Recommendation System for E-commerce using Alternating Least Squares (ALS) on Apache Spark

Subasish Gosh[1][0000−0002−9699−5684], Nazmun Nahar[2][0000−0002−4279−0878], Mohammad Abdul Wahab[3][0000−0003−3103−0878], Munmun Biswas[4][0000−0001−8370−0878−8773], Mohammad Shahadat Hossain[5][0000−0002−3090−7645], and Karl Andersson[6][0000−0003−0244−3561]

[1] BGC Trust University Bangladesh Bidyanagar, Chandanaish, Bangladesh
subhasish@bgctub.ac.bd
[2] BGC Trust University Bangladesh Bidyanagar, Chandanaish, Bangladesh
nazmun@bgctub.ac.bd
[3] BGC Trust University Bangladesh Bidyanagar, Chandanaish, Bangladesh
wahab@bgctub.ac.bd
[4] BGC Trust University Bangladesh Bidyanagar, Chandanaish, Bangladesh
munmun@bgctub.ac.bd
[5] University of Chittagong,Chittagong, Bangladesh
hossain_ms@cu.ac.bd
[6] Lulea University of Technology, SE-931 87 Skellefteå, Sweden
Karl.andersson@ltu.se

Abstract. Recommendation system can predict the ratings of users to items by leveraging machine learning algorithms. The use of recommendation systems is common in e-commerce websites now-a-days. Since enormous amounts of data including users' click streams, purchase history, demographics, social networking comments and user-item ratings are stored in e-commerce systems databases, the volume of the data is getting bigger at high speed, and the data is sparse. However, the recommendations and predictions must be made in real time, enabling to bring enormous benefits to human beings. Apache spark is well suited for applications which require high speed query of data, transformation and analytics results. Therefore, the recommendation system developed in this research is implemented on Apache Spark. Also, the matrix factorization using Alternating Least Squares (ALS) algorithm which is a type of collaborative filtering is used to solve overfitting issues in sparse data and increases prediction accuracy. The overfitting problem arises in the data as the user-item rating matrix is sparse. In this research a recommendation system for e-commerce using alternating least squares (ALS) matrix factorization method on Apache Spark MLlib is developed. The research shows that the RMSE value is significantly reduced using ALS matrix factorization method and the RMSE is 0.870. Consequently, it is shown that the ALS algorithm is suitable for training explicit feedback data set where users provide ratings for items.

Keywords: Recommendation System · Alternating Least Square (ALS) · Matrix Factorization · Spark MLlib · E-commerce .

1 Introduction

In a homogenous market similar types of products are usually found, while in heterogeneous markets different types of product found. Therefore it is necessary for the migration of homogenous markets to heterogeneous markets, enabling us to find all types of product in one place [1]. Pine claims that the design of one product is simply no longer sufficient. Organizations must at least be able to develop multiple goods that satisfy different customer needs. The e-commerce revolution helps companies to offer more products to consumers.

E-commerce is a process whereby goods, services and information are purchased and sold or exchanged through computer networks, including the Internet [2]. Nevertheless, this e-commerce generates vast amounts of information that customers need to process before selecting which goods to purchase. The use of the Recommendation System is the solution to this problem of overloading information. The recommendation system is used to propose products for its customers through e-commerce sites. The products may be recommended based on the online retailer, based on the demographics of the buyer, or based on an analysis of the customers past buying behavior in order to anticipate future behavior. Such strategies are usually a part of a website's personalization, since they allow any customer to follow a platform. Recommendation system automates web personalization, which enables individual customization to this extent is one way of realizing the idea of pines on the web. So Pine is presumably in agreement with JeffBezos, CEO of Amazon.com; "If I have two million online customers, I should have two million web stores".

The Recommendation System employs a range of technologies to filter the best result and to provide users with information they need. Recommendation System is divided into three broad categories: first one is collaborative filtering system, next one is con-tent based system and the last one is hybrid recommendation system [3]. Content-based systems attempt to check the actions of the item as suggested. This operates by learning new user's behavior, based on their knowledge needed, which is provided in objects by the user. It is a keyword-specific Recommendation System which uses keywords to describe the items.

