https://doi.org/10.31449/inf.v48i9.5969 Informatica **48** (2024) 177–190 **177**

Application of Recommendation Algorithm Based on Matrix Dimensionality Reduction Model in Network Information Analysis Model

Pei Yan, Yu Zhou
School of Humanities and Management, Xi'an Traffic Engineering Institute, Xi'an, 710300, China
E-mail: yanpeixjy@126.com

Keywords: matrix dimensionality reduction model, recommendation algorithm, network information analysis, collaborative filtering, personalized recommendation system

Received: April 1, 2024

The rapid advancement of communication and Internet technology has led to a significant increase in the quantity of network information, which has in turn created a significant challenge for users attempting to obtain the information they require. To address this issue, this study applies a recommendation algorithm based on matrix dimensionality reduction model to network information analysis models. This study provides an in-depth analysis of collaborative filtering recommendation algorithms, proposes a matrix dimensionality reduction model based on singular value decomposition, and constructs a network information analysis model to achieve accurate user behavior prediction and personalized recommendations. The results demonstrated that the collaborative filtering recommendation algorithm reached convergence after 5000 iterations. Furthermore, it was observed that continuing to iterate 10000 times would not affect the loss function value. The error stabilized below 0.5 after 800 iterations. The accuracy and recall of the network information analysis model were both above 0.9, demonstrating good performance in network information analysis. The root means square error and mean absolute error of the constructed model were both within 0.15, indicating that it could give users with more accurate recommendations and decision support in practical applications, as well as recommended content that better met their needs. This study provides new ideas and methods for effective filtering and personalized services of online information.

Povzetek: Algoritem priporočanja, ki temelji na modelu zmanjšanja dimenzionalnosti matrike, izboljšuje analizo omrežnih informacij z uporabo kolaborativnega filtriranja in singularne vrednostne dekompozicije za personalizirane priporočilne sisteme.

1 Introduction

In Network Information Analysis (NIA), the number of users and items is considerable, yet the items that users actually score or interact with are relatively few, resulting in an extremely sparse user-item scoring matrix. This sparsity makes traditional similarity-based Recommendation Algorithm (RA) ineffective, as the calculation of similarity relies on a large number of common scores [1]. By reducing the dimension of data, the Matrix Dimensionality Reduction Model (MDRM) can capture the key features of users and items in sparse data, thus improving the accuracy of recommendations. With the continuous growth of network information, the characteristic dimensions of users and projects are also increasing [2]. High dimensional data not only increases the computational complexity, but also introduces noise and redundant information, which affects the recommendation effect. The MDRM can effectively reduce the dimension of data, reduce the computational complexity, and retain the main features of data to improve the efficiency of recommendation. With the diversification of network information, heterogeneous NIA model has gradually become the focus of research. Heterogeneous networks contain various types of information and nodes, and how to effectively integrate this heterogeneous information is one of the challenges faced by RA [3]. The MDRM offers certain advantages in the integration of heterogeneous information, as it is capable of processing a diverse range of data types and extracting key features from them. Network information is real-time and dynamic, and users' interests and needs will change with time. Consequently, RA must be capable of detecting and incorporating these alterations in real time, thereby enabling users to receive timely recommendations. The matrix dimension reduction model has certain challenges in terms of real-time and dynamic performance, because the dimensionality reduction process usually requires a certain calculation time. In the RA, the user's personal information and behavior data are valuable resources. However, how to protect users' privacy and data security is a problem that RA must face. When the matrix dimension reduction model processes user data, it needs to comply with relevant privacy and security regulations to ensure the safe and legal use of user data [4-5]. Therefore, this study applies the MDRM-based RA to the NIA model, hoping to make positive contributions to the development and application of the NIA model. The innovation of this study lies in reducing the data dimensionality and computational complexity, improving the RA efficiency, while retaining the main features of the data to ensure an improvement in recommendation accuracy. It aims to apply matrix dimensionality reduction technology to heterogeneous NIA models to achieve accurate prediction of user behavior and personalized recommendations. The content has four parts. Part 1 is the introduction, which discusses the related research. Part 2 is the NIA study about Collaborative Filtering (CF)-RA grounded on MDRM. Part 3 is the performance testing of the NIA model built on the CF. Part 4 summarizes the paper.

2 Related works

In recent years, the CF-RA based on MDRM has been widely applied in the field of NIA. To deal with the poor prediction quality in the CF, Aljunid and Huchaiah proposed an innovative solution. They combined user and project-based CF models with Γ-linear regression models to deeply model the sparsity and scalability of user and item rating matrices. This integrated model demonstrated significant performance advantages [6]. To alleviate the difficulties consumers, encounter when searching for projects that meet their own needs, Papadakis et al. conducted a comprehensive review of various methods in the field of CF recommendation systems. They compared and evaluated these methods based on the CF system's ability to effectively respond to known defects [7]. Ajaegbu optimized the traditional project-based similarity measurement of CF to address the effectiveness issue of insufficient user numbers or lack of user rating records. This algorithm not only effectively reduced the shortcomings of traditional algorithms in data sparse or cold start scenarios, but also retained the advantages of existing project-based CF algorithms [8]. To handle the sparsity encountered by CF in recommendation systems about e-commerce, Koohi and Kiani proposed to fully utilize user rating information to overcome the challenge without changing the data dimension. This method could effectively improve the performance of recommendation systems [9]. Traditional recommendation systems mainly target individual users for project recommendations, but in group recommendation scenarios, existing methods often overlook the importance of metadata information when predicting rating information. Yannam et al. combined multi-layer perceptrons and general matrix factorization techniques, and fully utilized metadata and neural CF techniques. This method could effectively solve the cold start problem in group recommendation scenarios [10].

