

Matrix Factorization Based Recommendation System using Hybrid Optimization Technique

P. Srinivasa Rao^{1,*}, T.V. Madhusudhana Rao², Suresh Kurumalla³, and Bethapudi Prakash⁴

¹Associate Professor of CSE, MVGR College of Engineering, Vizianagaram, Andhrapradesh, India.

²Department of CSE, Vignan's Institute of Information Technology, Visakhapatnam, India.

³Department of CSE, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India.

⁴Department of IT, Vignan's Institute of Engineering for Women, Visakhapatnam, India.

psr.sri@gmail.com, madhu11211@gmail.com, kurumallasuresh@gmail.com, prakash.vza@gmail.com.

Abstract

In this paper, a matrix factorization recommendation algorithm is used to recommend items to the user by inculcating a hybrid optimization technique that combines Alternating Least Squares (ALS) and Stochastic Gradient Descent (SGD) in the advanced stage and compares the two individual algorithms with the hybrid model. This hybrid optimization algorithm can be easily implemented in the real world as a cold start can be easily reduced. The hybrid technique proposed is set side-by-side with the ALS and SGD algorithms individually to assess the pros and cons and the requirements to be met to choose a specific technique in a specific domain. The metric used for comparison and evaluation of this technique is Mean Squared Error (MSE).

Keywords: matrix factorization, ALS, SGD, optimization, recommendation system, latent factor, collaborative filtering.

Received on 30 December 2020, accepted on 13 February 2021, published on 19 February 2021.

Copyright © 2021 P. Srinivasa Rao *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](https://creativecommons.org/licenses/by/4.0/), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.19-2-2021.168725

*Corresponding author. Email: psr.sri@gmail.com

1. Introduction

There's an explosive growth of information in the present world, giving a tough time to find information that is appropriate from a tremendous amount of data present online. The data required obtained becomes mystical on the internet [3] [7]. So, we rely highly on recommendations [22]. A recommender system comes under the family of knowledge filtering systems whose main cause is a prediction [11] of the user's rating values to an item and thus drawing the interests of the user from the available data. Recommendation systems are implemented in a wide range of areas, including news, movies, text mining, books, etc... Not every recommender system is capable of handling all kinds of situations. Recommender systems can be classified into two strategies. The first category is the content-based filtering approach responsible for creating a profile for every user or product to delineate its nature. Consider, for example, a

movie profile that contains characteristics based on its genre, the cast and crew, the popularity of that movie, etc. Similarly, the profiles of users contain arithmetical information or answers to a suitable questionnaire. The profiles that eventuate can be utilized to map users with matching products through programs. The disadvantage of a content-based recommendation system is it is necessary to obtain external information which is very difficult to muster in practice [9].

Collaborative filtering (CF), which plays a crucial part in generating personalized recommendations, is an alternative strategy and also among the most traditional and striking recommendation algorithms [10][19]. CF inspects user dependencies and product relationships to identify new user-item associations [8][17]. It can be observed that in some scenarios, few CF systems determine items pairs that are similarly rated or harmonious users with a compatible history of purchasing or rating to infer foreign relationships between users and items with just the data about the history of users which

could be the way those users rate items or could be their previous transactions [13][14]. CF is domain free but still better than the content-based approach as it can deal with kinds of information that can be ambiguous and difficult [9]. However, the CF technique hampers the sparse user rating matrix problem, resulting in poor recommendation precision. A common solution to this type of trouble is replacing the missing values with the average value of all the ratings specific to that user or item. This approach can decrease the imprecision to only some extent. The value with which the missing values are being replaced has a serious effect on the produced recommendations' credibility [23].

Due to the leverages of perpetuating accuracy for scaling information, less estimation cost, and deducing the issues from high sparsity levels, matrix factorization latent factor models have received spotlight for their work in the recent days [6][21]. This is the most widely used collaborative filtering approach to identify hidden factors that affect the user's preferences. The Matrix factorization recommendation system approach is more memory-efficient and more specific than the similarity-based recommendation process, which takes the resemblances between users and objects to make suggestions only into consideration [4][12]. Matrix factorization is widely applied in collaborative filtering, and we could apply SGD and ALS as the learning algorithm. This type of method in which a large matrix is decomposed into smaller matrices is called Matrix Factorization, and it was invented and popularized by Netflix prize winners. This method boosted the performance of recommendation systems from old methods which were mostly neighbourhood-based. This kind of decomposition aims to find latent factors and reduce the dimensions [18].