Therefore, models work in a content-based Recommendation System so that they can suggest comparable user's products which have been enjoyed or are currently browsing. The collaborative filtering system is focused on similarities between customers' need and the items. The suggested items for a new user are the ones that other similar people in the past browsing history liked. In collaborative filtering, an item is combined, similarities are recognized based on user ratings and new recommendation is created based on multi-user comparisons. It

faces however many difficulties as well as limitations, such as data sparsity which is responsible for the evaluation of large items. It is also difficult to predict on the basis of the nearest neighboring algorithm. Scalability, when increasing the number of users and the number of items and the last thing is cold starts, where bad relationships between people with the same mind. Hybrid Recommendation System carries out their tasks by taking account of the combination of content-based and collaborative filtering approaches so that a particular item is suitable. The hybrid system is considered by many organizations to be the most widely used Recommendation System because it is able to remove any defect that could have arisen in the implementation of a Recommendation System.

The Main objective of this paper is to develop a recommendation system. Here for developing a recommendation system a computationally effective algorithm for matrix factorization ALS (Alternating Least Square) is proposed. ALS is one kind of collaborative filtering method is used to solve the overfitting issue in sparse data [4]. Apache Spark is used to implement the ALS algorithm. In this paper ALS algorithm is also compared with other collaborative filtering method which is SGD (Stochastic Gradient Descent) and SVD (Singular Value Decomposition).

In the first section of this research paper,. In the second section of this research paper, ALS Matrix Factorization Method is explained. In the third section of this research paper, predictive accuracy metrics used to evaluate the system is defined. In the fol-lowing section, system prototype along with system workflow is described with a diagram. In section VI, system architecture of the recommendation system is illustrated. Experimental Result is described in section VII and finally Conclusion and Future work of the system is discussed.

2 Literature Review

Many researchers have worked in the field of e-commerce utilizing techniques of big data over with machine learning and data mining methods. A number of applications have developed using those techniques. Recommendation system is used in large areas such as e-commerce, social networks, education, government etc. The development of recommendation system using machine learning methods of previous re-searches is described in this section.

A Hybrid Recommendation model was designed by Wei et al. [5].This study solved the issue of Complete Cold start (CCS) and Incomplete Cold Start (ICS) problem. Those remaining problem was overcome by using combined feature of Collaborative filtering(CF)and deep learning neural network. A Specific Deep learning neural net-work (SADE) used for extracting content features of items. CF model and time SVD++ used to predict the rating by taking content features of cold start items. Net-flix large movie dataset used to perform their analysis. This research found that CCS can able to do good recommendation than ICS.

But their experiment done by investi-gating less parameter and evaluated their performance by using RMSE rating predic-tion which is not efficient for real recommendation system.

Using collaborative filtering algorithm on the basis of item a recommendation system was developed by Kupisz and Unold [6].They used hadoop and apache spark platform to implement the system. MapReduce paradigm was used in this study to process large datasets for reducing problem of parallel programming. The limitation of access time and memory this prototype does not give better outcome for large amount of data.

Chen et al. [7]used the CCAM (co-clustering with augmented matrices) to develop multiple methods like heuristic scoring, conventional classification and machine learning to construct a recommendation system as well as the integration of content-based hybrid recommendation systems in collaboration with collaborative filtering model.

Zhou et al. [8] developed a collaborative filtering based ALS Algorithm for the Netflix Prize.ALS works to solve scalability issue of extensive datasets. This study used ALS algorithm to develop a movie recommendation system for predicting rating of users. Due to not refining the Restricted Boltzmann Machine (RBM) this system cannot show moderately better result.

The item-based collaborative filtering strategies was used by Dianping et al. [9].In this system, the user item rating matrix is first examined and the associations between different items categorized and use these relationships are used to evaluate the user's recommendations.

Dev et al. [10] have proposed a scalable method for developing recommendation systems based on similarity joins. To work with big data applications, MapReduce framework was used to design the system. Using a method called extended prefix filtering (REF), the process can significantly reduce the unnecessary overhead computation such as redundant comparisons in the similarity computing step.