In recent years, the NIA model has played an important role in multiple fields. Due to the excellent representational ability of neural networks, significant progress has been made in the development of Neural Recommendation Models (NRM), which have surpassed and improved traditional recommendation models. Wu et al. conducted a comprehensive and systematic review of NRM from the perspective of recommendation modeling. They reviewed various representative research works and finally explored the development direction of this field, providing valuable references for future research [11]. The traditional CF algorithm mainly relied on rating or attribute information, ignoring the hierarchical structure of the data, which led to sparsity and timeliness issues in the recommendation matrix. Chen et al. proposed a dynamic clustering CF-RA combined with a double-layer network, and its effectiveness has been fully demonstrated [12]. When the evaluation matrix between users and items exhibited sparsity, traditional NIA models often struggled to play an effective role. To address this issue, Manochandar and Punniyamoorthy proposed an improved method for measuring popularity similarity. This method achieved accurate prediction of available and unavailable ratings by comprehensively considering the relevant components of users and products. Compared to traditional methods, this solution performed better [13]. The general CF was difficult to capture complex interaction information in sparse Mashup Web service call matrices, resulting in poor recommendation performance. Liang et al. designed a secure CF service RA that integrates content similarity, in which the secure CF module is specifically used to address these matters. This algorithm could still efficiently perform web service recommendation tasks even in sparse data [14]. To address the cold start and data sparsity faced by NIA models, Anwar et al. suggested a memory-based method that executes CF by generating user item similarity matrices and prediction matrices. This method could effectively handle the sparsity and cold start, and recommend more related projects to users [15].

In summary, the CF algorithm, as an important tool of NIA, is widely applied in fields like e-commerce and social media. Nevertheless, the expansion of network data and the diversification of user requirements have increasingly complicated the functioning of this algorithm. For new users or items, it is difficult for RAs to make accurate recommendations due to the lack of historical data. Therefore, this study proposes an improved CF-RA to overcome the limitations of existing algorithms and improve the application effectiveness of the NIA model. The research is compared with methods in existing literature, as shown in Table 1.

Table 1: Literature review summary

Reference number	User-item matrix processing	Sparsity processing	Recommended performance	Contribution
[6]	Γ linear regression model	In-depth modeling	Performance advantage	Associative statistical model
[7]	Method combing	/	/	Comprehensive assessment
[8]	Optimized similarity measure	Reduce data sparsity	/	Optimizing traditional algorithm
[9]	User rating information	Overcome the sparsity challenge	Improve performance	No need to change the data dimension
[10]	Multi-layer perceptron + matrix decomposition	/	/	Group recommendation scenario
[11]	NRM	/	/	Systematic review
[12]	Two-layer network clustering	/	Proof of validity	Dynamic clustering
[13]	Improved similarity measures	Accurate prediction	Superior performance	The relationship between users and goods
[14]	Secure CF	Efficient execution	/	Capture complex interactions
[15]	Memory mechanism	Solve cold start and sparsity	Recommend related projects	Generating similarity matrix
Research and extraction method	Matrix dimensionality reduction based on singular value decomposition	Improve performance, error control below 0.5	The accuracy rate and recall rate are above 0.9	Achieve accurate user behavior prediction and personalized recommendation

As illustrated in Table 1, the value of the research lies in the utilization of matrix dimensionality reduction technology, the reduction of errors through iterative optimization, and the enhancement of model structure to achieve higher accuracy and recall rates, thereby providing a novel NIA model with enhanced prediction accuracy and practical application value.

3 NIA based on MDRM-CF recommendation algorithm

This study first introduces a Personalized Recommendation System (PRS) on the grounds of the CF and provides a specific analysis. Next, a MDRM on the basis of Singular Value Decomposition (SVD) is proposed, and finally a heterogeneous NIA model is constructed.

3.1 PRS based on CF recommendation algorithm

PRS, as an auxiliary tool, provides users with targeted recommendations by analyzing their characteristics and preferences for projects. PRS first collects user historical behavioral data, and then analyzes this data to extract user interests, preferences, and features [16]. The working principle of PRS is Figure 1.

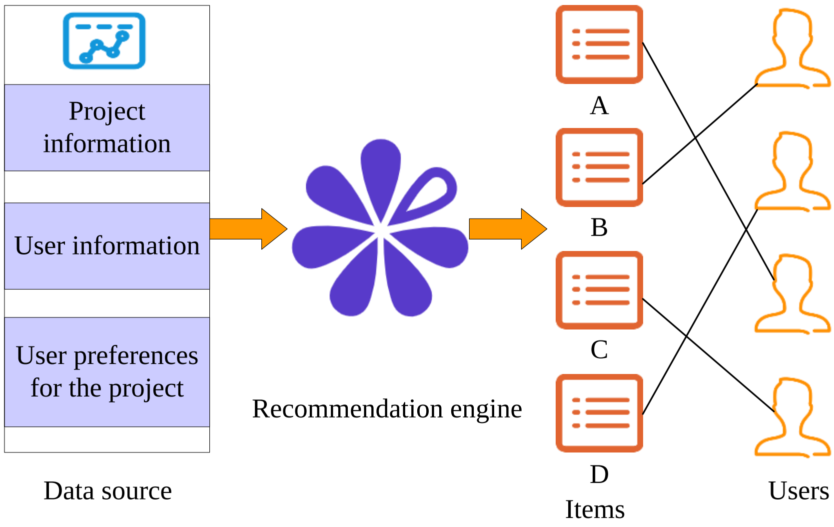


Figure 1: Personalized recommendation system works

In Figure 1, the top layer displays the information source of the item, which is not only the input information of the recommendation engine, but also the foundation of the entire recommendation system. The middle layer is the recommendation engine, which is the core part of the entire PRS. One of the simplest and most common similarity calculation methods in CF-based recommendation systems is Euclidean Distance (ED). ED is a method of measuring the straight-line distance between two points in a multi-dimensional space. Equation (1) is used in the context of CF to calculate the similarity between users or items.