Present matrix factorization methods are engineered with precise feedback data, which analyzes the data easy [18]. However, feedback pages for customer reviews and ratings have to design by the service providers, which is a difficult process and time consuming due to user involvement. Therefore, recommenders can speculate user preferences from the more ample implicit feedback, which indirectly makes suggestions based on the user's behaviour [4]. Online shopping history, browsing data, previous search patterns, or even mouse movements are few implicit feedback data types. For example, a user that bought many products from a brand probably likes that brand [5][6]. To develop a modified matrix factorization to take up precise feedback data, we integrate the advantages of alternating least squares and stochastic gradient descent into a Matrix Factorization recommendation system using incremental stochastic gradient descent alternating least squares updating technique [1].

A prominent literature part in the domain focuses on handling the explicit feedback; using this type of direct data from the users is much appreciated. But practically in many scenarios, recommender systems have to be mainly focused on latent data [5]. This may overcome users' hesitation or disinterest to rate items and allow the system

to gather feedback data explicitly. In a latent model, once the user accepts the cookies and gives permission to collect user data, they need not provide explicit feedback anymore (e.g., ratings). For interpretation of implicit-feedback, the recommender has to consider proper measures. In the conventional methods [15] [20], a user specifies a numeric score, and there are clear metrics such as mean squared error to calculate the success rate by predictions. Moreover, with latent models, the availability, competition of the item with other items, and repeat feedback have to be considered [24].

The remainder of the paper is sequenced as follows: Section 2 illustrates related work, the methodology is described in section 3, section 4 explains hybrid optimization technique, and section 5 details about environmental setup, section 6 describes the experimental evaluation, the conclusion is exploited in section 7, and section 8 briefs about future work [25].

2. Related Work

In the past few years, extensive work has been done by many researchers on recommendation systems. Hongmei H. Li et al. [1] have proposed a recommendation framework method with an all-weighted strategy, a more efficient and better-optimized scheme. Based on the outcomes of experimentation on a pair of recommendation techniques, it is determined that the suggested approach surpasses several prediction-oriented and ranking-oriented evaluation metrics. C. Lin et al. [2] presented a RI-SGD model developed for adept computations and precise time-variant implicit feedback MF recommendation system, consisting of ALS with weight regularization developing stage and SGD in the modifying stage. In comparison with the process of retraining the full model, the calculated scores display that the RI-SGD approach can obtain similar recommendation accuracy, but requires only about 0.02% of the retraining time. To evaluate the recommended quality, the discounted cumulative gain (DCG) technique is applied. DCG, which interprets ranking quality. M. Li et al. [5] has put forth an improvised model named TimeMF, based on latent feedback and includes temporal information, which is one of the main solutions for information overloading in social e-commerce networks by addressing the absence of negative information in user history. The optimized model gives a unique learning rate to each feature of the latent feature matrix and adopts adaptive gradient descent to update the learning rate to upgrade the accuracy level. The outcomes from experimentation prove that this model surpasses the standards concerning ranking-oriented evaluation. Y. He et al. [6] have proposed a novel model named Correlated Matrix Factorization (CMF) which integrates Matrix Factorization (MF) and Canonical Correlation Analysis (CCA) into a single enhanced model so that the interactions between the user and the items are observed precisely. The model is applied on four publicly available datasets: namely Yelp, Flixster, MovieLens, and

Ciao. A thorough analysis of four different datasets shows that CMF is competitive and better than present state-of-the-art baselines. J. Z. Sun et al. [16] proposed a new algorithm for prediction, estimation, and recommendation called the collaborative Kalman filter. In this paper, the author proposed an amplified Gaussian PMF which considers the user behaviour trajectories. The approach extends probabilistic matrix factorization in time through a state-space model. This leads to an estimation procedure with parallel Kalman filters and smoothers coupled through item factors. Learning of global parameters uses the expectation-maximization algorithm. When analyzed with current methods, this technique has shown that it is better on the computed information and movie recommendation data in the real-world. Evaluation metrics used are Root mean square error (RMSE) and Collaborative Kalman filtering (CKF).