Zeng et al. [11] proposed PLGM to boost the accuracy, scalability and processing power of large-scale data. Two matrix factorization algorithms, which are ALS and SGD, were considered in their work. SGD based parallel matrix factorization was implemented on spark and for its efficiency was compare with ALS and MLlib. Based on test result, advantage and disadvantage of each model was analyzed. A number of approaches to profile aggregation were studied and the model was adopted which gave the best result in terms of efficiency and precision; such as PLGM and LGM have been studied
.

Jooa et al. [12] was implemented a recommendation system by using both collaborative filtering and association rules. Distance data from Global positing system (GPS) was focused to recommend the items to customer for purchasing on the basis of customer's preference. The problem of information overloading solved by using Near Field Communication (NFC).NFC used to select high preference of user by analyzing time slot with association rules.

A combination of collaborative filtering and cluster techniques based recommendation system was proposed by Panigrahi et al. [13]. Apache spark platform was used to accelerate the running time of memory computation for performing recommendation. Spark native language scala used for speeding computational time. This study focused to reduce the cold start problem which is limitation of traditional collaborative filtering by correlating the customer to items on the base of features. But limitation of programming feature and verbose code of scala programming language this model did not evaluate better prediction on recommendation system.

An algorithm is proposed by Li et al [14] for paralleling the frequent item set of large dataset. This proposed algorithm discovered hidden pattern to help query recommendation. By calculating between cluster nodes this algorithm able to minimize computation cost. They gathered better performance for distributed processing by utilizing MapReduce framework concerning cluster computers. This study does not analyzing the query logs for google search engine.

From the analysis of previous work, the drawback is shown for over fitting issue of sparse data. Our proposed system overcomes this issue by applying matrix factorization in ALS algorithm. The use of matrix factorization in our system also increases prediction accuracy from previous researches.

3 3 Alternating Least Squares (ALS)

E-commerce companies encounter challenges in personalized product recommendation like Amazon and Netflix. In this personalized settings users rate the items and the ratings data are used to predict to find ratings for other items.
The rating data can be represented as an $m \times n$ matrix R where n users and m items. The $(u, i)^{th}$ entry is r_{ui} in the matrix R which implies that $(u.i)^{th}$ ratings for ith item by u user. The R matrix is sparse matrix as items do not receive ratings from many users. Therefore, the R matrix has the most missing values. Matrix factorization is the solution of this sparse matrix problem. There are two k dimensional vectors which are referred to as "factors".

- x_u is k dimensional vectors summarizing's every user u.
- y_i is k dimensional vectors summarizing's every item i.

Let,

$$r_{ui} \approx x_u^T Y_i \tag{1}$$

$$x_u = x_1, x_2, ..., x_n \epsilon R^k \tag{2}$$

$$y_i = y_1, y_2, ..., y_n \epsilon R^k \tag{3}$$

Equation 1 can be formulated as an optimization problem to find:

$$argmin \sum_{r_u i} (r_{ui} - x_u^T y_i)^2 + \lambda (\sum_u \parallel x_u \parallel^2 + \sum_i \parallel y_i \parallel^2) \tag{4}$$

Here, $\lambda$is the regularization factor, which is used to solve overfitting problem, is referred to as weighted-λ -regularization. The value of λcan be tuned to solve over-fitting whereas default value is 1 [15].

Let, the set of variables x_u is constant then the objective function of y_i is convex and the set of variables y_i is constant then the objective function of x_u is convex. Hence the optimize value of x_u and y_i can be found repeating the aforementioned approach until the convergence. This is called ALS (Alternating Least Squares).

