
248292	*0*	*96*	*0.0*
13496	*0*	*105*	*0.0*
User_id	***212071***	***User_size***	***213***

The second column represents the number of interactions between the friend i and the *target* user. For example, in Table 2, friend 242730 had 6 interactions with user 66717, Who is presented at the bottom of the table. The user_size is the size of the friend list of the *target* user. The last column, "activity %," represents in a percentage how many of the user's declared friends they had interacted with. This was calculated as follows:

$$acticity\% = \frac{i_{freinds}}{n}100 \tag{4}$$

where $i\_{friends}$ is the number of friends the user has interacted with, and n is the total number of friends they have listed.

As shown in Tables 2 to 5, we observed that the model performs best when the interactive friends have also interacted with other friends. The model grounds its recommendation decision on the established relationships between the friends who might be recommended and their friends. For instance, Table 2 shows that the model was only able to find 1 of 5 interactive friends, resulting in an HR of 0.2. In this case, the other four interactive users were not very interactive with their other friends. In other words, the model recommended more active people to the *target* user, even if they were not initially the user's friend. Other factors, such as the size of the user's friend list, also play a role in generating better recommendations. The *target* user 212071, shown in Table 5, had the largest friend list among the five test cases, and all their interactive friends also interacted with their friends. As a result, the model identified all the interactive friends and achieved an HR of 1. The non-interactive friends, such as those in Tables 2 and 3, should not have been friends with the respective *target* users in the first place. Instead, friends that the model recommended based on the negative samples actually had a higher probability of interacting with the *target* users, as their activity percentage was much higher.

We compared our approach with other interaction-driven friendship recommendation methods. The results of this comparison are presented in Table 6. All methods were tested using the same dataset described above. The table demonstrates the algorithm's effectiveness in suggesting relationships that lead to interactions. The results of the FoF algorithm were directly calculated from the collected dataset and exhibited the lowest HR rate. This is attributable to the algorithm's exclusive focus on suggesting mutual friends. Conversely, the other algorithms refine mutual friends to identify those most likely to engage in interactions. IDF accomplishes this by recommending "interaction-of-interaction" friends, yielding an achieved HR of 0.57. The CIDF algorithm, through the adoption of k-means clustering, was able to achieve 0.59 HR. Our proposed approach outperforms existing methods thanks to the use of NCF. On average, 68% of the recommendations generated by our approach were interactive friendship connections.

Table 6: Comparative Evaluation Results

Algorithm	***Approach***	***HR***
FoF	*Facebook Friends-of-Friends*	*0.26*
IDF [13]	*Interactive FoF*	*0.57*
CIDF [6]	*k-means clustering*	*0.59*
Our approach	*Neural Collaborative Filtering*	*0.68*

6. CONCLUSIONS

In this study, we explored a CF approach to top-N recommendation using a neural network model specifically designed to generate interactive friendship recommendations. The pre-existing friend connections in our dataset were likely suggested by Facebook's FoF friendship algorithm, a primary facilitator of friend connections. However, our findings reveal that our model outperforms state-of-the-art friendship algorithms, including Facebook's, yielding a significantly higher percentage of interactive friends. Looking ahead, we envision extending the NCF model to incorporate auxiliary information, such as users' geolocations (current city), interests, and social network structures. This expansion could enhance the representation of users and their friends. Additionally, our future plans involve exploring more sophisticated CF models and conducting comparative analyses to gain deeper insight into their performance.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the approval and support of the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, under Grant No. CSCR-2022- 11-1794.

REFERENCES

[1] M. Roser, H. Ritchie, E. Ortiz-Ospina. *''Internet,'' Our World in Data*, 2015, https://ourworldindata.org/internet.

[2] Radi Romansky. Digital age and personal data protection. *International Journal on Information Technologies and Security*, vol.14, no.3, 2022, pp. 89-100.

[3] Statista. *Global digital population as of April 2022*, 08 2022.

[4] M. D. Ekstrand, J. T. Riedl, J. A. Konstan et al. 'Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, vol. 4, no. 2, 2011, pp. 81–173.

[5] P. Lops, M. d. Gemmis, G. Semeraro. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, 2011, pp. 73–105.

[6] J. Bao, Y. Zheng, D. Wilkie, M. Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, vol. 19, 2015, pp. 525–565. https://doi.org/10.1007/s10707-014-0220-8

[7] H. Ko, S. Lee, Y. Park, A. Choi. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, vol. 11, no. 1, 2022, p. 141. 10.3390/electronics11010141

[8] A. Ansari, S. Essegaier, R. Kohli. Internet recommendation systems. *Journal of Marketing Research*, vol. 37, no. 3, 2000, pp. 363–375 [Online]. Available: https://doi.org/10.1509/jmkr.37.3.363.18779

[9] V. Mody, V. Mody, S. Parekh, K. Mistry. Friend.ly: A pilot study for a personality-based friend recommendation system. *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, 2022, pp. 1–8. 10.1109/I2CT54291.2022.9825012

[10] Y. Zheng, L. Zhang, Z. Ma, X. Xie, W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, 2011, pp. 1–44. 10.1145/1921591.1921596

[11] A. Alshammari, A. Rezgui. Better edges not bigger graphs: An interaction-driven friendship recommender algorithm for social networks. *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)*, 2020, pp. 1–8. doi: 10.1109/CloudTech49835.2020.9365918.