According to the literature survey, many methods were examined for recommendation systems using different frameworks. In this paper, a recommendation system based on matrix factorization using a hybrid optimization technique has been explored to improve the algorithm's efficiency.

3. Methodology

The proposed technique is an improved latent factor collaborative filtering model using a combination of the optimization algorithms stochastic gradient descent and alternating least squares for Matrix Factorization recommendation systems using explicit feedback from the users.

3.1. Latent Factor Collaborative Filtering Approach

The latent factor model is an advanced system of content-based recommendation techniques. It works on the supposition that one should know the factors that control the user's preferences on the respective item. Depending on the closeness of a user i with the hidden factors, the relation can be defined as u_i , and its corresponding latent factor v_j can define an item j . The high rating score is obtained usually when the two factors match (same as content-based filtering) which can be calculated using the inner product of the user's latent factors with the item's latent factors. The rating of the i th user on the j th item is modelled as:

$$x_{i,j} = u_i v_j, \forall i, j \quad (1)$$

If we take into account the complete rating matrix involving M users and N items, the equation (1) can be modified as follows, i.e., equation (2) & (3),

$$X = UV \text{ where } U = [u_1 | \dots | u_m] \quad (2)$$

and

$$V^T = [v_1 | \dots | v_N] \quad (3)$$

If the rating matrix is filled with no missing values, there would be no trouble generating personalized recommendations. But practically that's absurd as in practice with the extreme loads of data; the matrix is emptier than it's filled. Now, finding these missing values and filling them is the main issue. Once it is completed, the prediction for users on items can be made with improved accuracy. This can be represented in mathematical form as:

$$Y = R.X = R.(UV) \quad (4)$$

In the above equation (4), R is a sample binary matrix with values 1 if the ratings are known and 0 if unknown. And the symbol \cdot represents the dot product. By solving the following problem, estimation of the matrices of the implicit factor for the users and the products is possible as:

$$\min_{U,V} \|Y - R.(UV)\|_F^2 + \gamma(\|U\|_F^2 + \|V\|_F^2) \quad (5)$$

In the above equation (5), F represents the Frobenius norm which is used to solve the problem of over-fitting. For solving the above equation (5), many techniques are available from least squares to multiplicative updates or gradient descents. The best algorithm used is the simple alternating least squares to utilize the technique in practice.

For a bi-linear cost function like (5) the convergence of the values to a global minimum through iterative procedures is highly impossible and can only guarantee local minima convergence. One solution to this issue is to solve the ratings themselves directly rather than the factors by considering a supposition that the rating matrix is of low-rank. The rank of the matrix is considered the same as the number of factors. Explaining the data Y optimally, and finding X is the best and straightforward method for the ratings to be solved. There is no flexible fix to this issue as the rank minimization is known to be NP-hard. This issue can be tended by following the theoretical study that has shown that a low-rank solution (under certain assumptions) can be obtained by relying on the nuclear norm and relaxing the NP-hard rank minimization problem to its closest convex surrogate. This can be represented in the equation form as,

$$\min_x \|Y - R.X\|_F^2 + \gamma\|X\|_{NN} \quad (6)$$

In equation (6) shown above, NN or Nuclear Norm defines the sum of singular values. Equation (6) is a convex problem that semi-definite programming solvers can treat, for instance. In the present day, more advanced techniques are present. Compared to the number of values present in Y , X values are extremely less; for theoretical issues, only 5% of the data is available. In practical

scenarios, less than 1% of the information is present. And hence collaborative filtering became a highly underdetermined problem. In such a case, to advance the outcomes, even secondary information can also be utilized, as shown in figure 1.

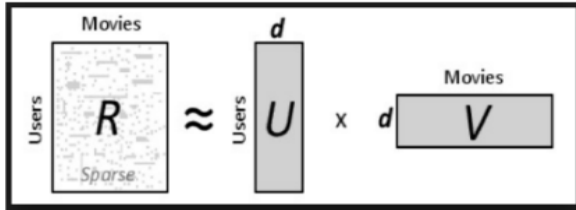


Figure 1. The Matrix Factorization technique

4. Proposed Hybrid optimization technique approach

Considering all the techniques proposed in the literature, a hybrid optimization technique is introduced, combining the Alternating Least Squares optimization technique with the Stochastic Gradient Descent technique for updating the model. By integrating the two optimization techniques using explicit user feedback, a practically possible updating of user data can be achieved and thus reduces the computational time and effort.