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{1}$$

In equation (1), d represents ED. x_i is the i-th attribute of projects x and y. n represents the number of dimensions of an attribute. Considering the project as points in an n-dimensional vector space, and the similarity between them can be represented by the cosine of the angle between them. This method is called cosine similarity, calculated as equation (2).

$$w_{i,j}=\cos(\vec{i},\vec{j})=\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|*\|\vec{j}\|} \tag{2}$$

In equation (2), $\|\vec{i}\|$ and $\|\vec{j}\|$ are the modules of vectors i and j. $\cdot$ represents the dot product of vectors. The similarity between projects can also be represented by the linear correlation between projects. Linear correlation is usually measured using the Pearson Correlation Coefficient (PCC) for evaluating the strength and direction of the linear relationship between two variables [17]. The PCC can be utilized to calculate the linear correlation between two items, as shown in equation (3).

$$Pearson(x,y)=\frac{\sum(x,y)}{\sigma_x\times\sigma_y} \tag{3}$$

In equation (3), $\sum(x,y)$ is the covariance of items x and y. σ represents the standard deviation. In neighborhood-based CF, predicting the score of the target item is one of the core tasks of recommendation systems. Based on the user's historical behavior and rating information similar to the user or item, the potential interest of the user in unexpressed items is predicted, thereby recommending items with higher ratings to the user. For each neighbor, the system will weight their rating by similarity, and then accumulate or average the weighted ratings of all neighbors to obtain the user's predicted rating for the target item, as shown in equation (4).

$$P_{u1,i}=\bar{r}_{u1}+\frac{\sum_{u2\in I}(r_{u2,i}-\bar{r}_{u2})\cdot w_{u1,u2}}{\sum_{u2\in I}|w_{u1,u2}|} \tag{4}$$

In equation (4), $w_{u1,u2}$ represents the similarity between users. $P_{u1,i}$ represents the user's predicted rating. $\bar{r}_{u1}$ and $\bar{r}_{u2}$ are the average ratings of user 1 and user 2 on the rated items, respectively. In recommendation systems, to quantitatively evaluate the accuracy of user predictions, this study uses Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate the performance of PRS. RMSE is the square root of the average square of the difference between the predicted and actual score, which measures the degree of deviation between the two values [18]. The calculation formula for RMSE is equation (5).

$$RMSE=\frac{\sqrt{\sum_{(u,i)\in Test}(r_{ui}-\hat{r}_{ui})^2}}{|Test|} \tag{5}$$

In equation (5), r_{ui} and $\hat{r}_{ui}$ represent the true and predicted ratings of user u on item i, while $Test$ represents the test set. $|Test|$ is the gross user ratings in $Test$. The average deviation between the predicted and actual values is measured by the average absolute difference between the MAE predicted score and the actual score, with equal weight given to all errors. The formula for MAE is equation (6).

$$MAE = \frac{\sum_{(u,i)\in Test} |r_{ui} - \hat{r}_{ui}|}{|Test|} \quad (6)$$

PRS based on CF-RA achieves personalized information recommendation for users through user behavior analysis, content collection, model training, and the application of CF-RA. Figure 2 is the PRS process based on CF.

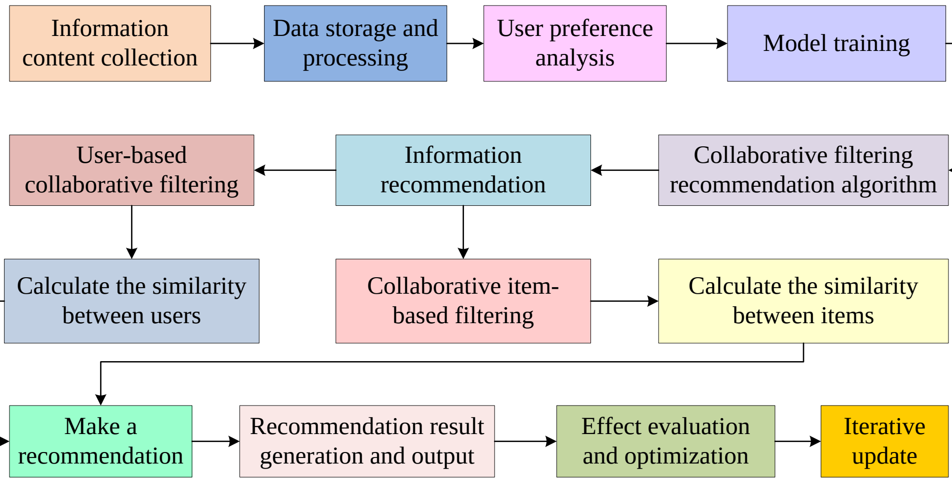


Figure 2: PRS flow based on CF algorithm

In Figure 2, firstly, the system will collect information content, use web scraping software to capture information, and automatically extract keywords for information recommendation. Secondly, based on the user's operating history, the system will analyze an interest model that can predict user preferences. Finally, the system will recommend information based on the CF-RA.

3.2 MDRM Based on SVD

This study analyzes PRS based on the CF algorithm. A large amount of information data requires dimensionality reduction processing. The traditional matrix dimensionality reduction method is SVD, which decomposes the matrix by solving for eigenvalues and eigenvectors. SVD not only preserves the important features of the matrix, but also reduces the dimensionality of the matrix, making the calculation of the matrix more convenient [19]. Firstly, to define a matrix of MN, it is necessary to decompose it as shown in equation (7).

$$R = USV^T \quad (7)$$

In equation (7), U is a M-order square matrix, and its column vectors, i.e. the left singular vectors, are orthogonal to each other. S is a MN matrix, with all elements except for the diagonal being 0, and the elements on the diagonal are called singular values. V^T is a transpose of V. V is a N-order matrix, and its row vectors, i.e. the right singular vector, are also orthogonal to each other. By such decomposition, the SVD form of the matrix can be obtained. Figure 3 shows the matrix factorization process.