Algorithm : *Alternating Least Squares (ALS)*

Procedure ALS (x_u ,y_i)
Initialization $x_u \leftarrow 0$
Initialization matrix y_i with random values
Repeat
    Fix y_i , solve x_u by minimizing's the objective function (the sum of squared errors)
    Fix y_i solve y_i by minimizing the objective function similarly
Until reaching the maximum iteration
Return x_u ,y_i
End procedure

4 Predictive Accuracy Metrics

The recommendation system is evaluated using off-line analysis. During off-line analysis, there is no actual users rather as large dataset is split into a training and test set. The recommendation system is trained with the training dataset to predict the ratings given by users in the test dataset and evaluation is carried out by comparing the resulting prediction with the actual prediction in the test dataset. Popular predictive accuracy metrics such as mean absolute error (MAE), root mean squared error (RMSE) and mean user gain (MUG) are used

to measure the accuracy of the prediction made by the recommendation system.

$$MAE = \frac{1}{|r_{ui}|} \sum_{r_k \epsilon r_{ui}} |r_i(r_k) - p_i(r_k)| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{|r_{ui}|}} \sum_{r_k \epsilon r_{ui}} |r_i(r_k) - p_i(r_k)| \tag{6}$$

Recommendation systems predict an item as interesting or not interesting to a user. So the accuracy of prediction made by recommendation system should also be measured with the amount of impact on users for a recommendation. This is user-specific accuracy metric often referred to as mean user gain (MUG) which measures the quality of a recommendation system helping users to buy right choice item.

$$UG(p_i(r_k)) = \{r_i(r_k)) - \theta_i if p_i(r_k) \geq \theta_i \tag{7}$$

Where $_i$ is the threshold value to calculate the quality of a recommendation system.

$$MUG = \frac{1}{|r_{ui}|} \sum_{r_k \epsilon r_{ui}} UG(p_i(p_k)) \tag{8}$$

5 System Prototype

The proposed system helps online customers with swift selection of their desired products among bigdata stored in a hadoop cluster. Customers, software bots/agents, and robotic shopping assistants are performing data transactions in the online shop-ping cart and e-commerce site frequently. Therefore, the size of information is growing up to zattabytes and processing of this requires significant amount of computing resources and time.

Apache spark runs on hadoop clusters. Apache spark can perform batch processing, stream processing, graph processing, interactive processing and in-memory processing of data. Applications built on Apache spark can process data in high speed as spark stores the Resilient Distributed Dataset (RDD) in the RAM of clusters nodes. RDD is immutable, logically partitioned, fault-tolerant records of the data as objects which are computed on different cluster nodes [16] [17] [18] [19]]. The dataset are created by different types of file such as json file, CSV file, text file etc. saved in the Hadoop file system. The text file RDDs are created by sparkcontext's textfile method.

In the system, the CSV file is loaded externally from HDFS in hadoop clusters. The file in RDD is logically partitioned which are stored in multiple nodes

of the cluster. The file is partitioned in smaller blocks so that RDD data can be distributed equally among threads. The full dataset is transformed coarse gained to take an input RDD producing many output RDDs. The transformation is pipelined to improve the per-formance. The transformation applies the computation like map(), filter(), reduce-ByKey(), groupByKey() etc. Spark delivers final result of all intermediate transfor-mations and write out to the HDFS of hadoop clusters, this is referred to as action. Examples of actions in spark are first(), take(), reduce(), collect(), count() etc.

The Recommendation system uses RDD based spark MLlib package. The spark MLlib implements alternating least squares (ALS) algorithm to predict the user's ratings of different items. The algorithm uses Matrix Factorization technique to pre-dict the explicit preferences of users to items.

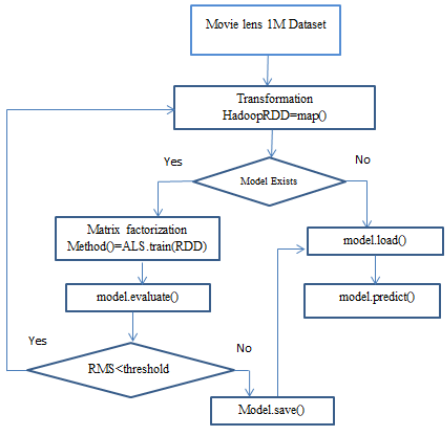

Fig. 1. Alternating Least Squares Matrix Factorization Method on Spark Recommendation System Workflow