4.1. Training the model

To develop a matrix factorization model effectively for a few million items and users, ALS is used in training the model using the equations as shown in (7) & (8). This technique aims to minimize the cost function by uncovering the hidden, latent factors users and items.

Algorithm-1: ALS for Matrix Completion

Initialize X, Y

repeat

for u = 1 ... n do

$$x_u = (\sum_{r_{ui} \in r_{u*}} y_i y_i^T + \gamma I_k)^{-1} \sum_{r_{ui} \in r_{u*}} r_{ui} y_i \quad (7)$$

end for

for i = 1 ... m do

$$y_i = (\sum_{r_{ui} \in r_{u*}} x_u x_u^T + \gamma I_k)^{-1} \sum_{r_{ui} \in r_{u*}} r_{ui} x_u \quad (8)$$

end for

until convergence

4.2. Minimizing the error function

The vector for each user (x_u) and item (y_i) in feature dimensions has to be obtained to diminish the below loss equation.

$$\min_{y_*, y_*} \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \gamma (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) \quad (9)$$

In the above equation (9), the r_{ui} is the true rating of the item. The two terms at the end of the function are added to prevent overfitting the user and item vectors. The main motto is to minimize this loss function.

4.3. Training the model using SGD

The ALS technique iteratively updates the complete model with now involving the new entries in the model. But this process is extremely time-taking. In contrast, SGD is comparatively quick when calculating one user and item is taken into account. Henceforth, iteratively advanced SGD is also put forward so that the matrix factorization recommendation model is modified. In equation (9) the first-order gradient of the cost function is calculated, and it's progressed as following functions which are equation (10) & (11):

$$\frac{\partial C(x,y)}{\partial x_u} = \gamma ((r_{u,i}(t) - x_u^T y_i) y_i) + 2\delta x_u \quad (10)$$

$$\frac{\partial C(x,y)}{\partial y_i} = \gamma ((r_{u,i}(t) - x_u^T y_i) x_u) + 2\delta y_i \quad (11)$$

Now suppose that the new rating values r_{ui} are incoming as new inputs and that latent feature matrices X, Y fit in each machine's memory. Consider a supposition that the incoming values are shuffled and can use Stochastic Gradient Descent (SGD) to advance the X, Y feature matrices, as shown in equation (12) & (13).

Algorithm-2: Streaming ALS using SGD

for new r_{ui} do

$$x_u \leftarrow x_u - \alpha (r_{ui} - x_u^T y_i) y_i + \gamma x_u \quad (12)$$

$$y_i \leftarrow y_i - \alpha (r_{ui} - x_u^T y_i) x_u + \gamma y_i \quad (13)$$

end for

4.4. Flowchart

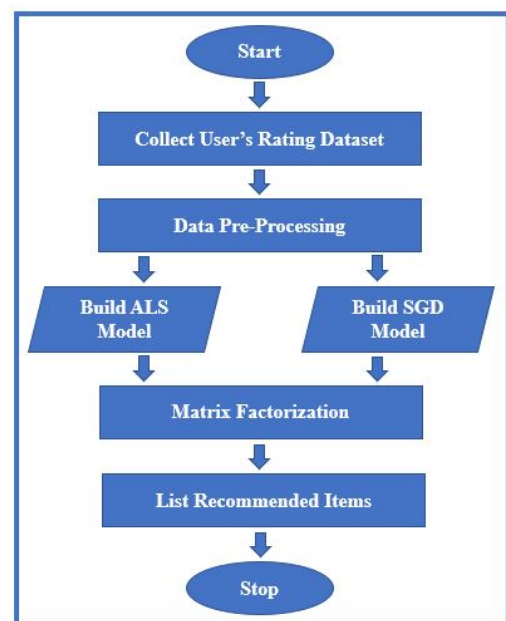


Figure 2. Proposed Matrix Factorization technique with a hybrid optimization

The above flowchart (figure 2) explains the processing of recommendation data by taking the user’s rating dataset as input. The model is trained using the ALS optimization technique induced in the MF algorithm. The model is retrained using the SGD optimization technique, and the obtained results are combined henceforth.