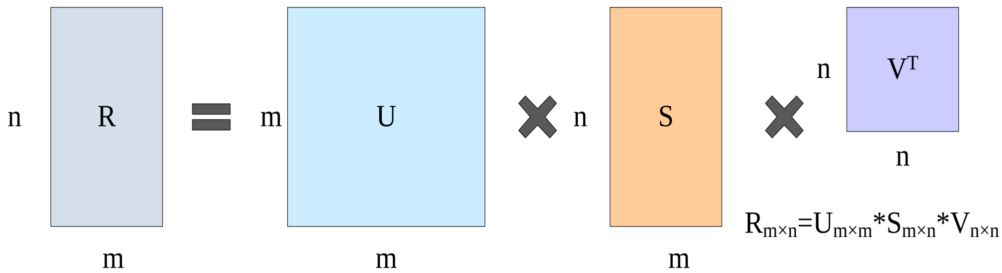


Figure 3: Matrix decomposition procedure

In Figure 3, matrix factorization is the process of decomposing a matrix into products of several standard matrices. Then, dimensionality reduction is performed on the decomposed matrix mentioned above. Transforming matrix R to obtain R^T. Multiplying R^T by R to obtain a new square matrix. The eigenvalue formula is used to solve the eigenvalues and corresponding eigenvectors of a square matrix, as expressed in equation (8).

$$(R^T * R) * v_i = \lambda v_i \tag{8}$$

In equation (8), λ is the eigenvalue and v_i is the eigenvector. The obtained eigenvalues and eigenvectors are operated on, and singular values and left singular vectors are calculated using formulas. In a matrix, singular values are usually arranged in descending order. In this way, the originally higher dimensional matrix is approximated as a lower dimensional matrix, achieving the effect of dimensionality reduction, as shown in equation (9).

$$\begin{cases} \sigma_i = \sqrt{\lambda_i} \\ u_i = \dfrac{1}{\sigma_i} R v_i \end{cases} \tag{9}$$

In equation (9), σ represents singular value and u represents left singular vector. In machine learning to reduce computational costs, modeling is usually achieved by iteratively fitting the raw data to obtain a good training model. Fitting is one of the key steps in building a model, and regression is a commonly used fitting method. When using linear regression in modeling, using $x_1, x_2, ..., x_n$ to describe the characteristic components of each independent variable. The estimation function of the linear regression model is equation (10).

$$f(x) = f_0(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{10}$$

In equation (10), θ represents the weight impact of each component on the estimation function. By using the least squares method, the values of these regression coefficients can be estimated, resulting in a linear regression model. This model can be used to predict new data points or explain the degree of influence of the independent variable on the dependent variable. The cost function is defined as equation (11).

$$H(\theta) = \frac{1}{2} \sum_{i=1}^{n} (f_0(x_i) - y_i)^2 \tag{11}$$

In equation (11), n is the number of samples. y_i is the actual value of sample i. $f_0(x_i)$ is the predicted value of the model for the i-th sample θ is a parameter vector containing regression coefficients. To minimize the cost function, gradient descent method is used for iterative optimization. Gradient descent is an optimization algorithm that searches along the opposite direction of the function gradient to find the local min-value of the function. In the gradient descent method, the parameter θ is first randomly initialized, and then the gradient of the cost function $H(\theta)$ with respect to θ is calculated [20]. The process of gradient descent method is Figure 4.

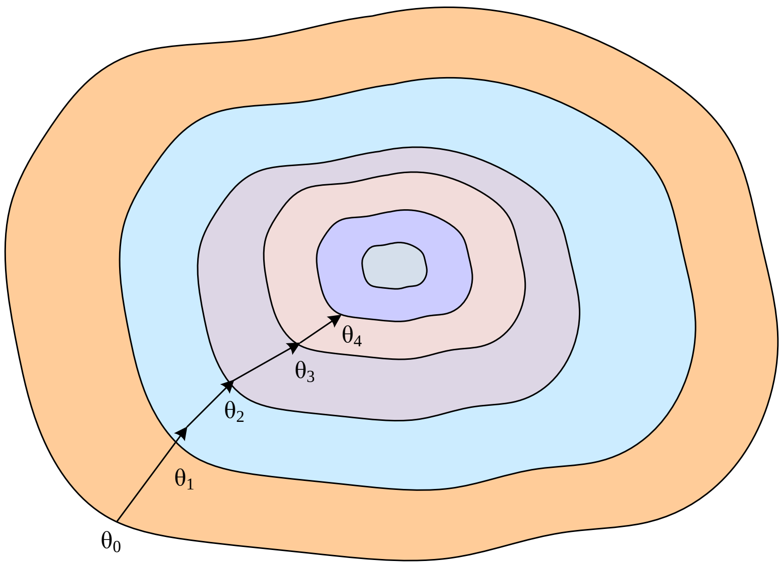


Figure 4: Gradient descent process

In Figure 4, the gradient direction represents the direction where the function grows the fastest, while the negative gradient direction is the direction where the function decreases. Parameter θ along the negative gradient direction is updated to gradually reduce the value of the cost function.

3.3 Construction of Heterogeneous NIA Model Based on CF Algorithm

After performing dimensionality reduction on the data in this study, a NIA model is constructed. An information network is a network structure formed by connecting information objects through their relationships. In this network, each information entity is a node, and the relationships between entities constitute the edges of the network. When there is more than one type of information object in an information network, it is called Heterogeneous Information Network (HIN). In HIN, nodes typically belong to different information categories [21]. The topology of HIN is Figure 5.

