Big data cloud-based recommendation system using NLP techniques with machine and deep learning

Hoger K. Omar[1,2], Mondher Frikha[3], Alaa Khalil Jumaa[4]
[1]ENETCOM, University of Sfax, Sfax, Tunisia
[2]Department of Computer Science, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq
[3]Department of Electronics, National School of Electronics and Telecommunications of Sfax, University of Sfax, Sfax, Tunisia
[4]Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani 46001, Kurdistan Region, Iraq

Article Info

Article history:

Received Dec 11, 2022
Revised Mar 29, 2023
Accepted May 01, 2023

Keywords:

Artificial intelligence
Big data
Keras
Natural language processing
Recommendation system

ABSTRACT

Recommendation systems (RS) are crucial for social networking sites. Without it, finding precise products is harder. However, existing systems lack adequate efficiency, especially with big data. This paper presents a prototype cloud-based recommendation system for processing big data. The proposed work is implemented by utilizing the matrix factorization method with three approaches. In the first approach, singular value decomposition (SVD) is used, which is an old and traditional recommendation technique. The second recommendation approach is fine-tuned using the alternating least squares (ALS) algorithm with Apache Spark. Finally, the deep neural network (DNN) algorithm is utilized with TensorFlow. This study solves the challenge of handling large-scale datasets in the collaborative filtering (CF) technique after tuning the algorithms by adjusting the parameters in the second approach, which uses machine learning, as well as in the third approach, which uses deep learning. Furthermore, the results of these two approaches outperformed conventional techniques and achieved an acceptable computational time. The dataset size is about 1.5 GB and it is collected from the Goodreads website API. Moreover, the Hadoop distributed file system (HDFS) is used as cloud storage instead of the computer's local disk for handling larger dataset sizes in the future.

Corresponding Author:

Hoger K. Omar
ENETCOM, Universite of Sfax, Sfax, Tunisia
Email: omar.hoger@enetcom.u-sfax.tn

1. INTRODUCTION

Big data generally consist of many basic data and valuable knowledge can be excavated by expanding these data. Occasionally useful knowledge can be found even in error data so, the researchers can mine more valuable information from the big data [1]. The advancement of big data is resulting a huge redundancy problem that interfered with the process of knowledge obtaining. Over the past few years, big data has become increasingly prominent and its definition varies from one source to another. Some people refer to big data as the process of extracting, transforming, and loading massive amounts of data and others have different perspectives on its various attributes, including volume, variety, speed, veracity, variability, visualization, and value. The field of big data is constantly evolving, and the amount of data being generated is in the range of terabytes to zettabytes [2]. The recommendation system is known as the best solution for that problem since it recommends the product to the users according to their interests and hobbies [3]. The recommendation system is a subfield of natural language processing (NLP). NLP utilizes algorithmic methods rooted in statistical approaches or it applies machine learning algorithms to determine semantic meaning from text data [4]. The recommendation system has four filter types which are collaborative, content-based, demographic, and hybrid as shown in Figure 1. The collaborative filtering (CF) method is broadly applied to personalized recommendations. The CF works by gathering user feedback in the form of ratings for items. Then it exploits similarities in rating behavior amongst various users in finding how to recommend an item. It works on the principle that the user who has the same opinion in the past will have similar choices in the future as well [5].