5. Environmental Setup

The dataset utilized in this article is an ml small dataset, from the Movie Lens website for training and testing data with a rating scale of 1 lakh users. The language used for this method is Python of version 3.7 with the platform Jupyter Notebook in software Anaconda.

6. Experimental Evaluation

This approach’s evaluation metric is the Mean Squared Error (MSE), as shown in equation (14). MSE calculates the measure of error that is present in between training and testing datasets. It can also be defined as it compares a predicted value with the observed or known value. The smaller an MSE value, the closer the predicted and observed values are, and the accuracy is better.

If $X_{obs, i}$ is the predicted rating on item i by user u , and $X_{model, i}$ is the true rating on item i by user u , MSE of n corresponding rating-prediction pairs are defined as:

$$MSE = \frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{n} \quad (14)$$

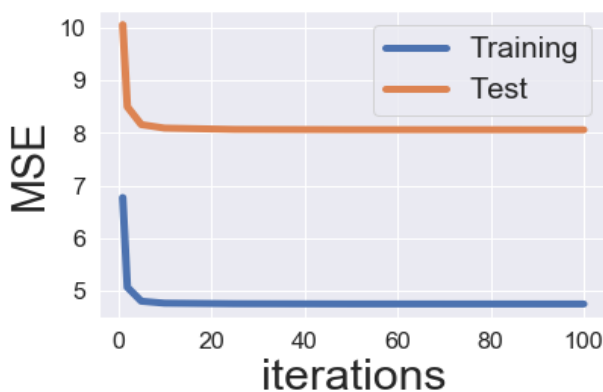


Figure 3. MSE Evaluation Graph for ALS algorithm

Figure 3 represents the MSE metric comparison of the training and testing data for the ALS algorithm, otherwise a learning curve of Matrix Factorization model. We can observe a reasonable amount of overfitting from the learning curve since the obtained test MSE is approximately 30% greater than the train MSE. The best performing parameters for the ALS algorithm were 20

latent factors and regularization parameter 0.1 with 100 iterations through the experimental analysis.

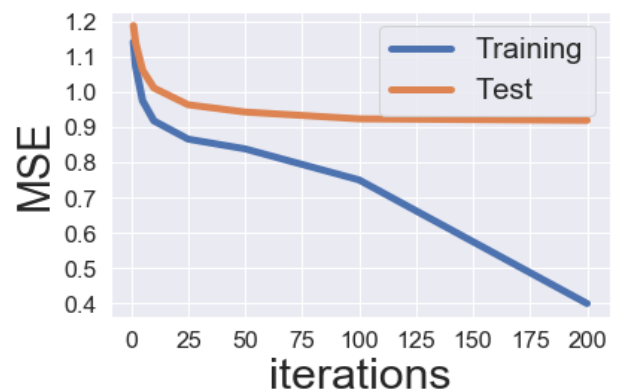


Figure 4. MSE Evaluation Graph for SGD algorithm

Figure 4 represents the MSE learning curve of the training and testing data for the SGD algorithm of the Matrix Factorization model. We can observe that the overfitting is reduced from this curve compared with the ALS algorithm to approximately 20% less. The best performing parameters for the SGD algorithm were 80 latent factors and regularization parameter 0.001 with 200 iterations through the experimental analysis.

The recommendations obtained individually by the two algorithms ALS and SGD are grouped by performing an inner join to output the user’s final recommendations. During the comparison of optimization algorithms, the crucial detail for all the end-users is time-to-solution. When set side-by-side the convergence rates of ALS, SGD, and hybrid optimization algorithm which combines the previous two algorithms, it is showed that no algorithm could act best in all the domains. Moreover, it can be observed that gradient descent is persistently quicker compared to alternating least squares in most of the domains. Alternating least squares scales better on the Movie Lens dataset which is extremely sparse. Also, often alternating least squares perform better than gradient descent but typically do not overcome the performance loss that the algorithm starts with. Whenever dealing with implicit datasets, which are usually not sparse, SGD is not practical. ALS is a much more efficient optimization technique in these cases. When both the algorithms are combined, it sure adds the advantages of both of them like performing better on sparse data and execution speed. But alongside can bring some of the disadvantages like dealing with feedback data. If dealt with explicit data, SGD helps overcoming the performance loss of ALS and ALS can help in parallelizing the iterative execution.