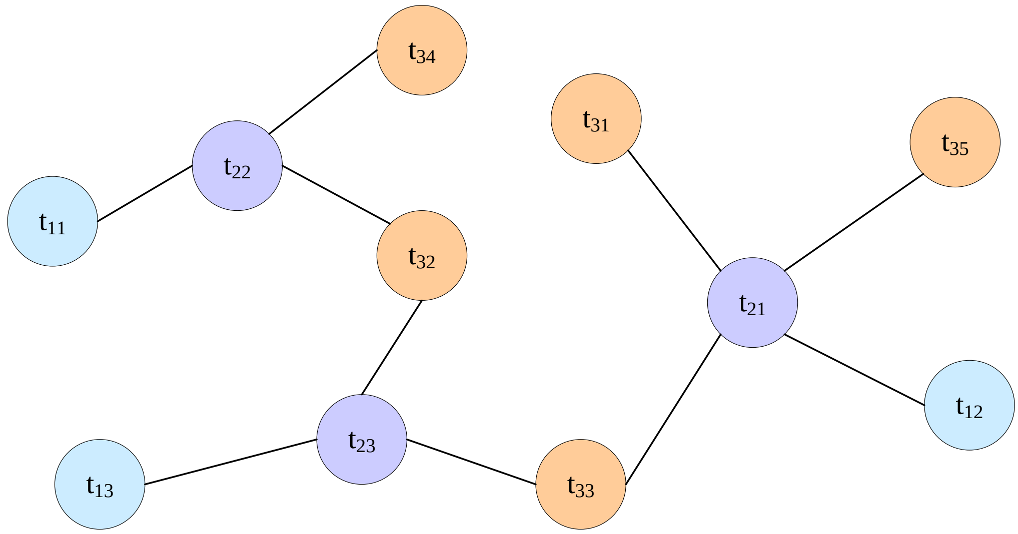


Figure 5: Information network topology

In Figure 5, there are three types of nodes, namely $T_1(t_{11},t_{12},t_{13})$, $T_1(t_{11},t_{12},t_{13})$, and $T_1(t_{11},t_{12},t_{13})$, where t_{ij} represents the object of the j-th node of node T_i. Three different types of nodes are connected to each other based on specific association relationships. There is no direct connection between objects of type T_1 and T_3, and their connection is achieved through objects of type T_2 as intermediaries. In heterogeneous service networks, network nodes are not just single type entities, but are composed of multiple objects, including Services (S), Service Providers (P), Service Requesters (R), and Service Tags (T). Figure 6 shows the extraction process of heterogeneous service networks.

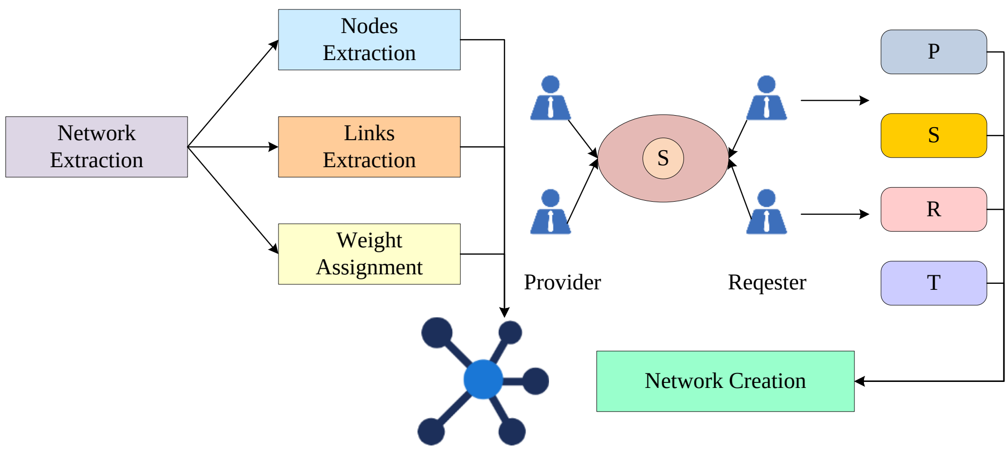


Figure 6: Heterogeneous service network extraction process

In Figure 6, first, it is necessary to identify and extract all nodes, namely S, P, R, and T, from the service request record. After extracting the nodes, it is necessary to further analyze the interactions and associations between these nodes to determine their relationships. The relationship between P and S is equation (12).

$$Provider-Service=\{e \mid e=<p,s>, p\in P, s\in S\} \quad (12)$$

In equation (12), p represents provider. s represents service, and e represents the relationship between label t and service s. If there is a P relationship, e is an edge in the network. The relationship between R and S is equation (13).

$$Requester-Service=\{e\mid e=<r,s>,r\in R,s\in S\} \quad (13)$$

In equation (13), the relationship between T and S is equation (14).

$$Tag-Service=\{e\mid e=<t,s>,t\in T,s\in S\} \quad (14)$$

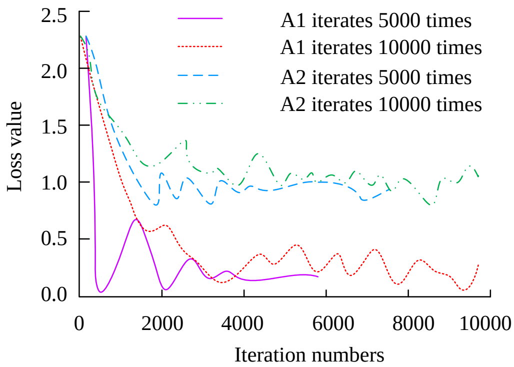
In equation (14), t represents the tag. If there is a T relationship, then e is an edge in the network. After determining the nodes and relationships, it is necessary to assign a weight to each relationship to reflect its importance or strength throughout the entire network.

4 Performance testing of NIA model based on CF algorithm

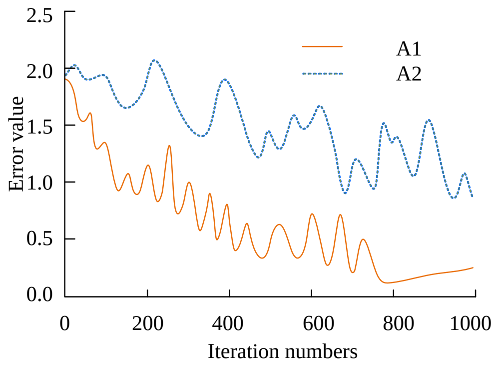
This study first tests and analyzes the NIA model based on the CF algorithm, verifies its performance by comparing and analyzing various indicators, and finally analyzes the actual application effect of the NIA model.