6 System Architecture

The recommendation system is designed for interactive e-commerce sites. In this case, real time recommendation and retraining the recommendation system are two most important requirements. Therefore, dealing with increasing volume of data and change of data are major challenges. The number of transactions on e-commerce sites varies with time as new users arrive frequently. Predicting the new users recommendations is important to keep users attracted and for good feedback.
The Movielens 1M data set has 71567 users ratings to 10681 movies [20]. The

users in this data set have performed rating s for minimum 20 movies. There are 10000054 ratings available in this data set. The Movielens data set is explicit feedback data set here users give ratings for items.
The recommendation system architecture is depicted in the Fig 2. The system is run-ning on Apache Spark 2.4.5, Apache Hadoop 3.2.1 and Aapche HBase. The training dataset storage and retrieval operations along with Apache HBase data files are managed by HDFS. The Apache Spark MLlib produces predictions and stores the recommendations in HBase nosql database. The spark system consists of Master nodes, cluster manager and worker nodes. The RDDs are created and processed in Worker nodes and the cluster manager keeps records of RDD locations in the worker nodes. The users visualize the recommendations in the e-commerce websites browsed by smart phone clients and personal computer clients connected to the HTTP servers in the cloud.

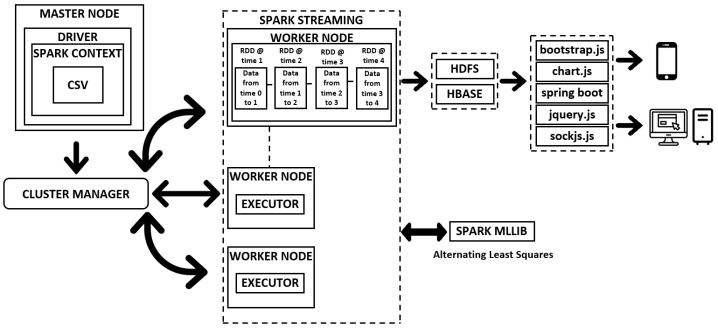

Fig. 2. System Architecture of the Proposed Recommendation System

The system stores and manages master dataset in a Hadoop distributed file system (HDFS) so that data can be accessed faster. The recommendations are pre-computed by processing master dataset in batch using Apache Spark and are stored in HBase. Real time views are computed on user data streaming by using Apache Spark stream-ing and are also stored in HBase. The real time recommendations are calculated by merging the batch view and real time view. Therefore, following the lambda architec-ture is a sustainable option to develop the system.
The project undergoes a task of experimenting the difference of total execution time required for calculating recommendations using Singular Value Decomposition train-ing algorithm in non-distributed system and spark system. The experiment is conduct-ed on a desktop with AMD Sempron(tm) processor (2.8 GHz) and 4GB RAM. The experiment shows that the total execution time is reduced in spark system significantly shown in the Table 1.

Table 1. Executions Time Comparison in Dataset.

	Dataset Total Executions time	**Non-distributed**
Movielens	1M	6 minutes 13 seconds
Apache Spark Movielens	1M	4minutes 43seconds

7 Experimental Result and Discussion

In this research, the recommender system is trained using Alternating Least Square (ALS) algorithm. The Movielens performance metric RMSE value is dependent on these parameters. The objective of the training is to decrease the value of RMSE and the ALS model producing least minimum RMSE value is saved for future recommen-dation. The system is implemented on Apache Spark and Apache Hadoop cluster. Therefore, the hardware configuration of the cluster's master node, data nodes, node manager, resource manager, name node, secondary name node, and history servers have impact on the performance.