[12] A. Alshammari, A. Rezgui. Cidf: A clustering-based interaction driven friending algorithm for the next-generation social networks. *IEEE Access*, vol. 7, 2019, pp. 153 555–153 565.

[13] M. Alshammari, A. Alshammari. Friend recommendation engine for facebook users via collaborative filtering. *International Journal of Computers Communications and Control*, vol. 18, no. 2, 2023. https://doi.org/10.15837/ijccc.2023.2.4998

[14] B. Abdollahi, O. Nasraoui. Explainable matrix factorization for collaborative filtering. *Proceedings of the 25th International Conference Companion on World Wide Web, ser. WWW '16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee*, 2016, p. 5–6. [Online]. Available: https://doi.org/10.1145/2872518.2889405

[15] M. Alshammari, O. Nasraoui, S. Sanders. Mining semantic knowledge graphs to add explainability to black box recommender systems. *IEEE Access*, vol. 7, 2019, pp. 110 563–110 579. doi: 10.1109/ACCESS.2019.2934633

[16] M. Alshammari., O. Nasraoui., B. Abdollahi. A semantically aware explainable recommender system using asymmetric matrix factorization. *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR, INSTICC*. SciTePress, 2018, pp. 268–273. 10.5220/0006937902680273

[17] S. Zhang, N. Zhang, N. Li, Z. Xie, J. Gu, J. Li. Social recommendation based on quantified trust and user's primary preference space. *Applied Sciences*, vol. 12, no. 23, 2022, p. 12141. https://doi.org/10.3390/app122312141

[18] Jihane Ben Slimane. Efficient Real time Zynq 7000 FPGA deployment of optimized YOLOv2 deep leaning model for target detection, based on HDL Coder Methodology. *International Journal on Information Technologies and Security*, vol.16 , no.2, 2024, pp. 15-26. https://doi.org/10.59035/ZBTE3810

[19] Bandar Alshawi, Amr Munsh, Majid Alotaibi, Ryan Alturki, Nasser Allheeib. Classification of SPAM mail utilizing machine learning and deep learning techniques. *International Journal on Information Technologies and Security*, vol.16, no.2, 2024, pp. 71-82. https://doi.org/10.59035/FPKO7430

[20] K. He, X. Zhang, S. Ren, J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026– 1034. doi: 10.1109/ICCV.2015.123.

[21] Y. LeCun, Y. Bengio, G. Hinton et al. Deep learning. nature, 521 (7553), 436-444, Google Scholar Cross Ref Cross Ref, p. 25, 2015.

[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805, 2018. https://doi.org/10.48550/arXiv.1810.04805

[23] S. Khalid, S. Wu, A. Wahid, A. Alam, I. Ullah. An effective scholarly search by combining inverted indices and structured search with citation networks analysis. *IEEE Access*, vol. 9, 2021, pp. 120 210–120 226. doi: 10.1109/ACCESS.2021.3107939

[24] S. Zhang, L. Yao, A. Sun, Y. Tay. Deep learning-based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, vol. 52, no. 1, 2019, pp. 1–38. https://dl.acm.org/doi/10.1145/3285029

[25] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua. Neural collaborative filtering. *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[26] S. Singh, R. Rajan, S. Nandini, D. Ramesh, C. Prathibhamol. Friend recommendation system in a social network based on link prediction framework using deep neural network. in *2022 2nd International Conference on Intelligent Technologies (CONIT)*. IEEE, 2022, pp. 1–7. 10.1109/CONIT55038.2022.9848093

[27] T. Ma, Y. Shi. Aicf: Attentional item-interaction collaborative filtering. in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*. IEEE, 2022, pp. 146–149. doi: 10.1109/ICPECA53709.2022.9719163.

[28] G. He, D. Zhao, and L. Ding. Dual-embedding based neural collaborative filtering for recommender systems. *arXiv preprint arXiv:2102.02549*, 2021. https://doi.org/10.48550/arXiv.2102.02549

[29] S. Kabbur, X. Ning, G. Karypis. Fism: factored item similarity models for top-n recommender systems. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 659–667. https://doi.org/10.1145/2487575.2487589

Information about the authors:

Aadil Alshammari– received a Ph.D. in computer science from the New Mexico Institute of Mining and Technology in the United States. His research interests include recommender algorithms, AI, and data science.

Rakan Alanazi– received a Ph.D. in computer science from the University of Missouri - Kansas City in the United States. His research interests include software analytics, software engineering, machine learning, software visualization, and program comprehension.

Manuscript received on 16 May 2024