4.1 Index Analysis of NIA Model Based on CF Algorithm

The data set used in the experiment came from micro-blog forums, covering the hot topics on the forums and users' comments on these topics. First, the locomotive data collector is used to crawl the data on the forum. The collection strategy has been designed to align with the forum structure and page layout, with the objective of ensuring accurate and efficient access to the information required by users. The original data collected includes topic title, user ID, comment content, comment time and other fields. To construct the user-project rating matrix, the review content is quantified and divided into five rating levels according to the number of comments: 1 (less than 10 words), 2 (10-30 words), 3 (30-50 words), 4 (50-70 words), and 5 (more than 70 words). After preprocessing, a data set for the recommendation system is obtained. The dataset contains thousands of user-topic ratings, each of which reflects how much a user likes a particular topic. To evaluate the performance of the recommendation system, the data set is divided into a training set (80%) and a test set (20%). The training set is used for the model's learning and score prediction, while the test set is used to verify the model's recommendation accuracy. The experimental hardware environment uses Intel Core i7 processor, 16GB DDR4 RAM, at least 500GB SSD disk, and Windows 10 64-bit operating system. The software environment uses Java 1.7, the IDE uses Eclipse and IntelliJ IDEA, the database system uses MySQL, and the data processing tool uses JDBC. To analyze the performance of CF-RA (Algorithm 1, A1), this study compares its loss function and error with association rule-based RA (Algorithm 2, A2) at different iterations, as shown in Figure 7.



(a) Loss curve of different iterations



(b) The error of the algorithm

Figure 7: Loss function and error curve

In Figure 7(a), as the iterations increase, the loss function value gradually decreases and tends to stabilize. The algorithm has converged after 5,000 iterations, and continuing to iterate 10,000 times will not affect the loss function value. In Figure 7(b), the error of A1 stabilizes below 0.5 after 800 iterations, while the error of A2 remains high and above 1.0. This indicates that as the number of iterations increases, A1 will gradually optimize the similarity calculation between users or items, thereby more accurately finding others that are similar to the target user. This study further compares and analyzes the accuracy and recall of NIA models based on CF algorithm (M1), association rule RA (M2), graph algorithm (M3), clustering algorithm (M4), and classification algorithm (M5), as shown in Figure 8.

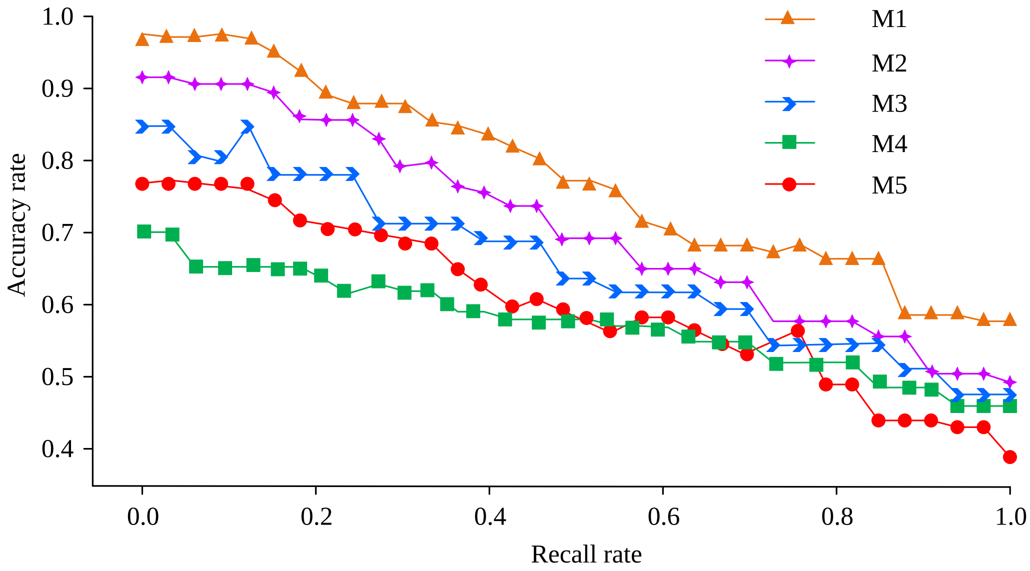


Figure 8: Accuracy and recall curve

In Figure 8, the M1's recall and accuracy are higher than the other four models, above 0.9. The accuracy and recall of the other four models are in the range of 0.6~0.9, indicating that M1 performs well in NIA tasks, with high accuracy and reliability. This study further analyzes the RMSE of the NIA based on the CF, as well as MAE, as shown in Figure 9.

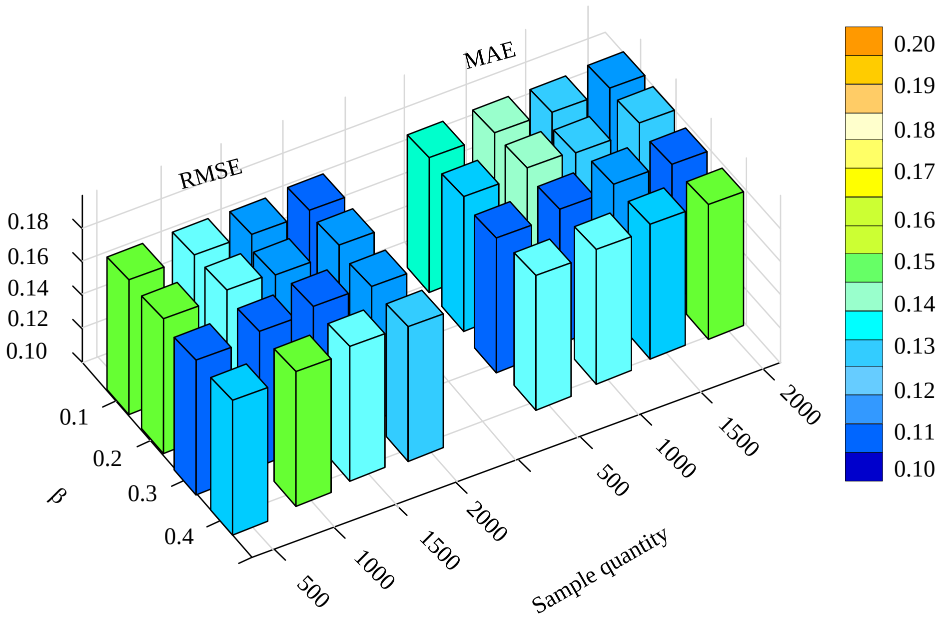


Figure 9: RMSE and MAE of the model

In Figure 9, the RMSE and MAE are both within 0.15, and when β=0.3, they are the lowest, only 0.11. This indicates that the smaller the difference between the predicted results and the actual values, the better the predictive performance. This model can give users more precise recommendations and decision support in practical applications.