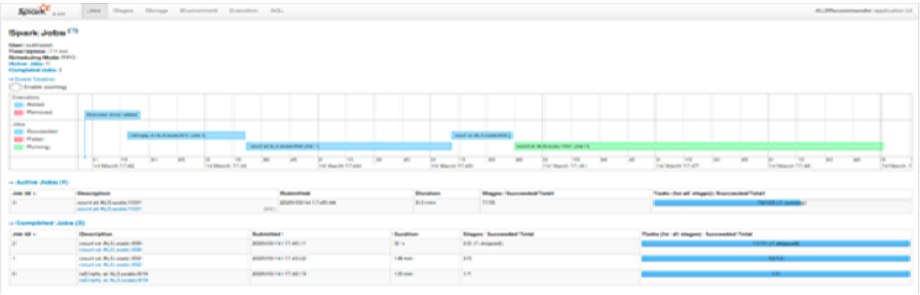

Fig. 3. Spark Jobs Execution Gantt chart for Active Jobs.

The fig 3 shows the process of job submission in Spark. It is noticed that the maxi-mum time is required for building top recommendations for each user and each movie. Because, during the job execution the datasets are divided into 20 RDDs (Resilient Distributed Dataset) for user Factors and item Factors equally 10 each.

The results for these two factors are aggregated to produce recommendations. Initial parameter for the system is shown in Table 2.

Table 3 shows the top 5 recommendations of items for a particular user along with the average rating and prediction values. The RMSE value resulted as 0.9 in this experiment.

Table 2. Parameter Setting.

Parameters	Value
Lambda	0.01
Iteration	5

Table 3. Recommendations using ALS Algorithm.

UserID	ItemID	Avg Rating	Prediction
5192	557	5	5.402
5192	864	4	5.284
5192	2512	3.9	5.189
5192	1851	4	5.125
5192	2905	4.6	5.098

As shown in Table 4 the training process includes a RMSE threshold check step where the lambda value can be changed to optimize the recommendation model. In Figure 4 by keeping the number of iteration fixed at 5 and by changing the lambda value in the range between 0.005 to 1 the minimum RMSE value is found as 0.870 against the lambda value 0.05. From Table 4 it has been shown that as when the lambda value is between 0.005 and 0.006 the RMSE value is respectively 0.907 and 0.905.As the lambda value is increased the value of RMSE is reduced than the previous value. After that RMSE value is changed as the lambda value is increased and minimum RMSE value is found when the lambda value is 0.05. The minimum RMSE value is 0.870 at iteration number 5.

Table 4. RMSE Value with Number of Iteration.

Lamda	Iteration	RMSE
0.005	5	0.907
0.006	5	0.905
0.008	5	0.898
0.009	5	0.893
0.01	5	0.896
0.05	5	0.870
0.1	5	0.892
0.15	5	0.916
0.2	5	0.931
0.25	5	0.941
0.5	5	1.022

In this research, a comparative analysis is performed to evaluate the performance of proposed recommendation system training algorithm Alternating Least Square (ALS). Therefore, a comparison is carried out among ALS, SVD++

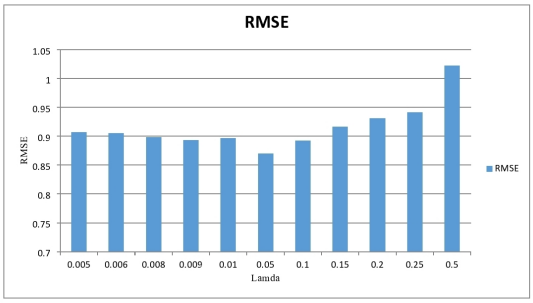

Fig. 4. Graphical Representation of RMSE value.

and Parallelized SGD matrix factorization methods. The result shows that the RMSE value is significantly reduced using ALS matrix factorization method and the RMSE is 0.870. Consequently, the experiment shows that the ALS algorithm is suitable for training explicit feed-back data set where users provide ratings for items.

Table 5. RMSE Comparison among ALS, SVD++ and SGD.

Algorithm	**RMSE**
ALS	0.870
SVD++	0.977
SGD	0.919

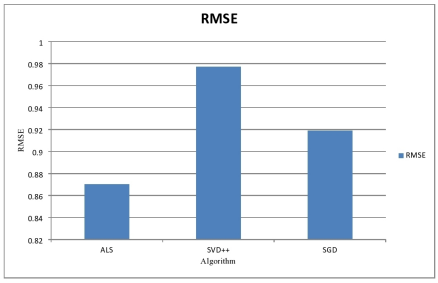

Fig. 5. Graphical Representation of RMSE Comparison among ALS, SVD++ and SGD.