Similarly, if dealt with implicit data, ALS can increase the efficiency of calculating latent factors, but SGD lacks the same, dealing with user history. Since the proposed hybrid model uses the users’ explicit data, the performance of the model overcomes the issues dealt with by the algorithms to some extent. But factors like the size

and sparsity of the dataset, type of user data, and so on need to be considered and analyzed when choosing a particular technique.

7. Conclusion

This paper proposes an effective and adequate recommendation system using matrix factorization technique with fast optimization scheme. The proposed framework addresses the cold-start problem by recommending popular items to the new user from user feedback while iteratively updating the model by considering both Alternating Least Squares and Stochastic Gradient Descent algorithms training phase of the model. The proposed framework has obtained better performance than the state-of-art algorithms.

7.1. Future Work

Exploring other collaborative filtering algorithms for hybrid optimization and using implicit feedback data, as the ALS algorithm works better while using user history rather than ratings are considered for future work.

References

- [1] H. Li, X. Diao, J. Cao and Q. Zheng, "Collaborative Filtering Recommendation Based on All-Weighted Matrix Factorization and Fast Optimization," in *IEEE Access*, vol. 6, pp. 25248-25260, 2018, doi: 10.1109/ACCESS.2018.2828401.
- [2] C. Lin, L. Wang and K. Tsai, "Hybrid Real-Time Matrix Factorization for Implicit Feedback Recommendation Systems," in *IEEE Access*, vol. 6, pp. 21369-21380, 2018, doi: 10.1109/ACCESS.2018.2819428.
- [3] P Srinivasa Rao, MHM Krishna Prasad, K Thammi Reddy, "A Novel Approach For Identification Of Hadoop Cloud Temporal Patterns Using Map Reduce" Published In *IJITCS (MECS)* Vol. 6, No. 4, Pp:37-42, March 2014.
- [4] B. Yi et al., "Deep Matrix Factorization With Implicit Feedback Embedding for Recommendation System," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4591-4601, Aug. 2019, doi: 10.1109/TII.2019.2893714.
- [5] M. Li, H. Wu and H. Zhang, "Matrix Factorization for Personalized Recommendation With Implicit Feedback and Temporal Information in Social Ecommerce Networks," in *IEEE Access*, vol. 7, pp. 141268-141276, 2019, doi: 10.1109/ACCESS.2019.2943959.
- [6] Y. He, C. Wang and C. Jiang, "Correlated Matrix Factorization for Recommendation with Implicit Feedback," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 451-464, 1 March 2019, doi: 10.1109/TKDE.2018.2840993.
- [7] P Srinivasa Rao, S Satyanarayana, "Privacy Preserving Data Publishing Based on Sensitivity in Context of Big Data Using Hive", *Journal of Bigdata* (Springer), Volume:5, Issue:20, ISSN: 2196-1115, July 2018.
- [8] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele and F. Xia, "Context-Based Collaborative Filtering for Citation Recommendation," in *IEEE Access*, vol. 3, pp. 1695-1703, 2015, doi: 10.1109/ACCESS.2015.2481320.
- [9] S. H. Choi, Y. Jeong and M. K. Jeong, "A Hybrid Recommendation Method with Reduced Data for Large-Scale Application," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 557-566, Sept. 2010, doi: 10.1109/TSMCC.2010.2046036.
- [10] Z. Chen, Z. Ning, Q. Xiong and M. S. Obaidat, "A Collaborative Filtering Recommendation-Based Scheme for WLANs With Differentiated Access Service," in *IEEE Systems Journal*, vol. 12, no. 1, pp. 1004-1014, March 2018, doi: 10.1109/JSYST.2016.2542179.