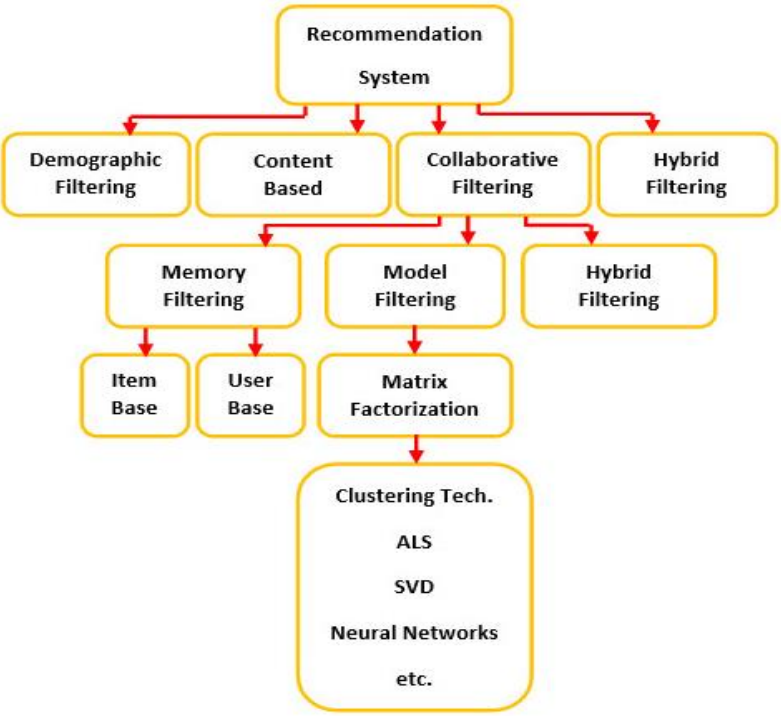

Figure 1. Recommendation system methods

Matrix factorization (MF) is a powerful method for finding hidden information inside the data. MF is characterized by both products and users by vectors of factors derived from product rating forms. The high correspondence between user factors and product factors conducts the recommendation [6].

Singular value decomposition (SVD) is well known MF example. It is used for recognizing latent factors in the area of Information retrieval to treat collaborative filtering problems. In the recommendation system, the matrix of user-item can be decomposed to the matrix of low dimensional through SVD [7]. The main disadvantage of this method is that the process of model building is computationally expensive as well as, the volume of memory usage is extremely intensive. In addition, SVD does not reduce the problem of cold start [8].

Therefore, finding alternative methods is highly recommended especially, methods that tackle with big data. Hence, the proposed MF model is constructed in this work three times to compare modern methods such as alternating least squares (ALS) and deep neural network (DNN) with traditional methods such as SVD. So firstly, the SVD is used to check how the traditional method deal with the big data. Secondly, the ALS algorithm has been used which is one of the algorithms inside the machine learning package of the Apache Spark big data tool. Finally, a deep neural network algorithm is utilized by operating the Keras framework on top of TensorFlow.

The justification behind operating DNN that it is works perfectly when a massive of complexities are exists or when there are huge amounts of training cases [9]. Also, the justification behind operating ALS is that it has a practical method for dealing with implicit data that is commonly non-sparse. Besides, ALS is a more effective optimization technique and quite easy to parallelize [10].

The big dataset is previously collected from Goodreads social networks website which is the world’s largest site for readers and book recommendations. Hadoop distributed file system cloud storage is employed to handle the utilized big dataset. Also, the proposed cloud storage system was designed to handle bigger datasets for future work.

This article is structured as: in section two the related work to this article is provided. Section three describes the system algorithms and tools, section four describes the proposed system architecture precisely. Section five shows the results and finally, the conclusion is presented in section six.

2. RELATED WORK

A tremendous number of articles are published on the topic of recommendation system recently using machine learning and deep learning. Essentially, there have been several approaches to building an effective system. In this part, the concentration will be on credible works in this field. Liu *et al.* [11] proposed explicit-implicit feedback based on the algorithm of neural matrix factorization. They discover modern loss function depending on direct and indirect feedback with neural networks for predicting the user's preference. Zhang *et al.* [12] explored a framework that combined collaborative filtering with deep learning. They separate the framework into two sections the first one utilizes the feature representation technique according to the quadric polynomial regression. In section two, the latent features are employed to be an input for the neural network to estimate the ratings. Yanes *et al.* [13] suggested a recommendation system for expecting the suitable actions that can be offered by college staff to improve the quality of courses they teach and consequently the complete educational program. The recommendation process was according to the specifications of the courses, academic archives, and course learning evaluations. They tested five important algorithms of machine learning for expecting suitable actions however, four approaches are categorized as problem transformation techniques. Zhang *et al.* [14] proposed topical attention matrix factorization with the probability method using a social network dataset. The work consists of three learning phases and performs a good result in treating the cold start method. Moreover, they found that the ratings and comments are time-sensitive which means old comments might become noise data for recommendations. Awan *et al.* [15] applied a movie recommender system according to a collaborative filtering method utilizing the ALS algorithm inside Spark to anticipate the rated movies. In their implementation, the last search data of a user regarding movies have been used to train the recommendation system and find the list of forecasts for top ratings. The work utilized a model-based method of matrix factorization and solved many problems of that method. Prasetyaningrum *et al.* [16] present a method for making decisions based on multiple criteria, which incorporates feedback from social media. They merge sentiment analysis with the analytical hierarchy process (AHP), allowing for the integration of user and public opinion in the decision-making process. This approach aims to provide users with optimal recommendations by combining AHP calculations with criteria obtained from social media.