4.2 Analysis of the application effect of NIA model

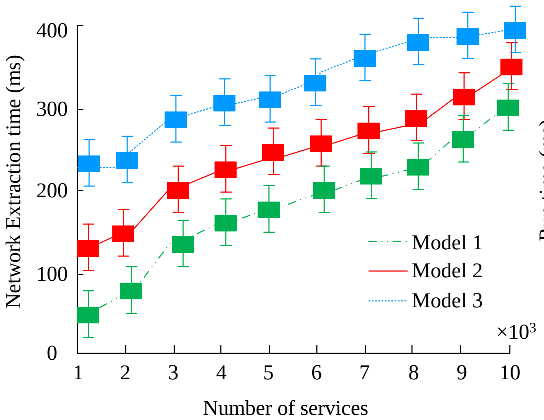
To verify the NIA effectiveness, this study introduces it into the service network maintenance mechanism. The experiment utilizes the tuple network structure extracted from the IMDB-ALL dataset to trigger network update operations by simulating changes in data in the database. The experimental design covers two types of operations:

adding and deleting, and compares the update efficiency with and without cache considerations, respectively. Table 2 shows the final results.

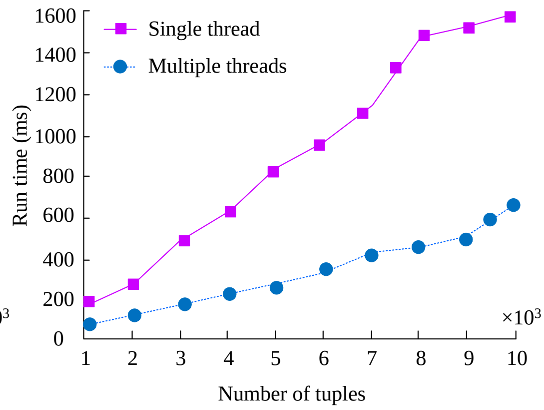
Table 2: Experimental results of network updating

Frequency of change	New update	Consider new updates to the cache	Delete update	Consider cache deletion updates
100	295	72	201	30
200	638	145	398	60
300	940	209	560	80
400	1283	294	783	120
500	1457	362	961	144
600	1825	354	1271	175
700	2093	481	1386	207
800	2648	493	1528	249
900	2866	635	1829	268
1000	3220	694	1951	301

In Table 2, network update maintenance using caching mechanism is more efficient than direct update maintenance. Batch processing not only reduces the number of database connections, but also reduces the burden on the database. Meanwhile, the caching mechanism effectively avoids frequent operations in tuple networks, thereby reducing the burden on processes. To verify the practical application performance in service networks, this study conducts experimental analysis from two dimensions: time complexity and network extraction time. In the experiment, 10 sets of service data and their related publication and call records are randomly selected from the big dataset for service network extraction testing. The entire experiment is repeated 20 times, and the ultimate result obtained at last is the mean value. The comparison between time complexity and network extraction time is Figure 10.



(a) Time complexity



(b) Network extraction time

Figure 10: Comparison of time complexity and network extraction time

In Figure 10(a), as the services grow, the time required for service network extraction also shows a nearly linear trend, ultimately reaching 320 ms. Compared to M2 and M3, M1 has lower time complexity. In Figure 10(b), the results of the multi-threaded network extraction indicate that the network extraction efficiency has significantly improved after considering multi-threaded processing, saving the time cost of tuple network extraction, especially in the case of large data volumes. This study further compares the actual application prediction results of five models, as shown in Figure 11.