- [11] T. V. Madhusudhana Rao, P Srinivasa Rao, P.S. Latha Kalyampudi, "Iridology based Vital Organs Malfunctioning identification using Machine learning Techniques", *International Journal of Advanced Science and Technology*, Volume: 29, No. 5, PP: 5544 – 5554,2020.
- [12] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang and T. Zhang, "Collaborative Web Service Quality Prediction via Exploiting Matrix Factorization and Network Map," in *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 126-137, March 2016, doi: 10.1109/TNSM.2016.2517097.
- [13] S. Ajoudanian and M. N. Abadeh, "Recommending human resources to project leaders using a collaborative filtering-based recommender system: A case study of GitHub," *IET Softw.*, vol. 13, no. 5, pp. 379–385, Oct. 112019, doi: 10.1049/iet-sen.2018.5261.
- [14] S. Jiang, X. Qian, J. Shen, Y. Fu and T. Mei, "Author Topic Model-Based Collaborative Filtering for Personalized POI Recommendations," in *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 907-918, June 2015, doi: 10.1109/TMM.2015.2417506.
- [15] Madhusudhana Rao T. V., P. Srinivasa Rao, Srinivas, Y., "A Secure Framework for Cloud Using Map Reduce", *Journal of Advanced Research in Dynamical and Control Systems (IJARDCS)*, Volume:9, Sp-14, Pp:1850-1861, ISSN:1943-023x, Dec 2017.
- [16] J. Z. Sun, D. Parthasarathy and K. R. Varshney, "Collaborative Kalman Filtering for Dynamic Matrix Factorization," in *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3499-3509, July 15, 2014, doi: 10.1109/TSP.2014.2326618.
- [17] Y. Gao and L. Ran, "Collaborative Filtering Recommendation Algorithm for Heterogeneous Data Mining in the Internet of Things," in *IEEE Access*, vol. 7, pp. 123583-123591, 2019, doi: 10.1109/ACCESS.2019.2935224.
- [18] an, F., Li, L., Zhang, Z. et al., "A multi-attribute probabilistic matrix factorization model for personalized recommendation," *Pattern Anal Applic* 19, 857–866 (2016). <https://doi.org/10.1007/s10044-015-0510-2>.
- [19] Li J., Chang C., Yang Z., Fu H., Tang Y. (2019) Probability Matrix Factorization Algorithm for Course Recommendation System Fusing the Influence of Nearest Neighbor Users Based on Cloud Model. In: Tang Y., Zu Q., Rodríguez García J. (eds) *Human Centered Computing. HCC 2018. Lecture Notes in Computer Science*, vol 11354. Springer, Cham. https://doi.org/10.1007/978-3-030-15127-0_49.
- [20] P Srinivasa Rao, MHM Krishna Prasad, K Thammi Reddy, "An Efficient Semantic Ranked Keyword Search of Big Data Using Map Reduce", *IJDTA*, Vol.8, No. 6, Pp:47-56,2015.

- [21] Yuen MC., King I., Leung KS. (2012) TaskRec: Probabilistic Matrix Factorization in Task Recommendation in Crowdsourcing Systems. In: Huang T., Zeng Z., Li C., Leung C.S. (eds) Neural Information Processing. ICONIP 2012. Lecture Notes in Computer Science, vol 7664. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34481-7_63.
- [22] Pierre Hanna (2019), "Considering Durations and Replays to Improve Music Recommender Systems", SAS EAI, doi: 10.4108/eai.5-2-2018.156379.
- [23] Avijit Mondal, Radha Tamal Goswami, "Enhanced Honey-pot cryptographic scheme and privacy preservation for an effective prediction in cloud security, Microprocessors and Microsystems, Volume 81, 2021,103719, <https://doi.org/10.1016/j.micpro.2020.103719>.
- [24] D Datta, S Mishra, SS Rajest, (2020) "Quantification of tolerance limits of engineering system using uncertainty modeling for sustainable energy" International Journal of Intelligent Networks, Vol.1, 2020, pp.1-8, <https://doi.org/10.1016/j.ijin.2020.05.006>
- [25] Avijit Mondal, Arnab Kumar Das, Sayan Nath, Radha Tamal Goswami, "Review Study on Different Attack Strategies of Worm in a Network", Webology, Volume 17, Number 2, December, 2020.