8 Conclusion & Future Work

Apache spark is well suited for applications which require high speed query of data, transformation and analytics results. Therefore, the recommendation system developed in this research is implemented on Apache Spark. Also, the matrix factorization using Alternating Least Squares (ALS) algorithm which is a type of collaborative filtering is used to solve overfitting issues in sparse data and increases prediction ac-curacy. The overfitting problem arises in the data as the user-item rating matrix is sparse. In this research a recommendation system for e-commerce using alternating least squares (ALS) matrix factorization method on Apache Spark MLlib is developed. The research shows that the RMSE value is significantly reduced using ALS matrix factorization method and the RMSE is 0.870. Consequently, it is shown that the ALS algorithm is suitable for training explicit feedback data set where users provide ratings for items. The proposed recommendation system architecture is designed following lambda architecture so that the users can see the recommendations real time. This will in turn bring enormous benefit to the customers during purchasing goods.

In future, the recommendation system will be integrated with content based filtering, users click stream predictive analytics, deep collaborative filtering, and convolutional neural network based [21] [22] [23] [24] recommendation system . The integration might use results produced by these different systems and correlate these results to show recommendations to the online users. Thus, the future research direction will be focusing on hybrid models of recommendation systems [25] [26] [27] [28] [29] [30] [31] [32].

References

1. B.J Pine, "Mass customization,"(Vol. 17). Boston: Harvard business school press,1993.
2. R. Shahjee,"The impact of electronic commerce on business organization," Scholarly Research Journal for interdisciplinary studies, 4(27), 3130-3140,2016.
3. J.P Verma, B. Patel, A. Patel, "Big data analysis: recommendation system with Hadoop framework," In 2015 IEEE International Conference on Computational Intelligence Communication Technology (pp. 92-97), 2015, February, IEEE.
4. Alzogbi, P. Koleva, G. Lausen, "Towards Distributed Multi-model Learning on Apache Spark for Model-Based Recommender," In 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW) (pp. 193-200). IEEE,2019,April.
5. J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, "Collaborative filtering and deep learning based hybrid recommendation for cold start problem," In 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) (pp.874-877). IEEE,2016,August.
6. B. Kupisz, O. Unold, "Collaborative filtering recommendation algorithm based on Hadoop and Spark," In 2015 IEEE International Conference on Industrial Technology (ICIT). IEEE,2015.