This study employs the capabilities of NLP with both machine learning and deep learning to build a prototype recommendation system that handles and processes big data. However, Hadoop cloud storage is used instead of the computer's local disk for handling tremendous size of data. The constructed systems based on a collaborative filtering method showed their effectiveness in both ALS and DNN models.

3. SYSTEM ALGORITHMS AND TOOLS

The study mainly consists of two fundamental models that play a significant role in the field of artificial intelligence: the machine learning model that employs the ALS algorithm and the deep learning model that utilizes the DNN algorithm. These models employ different methods to learn patterns from data, which makes them suitable for various applications. The following subsection illustrates both models separately and demonstrates all the utilized tools.

3.1. Alternating least squares with Spark

ALS algorithm offers an expert technique for dimensionality reduction in the collaborative filtering method. Recently ALS is used with the latest big data tool Apache Spark MLlib because it can handle the complex computations of ALS [17]. Spark is an open-source framework that processes the data inside the RAM. It permits the fast processing of massive data with the capabilities of parallel data processing over distributed nodes. However, multi-threaded lightweight processes can run on Spark inside Java virtual machine (JVM). Spark can upload and download the data from Apache Hadoop by accessing Hadoop distributed file system (HDFS) since it works on top of the existing Hadoop cluster [18]. The management of several operations is quite simple with Apache Spark by providing a data pipeline method. Also, the characteristics of Spark are appropriate from the bottom-up for treating big data and it is much faster than other big data tools such as Hadoop. Besides, it supports many programming languages such as Java, Scala, Python, and R [19]. Fortunately, the Spark machine learning library consists of an implementation of the ALS algorithm for building a model in the form of collaborative filtering [20]. However, Spark MLlib is a well-known open-source machine learning library that operates on large datasets and uses automatic data parallelization. It supports a wide range of machine learning tasks, including regression, dimensionality reduction, clustering, classification, and feature extraction [21].

3.2. Deep neural networks with Keras

DNN algorithm recently have been discovered to be efficient in various fields, starting from computer vision, face recognition, to natural language processing. Besides, there are fairly few articles on operating DNN for recommendation systems and showing astonishing results [22]. The structures of deep neural networks outperform the other machine learning algorithms, especially in the recommendation topic. However, it provides better accuracy with further feature abstraction, and also offers the best ability of learning with complex data [23]. One of the most famous used packages for functioning DNN is Keras.

Keras is a well-known framework that provides uncomplicated application programming interfaces. It is one of the most utilized deep learning models between top-five winning groups on Kaggle. Keras is written in Python and used by many scientific organizations around the globe such as NASA due to its quick model training. Furthermore, it takes the advantage of TensorFlow's deployment abilities [24], [25]. TensorFlow is an open-source platform founded by Google and it can be run on the different operating system and support both central processing unit (CPU) and graphics processing unit (GPU). Also, it is a proven software to generate models and productionize deep neural learning according to the data-flow charts [26].

4. PROPOSED SYSTEM ARCHITECTURE

In this section, the proposed techniques for building a recommendation system using three approaches which are SVD, ALS using Spark, and DNN using TensorFlow deep learning library are presented. The proposed framework is used the same big dataset three times and each time with one of the utilized approaches separately to test the most accurate approach. The overall framework of the proposed system architecture and system steps for all three approaches are shown in Figure 2 and Figure 3 respectively. As can be seen, the framework steps consist of eight stages:

a) Collecting and aggregating the big dataset from Goodreads API which allows developers access to Goodreads data.
b) In this step, to discover the power of the utilized approaches with any type of data even heterogeneous data, only a few pre-processing steps have been used which are feature selection and changing the rating text into a numeric value by a few steps of natural language processing techniques for being accepted by the algorithms. Hence, the numeric rating is scaled from 0 to 5 where (0 = no rating) and (5 = highest rating).
c) In this step, the collected dataset has been uploaded to the Hadoop distributed file system (HDFS) and transformed to be processed by SVD, ALS, and DNN algorithms separately which means the data fed to the algorithms from the cloud storage instead of the PC local disk.
d) In this step, three recommendation models (approaches) have been built. The first model employed the SVD algorithm. Also, for the second model, the ALS algorithm of the Apache Spark machine learning library is utilized. The other model employed the DNN algorithm by using the Keras library on top of TensorFlow.
e) In this step, building three matrix factorization models, each model representing one of the utilized approaches.
f) Gaining three lists of the recommended books from the three utilized approaches.
g) Evaluating each approach list individually.
h) Comparison between the results of the approaches based on some measures that are appropriate with the recommendation system as well as to the time performance comparison.

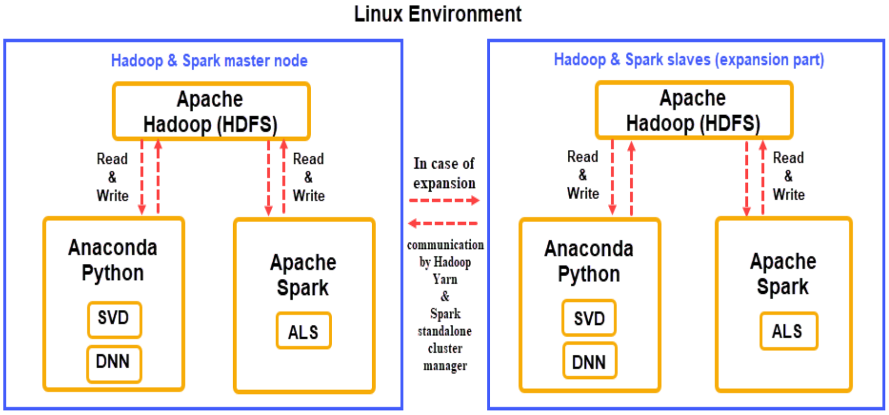

Figure 2. The proposed recommendation system architecture

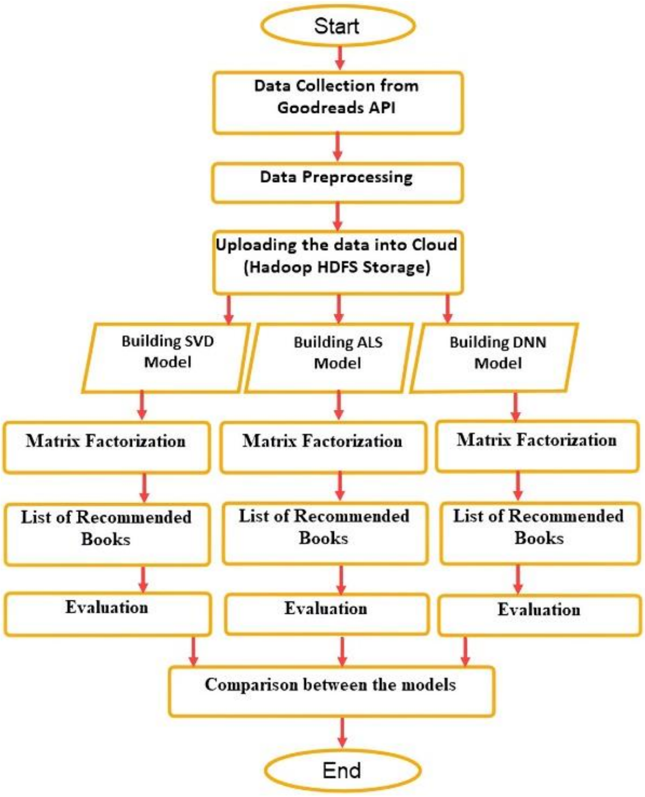
Start
Data Collection from Goodreads API
Data Preprocessing
Uploading the data into Cloud (Hadoop HDFS Storage)
Building SVD Model
Building ALS Model
Building DNN Model
Matrix Factorization
Matrix Factorization
Matrix Factorization
List of Recommended Books
List of Recommended Books
List of Recommended Books
Evaluation
Evaluation
Evaluation
Comparison between the models
End