7. Y. C Chen, "User behavior analysis and commodity recommendation for point-earning apps," In 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI). IEEE,2016.
8. Y.H Zhou, D. Wilkinson, R. Schreiber, "Large scale parallel collaborative filtering for the Netflix prize," In Proceedings of 4th International Conference on Algorithmic Aspects in Information and Management (pp. 337–348). Shanghai: Springer,2008
9. L.T Ponnam, et al, " Movie recommender system using item based collaborative-filtering technique," In International Conference on Emerging Trends in Engineering, Technology, and Science (ICETETS). IEEE,2016.
10. A.V Dev, A. Mohan, "Recommendation system for big data applications based on set similarity of user preferences," In International Conference on Next Generation Intelligent Systems (ICNGIS). IEEE.2026.
11. X. Zeng, et al, "Parallelization of latent group model for group recommendation algorithm," In IEEE International Conference on Data Science in Cyberspace (DSC). IEEE,2016.
12. J. Jooa, S. Bangb, G. Parka. "Implementation of a recommendation system using association rules and collaborative filtering". Procedia Computer Science, 91, 944-952.2016
13. S.Panigrahi, R.K Lenka, A. Stitipragyan. "A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark". In ANT/SEIT (pp. 1000-1006),1026,january.
14. Y. Koren, R. Bell, C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, 42(8), pp.30-37,2009.
15. X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D.B. Tsai, M. Amde, S. Owen, D. Xin, "Mllib: Machine learning in apache spark," The Journal of Machine Learning Research, 17(1), pp.1235-1241,2016.
16. Li, H., Wang, Y., Zhang, D., Zhang, M., Chang, E. Y. (2008, October). Pfp: parallel fp-growth for query recommendation. In Proceedings of the 2008 ACM conference on Recommender systems (pp. 107-114).
17. 17. J.G Shanahan, L. Dai, "Large scale distributed data science using apache spark," In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 2323-2324),2015,August.
18. S. Sharma "Dynamic Hashtag Interactions and Recommendations: An Implementation Using Apache Spark Streaming and GraphX ," In Data Management, Analytics and Innovation (pp. 723-738). Springer, Singapore,2020
19. M. Armbrust, R.S. Xin, C. Lian, Y. Huai, D. Liu, J.K Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, M. Zaharia, M, "Spark sql: Relational data processing in spark," In Proceedings of the 2015 ACM SIGMOD international conference on management of data (pp. 1383-1394),2015,May.
20. F.M Harper, J.A Konstan. "The movielens datasets: History and context". Acm transactions on interactive intelligent systems (tiis), 5(4), 1-19,2015.
21. Chowdhury, R. R., Hossain, M. S., ul Islam, R., Andersson, K., Hossain, S. (2019, May). Bangla handwritten character recognition using convolutional neural network with data augmentation. In 2019 Joint 8th International Conference on Informatics, Electronics Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision Pattern Recognition (icIVPR) (pp. 318-323). IEEE.
22. Ahmed, T. U., Hossain, M. S., Alam, M. J., Andersson, K. (2019, December). An Integrated CNN-RNN Framework to Assess Road Crack. In 2019 22nd International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

23. Ahmed, T. U., Hossain, S., Hossain, M. S., ul Islam, R., Andersson, K. (2019, May). Facial expression recognition using convolutional neural network with data augmentation. In 2019 Joint 8th International Conference on Informatics, Electronics Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision Pattern Recognition (icIVPR) (pp. 336-341). IEEE.
24. Islam, M. Z., Hossain, M. S., ul Islam, R., Andersson, K. (2019, May). Static hand gesture recognition using convolutional neural network with data augmentation.
25. Sinitsyn, Sergey, et al. "The Concept of Information Modeling in Interactive Intelligence Systems." International Conference on Intelligent Computing Optimization. Springer, Cham, 2019.
26. Alhendawi, Kamal Mohammed, and Ala Aldeen Al-Janabi. "An intelligent expert system for management information system failure diagnosis." International Conference on Intelligent Computing Optimization. Springer, Cham, 2018.
27. Biswas, M., Chowdhury, S. U., Nahar, N., Hossain, M. S., Andersson, K. (2019, November). A Belief Rule Base Expert System for staging Non-Small Cell Lung Cancer under Uncertainty. In 2019 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON) (pp. 47-52). IEEE.
28. Kabir, S., Islam, R. U., Hossain, M. S., Andersson, K. (2020). An Integrated Approach of Belief Rule Base and Deep Learning to Predict Air Pollution. Sensors, 20(7), 1956.
29. Monrat, A. A., Islam, R. U., Hossain, M. S., Andersson, K. (2018, October). A belief rule based flood risk assessment expert system using real time sensor data streaming. In 2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops) (pp. 38-45). IEEE.
30. Karim, R., Hossain, M. S., Khalid, M. S., Mustafa, R., Bhuiyan, T. A. (2016, September). A belief rule-based expert system to assess bronchiolitis suspicion from signs and symptoms under uncertainty. In Proceedings of SAI Intelligent Systems Conference (pp. 331-343). Springer, Cham.
31. Hossain, M. S., Monrat, A. A., Hasan, M., Karim, R., Bhuiyan, T. A., Khalid, M. S. (2016, May). A belief rule-based expert system to assess mental disorder under uncertainty. In 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV) (pp. 1089-1094). IEEE.
32. 32. Hossain, M. S., Habib, I. B., Andersson, K. (2017, July). A belief rule based expert system to diagnose dengue fever under uncertainty. In 2017 Computing conference (pp. 179-186). IEEE.