Figure 3. The proposed system steps

5. EXPERIMENTAL AND RESULTS

In the following subsections, the details of the experiments and results is shown. An experimental setup is presented in subsection 5.1, dataset description is given in subsection 5.2. Finally, results and analysis are introduced in subsection 5.3.

5.1. Experimental setup

All the experiments were carried out on the Linux-Debian 9 64-bits operating system. To implement the ALS algorithms firstly, Apache Spark should be installed with all its dependencies such as Scala, and JDK. Similarly, to operate a DNN initially, TensorFlow and Keras frameworks have been installed. The rest information about the tested environment is shown in Table 1.

Table 1. Tested environment

No.	Resource type	Details
1	Host O.S	Windows 10, 64-bits
2	Guest O.S	Debian 9, 64-bits
3	VMware version	15.0.2 build-10952284
4	Computer CPU	Intel® core™ i7-8850H CPU @ 2.60GHz 2.59 GHz
5	VMware RAM size	30 GB
6	VMware hard disk size	120 GB
7	Type of hard disk drive	SSD
8	Spark version	Spark-3.2.0-bin-hadoop2.7
9	Tensor flow version:	2.3.0
10	Keras version:	2.4.0
11	Pandas	1.2.4
12	Python version	The latest release of Anaconda with Python 3.8.8
13	Hadoop version	Hadoop 2.8.5 with 128 block size

5.2. Datasets description

To validate the proposed techniques, two real datasets have been used in this work. The datasets were taken from Goodreads social site API which presents the information’s on the books [27]. It consists of two

CSV parts the first part is about the books, author, publisher, and ratings. and the second dataset part is about the book ratings by the users and the rating presents as a short text in the dataset file which means it needs many natural language processing techniques for acquiring a good result. The size of the datasets is about 1.5 GB and have more than 2 million user ratings. Table 2 demonstrates additional details on the datasets.

Table 2. Characteristics of the datasets

No.	Name of the dataset	Size	No. of fields (attributes)	No. of rows
1.	Book.csv	1.49 GB	18	1795474
2.	Rating.csv	28 MB	3	362602
3.	Total	1.5 GB	21	2158076

5.3. Results and analysis

As mentioned before, the proposed work is divided into three approaches using three types of algorithms. The utilized algorithms are singular value decomposition in approach one, alternating least squares in approach two, and deep neural networks in approach three. For each approach, the measures of root mean squared error (RMSE), mean absolute error (MAE), and execution time have been computed. For calculating these measures, it should cut a sub-matrix of several dimensions from several parts of the matrix and then compute the scores to find out how well the recommender system is performed partly and entirely. The following subsections show all three approaches as well as a brief comparison between them.

5.3.1. Approach one

The SVD algorithm is used in this approach and the datasets are randomly divided into 80% train data and 20% test data. A few data preprocessing steps have been done for applying SVD using the matrix factorization technique. The outcome obtained a low proportion of both RMSE and MAE measures with a long-time performance as shown in Figure 4, Figure 5, and Figure 6. Besides, this algorithm can not deal with the of problem cold start.

5.3.2. Approach two

In this approach, the recommendation system was built on matrix factorization and the ALS algorithm by utilizing the Apache Spark machine learning library (MLlib). Python programing language has been used which already exists in the Spark API under the name of Pyspark. The obtained result is much better than the SVD. However, the ALS records a smaller time performance than the SVD and DNN approaches.

5.3.3. Approach three

In the final approach, the recommendation system was built on matrix factorization and the DNN algorithm. The proposed DNN approach has been executed using the Keras framework on top of the TensorFlow library. The datasets are randomly divided into 80% train data and 20% test data. After many examinations, the model shows its best tunning in the epochs of 25, with 64 batch-size. The acquired result scored the best ratio among the utilized approaches which are 0.67 RMSE and 0.56 MAE which means more than 75% of accuracy. In addition, it recorded a slightly bigger time performance than the ALS approach and a smaller time performance compared to the SVD.

5.3.4. Comparison between the approaches

After performing all the approaches, now presenting a few comparisons among them are compulsory to determine the best approach. The comparisons concentrate on the time performance and also on the other acquired measures such as RMSE and MAE. It can be concluded that the SVD algorithm produced a negative impact on big data because it needs a high computational time for model building due to its structure. In addition, the SVD is memory (RAM) consuming and also provides a very low accuracy rate compared to ALS and DNN approaches. Likewise, the SVD suffers from a cold-start problem, which describes the trouble of making recommendations when the users or the items are new, which remains a great challenge for the SVD in collaborative filtering. On the other hand, the measures for both ALS and DNN approaches are very close to each other. The time performance of the DNN approach is slightly bigger than the ALS. Likewise, these two approaches can recommend three books out of four and outperformers compared to the SVD approach especially if the dataset is huge. The results of all approaches have been shown in Figure 4, Figure 5, and Figure 6.

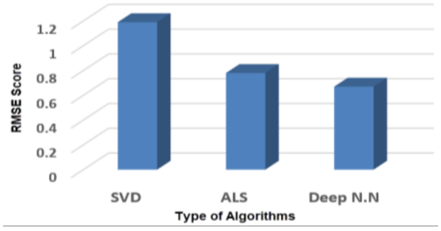

Figure 4. RMSE results

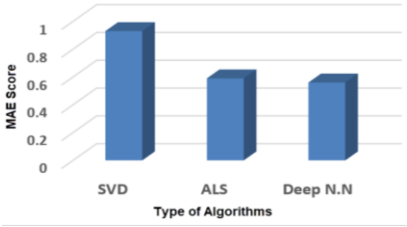

Figure 5. MAE results

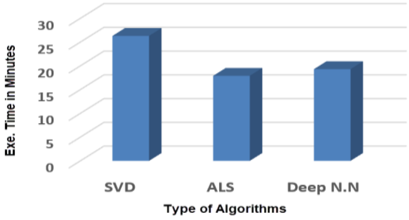

Figure 6. Execution time results

6. CONCLUSION

In this article, three approaches of the matrix factorization method have been tested to find out an accurate big data recommendation system among them and then recommend the relevant type of books to the reader. The first approach is SVD which used in this work just for comparing the efficiency of this traditional method with the modern methods in treating big data. For the second approach, the ALS algorithm is utilized within the machine learning package of Apache Spark 3.2.0. Finally, operating the capabilities of the DNN algorithm utilizing the Keras framework on top of TensorFlow. The datasets consist of two files, the first one is consisting of information about the books, and the second file consisting a user rating as a short natural language text with a size of 1.5 GB. Moreover, Hadoop HDFS cloud storage is employed to handle the utilized big dataset instead of the local disc. Besides, the proposed cloud storage system was designed to handle bigger datasets for the future. The study tuned the architecture of the ALS and DNN algorithms and presents its effectiveness with big data for collaborative filtering techniques. The results of approach one (SVD) show that conventional techniques cannot deal efficiently with big data and it has a problem of cold start. On the other hand, the results of the other approaches (ALS and DNN) show that they can recommend about 3 out of 4 books correctly to the readers with acceptable computational time and they have outperformed the conventional techniques. Future work will concentrate on gaining better results by adding more NLP techniques and also by employing optimization techniques. In addition, using parallel data processing (multi-nodes) for recommending a tremendous size of data.

REFERENCES

[1] P. Sun, “Music Individualization Recommendation System Based on Big Data Analysis,” *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/7646000.

[2] B. Nirmala, R. Abueid, and M. A. Ahmed, “Big Data Distributed Support Vector Machine,” *Mesopotamian journal of Big Data*, vol. 2022, 2022, doi: 10.58496/MJBD/2022/002.

[3] S. Bin and G. Sun, “Matrix Factorization Recommendation Algorithm Based on Multiple Social Relationships,” *Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/6610645.

[4] W. Leeson, A. Resnick, D. Alexander, and J. Rovers, “Natural Language Processing (NLP) in Qualitative Public Health Research: A Proof of Concept Study,” *International Journal of Qualitative Methods*, 2019, doi: 10.1177/1609406919887021.

[5] I. A. A. Q. Al-Hadi, N. M. S. M. N. Sulaiman, and N. Mustapha, “Review of the temporal recommendation system with matrix factorization,” *International Journal of Innovative Computing, Information and Control*, vol. 13, no. 5, pp. 1579-1594, 2017. [Online]. Available: http://www.ijicic.org/ijicic-130511.pdf

[6] A. K. Sahoo, C. Pradhan, R. K. Barik, and H. Dubey, “DeepReco: Deep Learning Based Health Recommender System Using Collaborative Filtering,” *Computation*, vol. 7, no. 2, 2019, doi: 10.3390/computation7020025.

[7] A. M. A. Al-Sabaawi, H. Karacan, and Y. E. Yenice, “Exploiting implicit social relationships via dimension reduction to improve recommendation system performance,” *PLOS ONE*, 2020, doi: 10.1371/journal.pone.0231457.

[8] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261-273, 2015, doi: 10.1016/j.eij.2015.06.005.

[9] 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, 2019, doi: 10.1145/3285029.

[10] J. -B. Li, S. -Y. Lin, Y. -H. Hsu, and Y. -C. Huang, “An empirical study of alternating least squares collaborative filtering recommendation for Movielens on Apache Hadoop and Spark,” *International Journal of Grid and Utility Computing*, vol. 11, no. 5, pp. 674-682, 2020, doi: 10.1504/IJGUC.2020.110053.

[11] H. Liu, W. Wang, Y. Zhang, R. Gu, and Y. Hao, “Neural Matrix Factorization Recommendation for User Preference Prediction Based on Explicit and Implicit Feedback,” *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/9593957.

[12] L. Zhang, T. Luo, F. Zhang, and Y. Wu, “A Recommendation Model Based on Deep Neural Network,” *in IEEE Access*, vol. 6, pp. 9454-9463, 2018, doi: 10.1109/ACCESS.2018.2789866.

[13] N. Yanes, A. M. Mostafa, M. Ezz, and S. N. Almuayqil, “A Machine Learning-Based Recommender System for Improving Students Learning Experiences,” *IEEE Access*, vol. 8, pp. 201218-201235, 2020, doi: 10.1109/ACCESS.2020.3036336.

[14] W. Zhang, F. Liu, D. Xu, and L. Jiang, “Recommendation system in social networks with topical attention and probabilistic matrix factorization,” *PLOS ONE*, 2019, doi: 10.1371/journal.pone.0223967.

[15] M. J. Awan *et al.*, “A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach,” *Electronics*, vol. 10, no. 10, 2021, doi: 10.3390/electronics10101215.

[16] I. Prasetyaningrum, K. Fathoni, and T. T. J. Priyantoro, “Application of recommendation system with AHP method and sentiment analysis,” *Telecommunication, Computing, Electronics and Control (TELKOMNIKA)*, vol. 18, no. 3, pp. 1343-1353, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14778.

[17] Y. Niu, “Collaborative Filtering-Based Music Recommendation in Spark Architecture,” *Mathematical Problems in Engineering*, vol. 2022, 2022, doi: 10.1155/2022/9050872.

[18] H. K. Omar and A. K. Jumaa, “Big Data Analysis Using Apache Spark MLlib and Hadoop HDFS with Scala and Java,” *Kurdistan Journal of Applied Research (KJAR)*, vol. 4, no. 1, pp. 7-14, 2019, doi: 10.24017/science.2019.1.2.

[19] H. K. Omar and A. K. Jumaa, “Distributed big data analysis using Spark parallel data processing,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1505-1515, 2022, doi: 10.11591/eei.v11i3.3187.

[20] M. Winlaw, M. B. Hynes, A. Caterini, and H. D. Sterck, “Algorithmic Acceleration of Parallel ALS for Collaborative Filtering: Speeding up Distributed Big Data Recommendation in Spark,” *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, 2015, pp. 682-691, doi: 10.1109/ICPADS.2015.91.

[21] Z. Hasan, H. -J. Xing, and M. I. Magray, “Big Data Machine Learning Using Apache Spark Mllib,” *Mesopotamian Journal of Big Data*, vol. 2022, 2022, doi: 10.58496/MJBD/2022/001.

[22] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. -S. Chua, “Neural Collaborative Filtering,” in *WWW ‘17: Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[23] N. Liang, H. -T. Zheng, J. -Y. Chen, A. K. Sangaiah, and C. -Z. Zhao, “TRSDL: Tag-Aware Recommender System Based on Deep Learning–Intelligent Computing Systems,” *applied sciences*, vol. 8, no 5, 2018, doi: 10.3390/app8050799.

[24] M. Kula *et al.*, “keras.” keras.io. https://keras.io/. (accessed Sep. 17, 2022).

[25] J. Bobadilla, A. G. -Prieto, F. Ortega, and R. L. -Cabrera, “Deep learning approach to obtain collaborative filtering neighborhoods,” *Neural Computing and Applications*, vol. 34, pp. 2939–2951, 2022, doi: 10.1007/s00521-021-06493-7.

[26] N. Chockwanich and V. Visoottiviseth, “Intrusion Detection by Deep Learning with TensorFlow,” *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019, pp. 654-659, doi: 10.23919/ICACT.2019.8701969.

[27] *Goodreads*, goodreads.com, Jun. 8, 2022. [Online]. Available: https://www.goodreads.com/

BIOGRAPHIES OF AUTHORS

Hoger K. Omar is currently an instructor at the University of Kirkuk and head of the Lab section in the quality assurance Department/presidency of Kirkuk University. His research interests include big data analysis, data mining, web mining, text classification, machine learning, operating systems, distributed systems with Hadoop, Recommendation system, and NLP. He received a bachelor’s degree in Computer Science from the University of Kirkuk/College of Science, Kirkuk, Iraq in 2008 and a Master’s degree in Information Technology from SPU University, Sulaimaniyah, Iraq in 2019. He can be contacted at email: hogeromar@uokirkuk.edu.iq.

Mondher Frikha is currently a full professor at the National School of Electronics and Telecommunications, University of Sfax, Tunisia. He is also a director of the ‘Advanced Technologies of Image and Signal Processing’ research lab. His research interests include digital signal and image processing, Speech and audio processing, pattern recognition, and IA applications. He received the master of applied sciences in electrical engineering from the University of Ottawa Canada in 1991. He then worked as a head project at the Industrial Land Agency in Tunisia. In 2003, he started pursuing his graduate research and obtained 2007 his Ph.D. degree from the National School of Engineering of Sfax, Tunisia. He can be contacted at email: mondher.frikha@enetcom.usf.tn.

Alaa Khalil Jumaa currently is a director of the Scientific Affairs and Postgraduate Studies Unit at the Technical College of Informatic, SPU University, Iraq. He obtained his BSc (1997) and MSc (2004) degrees in Computer Engineering from the University of Technology, Baghdad, Iraq. He received his Ph.D. in Database and Data Mining Techniques from the University of Sulaimani, Kurdistan Region, Iraq in 2013. His research interests include database techniques, PPDM, and big data analysis. He can be contacted at email: alaa.alhadithy@spu.edu.iq.