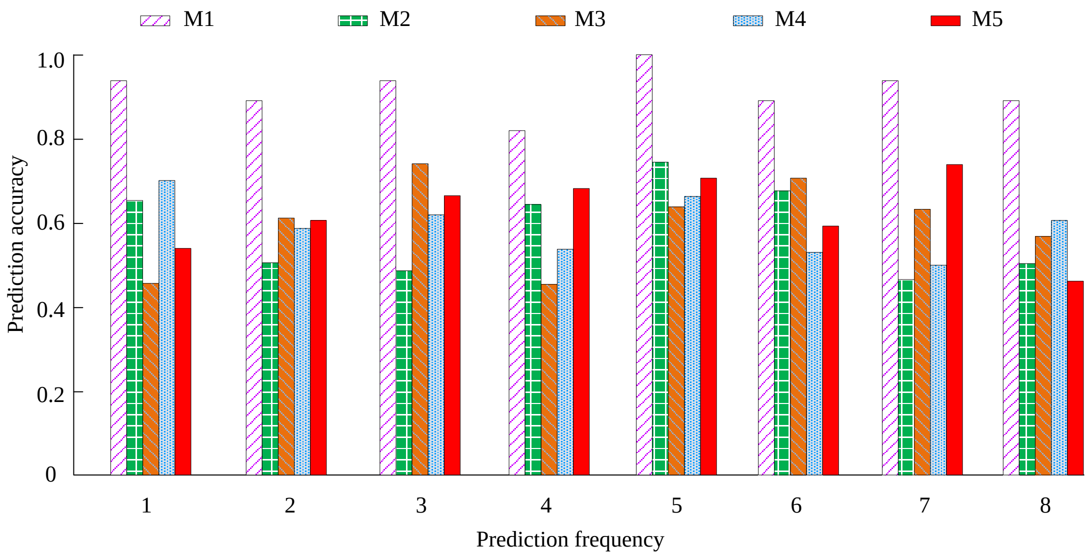


Figure 11: The practical application of five models predicted results

In Figure 11, M1 has the highest prediction accuracy, with an average accuracy of 92%, which outperforms the other four. Therefore, the CF-based NIA has shown excellent performance in network data prediction. This indicates that under the same test dataset and scenario, the model has stronger predictive ability, which means it can more accurately capture the characteristics and patterns of network data, thus making more accurate predictions. Finally, the proposed NIA model is compared with the State Of The Art (SOTA) mentioned in the literature review. Comparison indicators include RMSE, MAE, Mean Absolute Percentage Error (MAPE), Accuracy, Precision, Recall and F1 Score. All indicators are normalized, as shown in Table 3.

Table 3: Comparison of SOTA method indexes

Model index	RMSE	MAE	MAPE	Accuracy	Precision	Recall	F1 Score
Reference [6]	0.93	0.95	0.82	0.77	0.79	0.82	0.85
Reference [8]	0.73	0.71	0.64	0.85	0.88	0.89	0.74
Reference [12]	0.86	0.88	0.63	0.74	0.91	0.89	0.82
Reference [13]	0.97	0.95	0.83	0.81	0.85	0.87	0.89
Reference [14]	0.82	0.83	0.79	0.71	0.76	0.75	0.72
Reference [15]	0.68	0.72	0.75	0.78	0.89	0.93	0.85
Model 1	0.91	0.89	0.93	0.94	0.90	0.95	0.92

In Table 3, RMSE, MAE, MAPE, Accuracy, Precision, Recall and F1 Score of the NIA model proposed in this study are 0.91, 0.89, 0.93, 0.94, 0.90, 0.95 and 0.92, respectively. The comprehensive comparison is better than the SOTA methods in other literature. Other SOTA methods are excellent in some indicators, but poor in other indicators. The model's performance is balanced and excellent across a range of metrics, demonstrating its versatility and reliability. The comprehensive and balanced nature of the model renders it more reliable and stable in practical application. It is capable of accurately predicting the future trend or result of network information, identifying the category or label of network information with greater precision, and thus providing users with more valuable information.

5 Conclusion

With the diversification of network information, user behavior data has become increasingly complex. Therefore, this study proposed an SVD-based MDRM and applied it to the NIA model to strengthen the efficiency and accuracy of the CF. The experiment showed that the MAE and RMSE were both within 0.15, and when β was 0.3, the two were the lowest, only 0.11. The NIA model based on the CF-RA had the highest prediction accuracy, with an average accuracy of 92%, which was higher than other network analysis models. When applying the model to a service network, as the number of services increased, the time required for service network extraction also showed a nearly linear trend, ultimately reaching 320ms. The efficiency of network extraction has significantly improved after considering multi-threaded processing, saving the time cost of tuple network extraction, especially in situations with large amounts of data. This model exhibits good performance in processing large-scale datasets and can push more accurate individualized recommendations to users. This study mainly focuses on predicting user behavior and personalized recommendations. In the future, more factors such as user social relationships and contextual information can be considered in the model to provide more comprehensive recommendation services.

References

[1] Umair Javed, Kamran Shaukat, Ibrahim A. Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. A review of content-based and context-based recommendation systems. International journal of emerging technologies in learning, 16(3):274-306, 2021. https://doi.org/10.3991/ijet.v16i03.18851

[2] Nisha S. Raj, and V. G. Renumol. A systematic literature review on adaptive content recommenders in personalized learning environments from 2015 to 2020. Journal of computers in education, 9(1):113-148, 2022. https://doi.org/10.1007/s40692-021-00199-4

[3] Fethi Fkih. Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. Journal of king saud university-computer and information sciences, 34(9):7645-7669, 2022. https://doi.org/10.1016/j.jksuci.2021.09.014

[4] Taushif Anwar, V. Uma, Md. Imran Hussain, and Muralidhar Pantula. Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: A comparative analysis. Multimedia tools and applications, 81(25):35693-35711, 2022. https://doi.org/10.1007/s11042-021-11883-z

[5] Bo Jiang, Junchen Yang, Yanbin Qin, Tian Wang, Muchou Wang, and Weifeng Pan. A service recommendation algorithm based on knowledge graph and collaborative filtering. IEEE access, 9(1):50880-50892, 2021. https://doi.org/10.1109/ACCESS.2021.3068570

[6] Mohammed Fadhel Aljunid, and Manjaiah Doddaghatta Huchaiah. An efficient hybrid recommendation model based on collaborative filtering recommender systems. CAAI transactions on intelligence technology, 6(4):480-492, 2021. https://doi.org/10.1049/cit2.12048

[7] Harris Papadakis, Antonis Papagrigoriou, Costas Panagiotakis, Eleftherios Kosmas, and Paraskevi Fragopoulou. Collaborative filtering recommender systems taxonomy. Knowledge and information systems, 64(1):35-74, 2022. https://doi.org/10.1007/s10115-021-01628-7

[8] Chigozirim Ajaegbu. An optimized item-based collaborative filtering algorithm. Journal of ambient intelligence and humanized computing, 12(12):10629-10636, 2021. https://doi.org/10.1007/s12652-020-02876-1

[9] Hamidreza Koohi, and Kourosh Kiani. Two new collaborative filtering approaches to solve the sparsity problem. Cluster computing, 24(1):753-765, 2021. https://doi.org/10.1007/s10586-020-03155-6

[10] Fan Yang. Optimization of personalized recommendation strategy for E-commerce platform based on artificial intelligence. Informatic, 47(2):235-242, 2023. https://doi.org/10.31449/inf.v47i2.3981

[11] Wu Le, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. IEEE transactions on knowledge and data engineering, 35(5):4425-4445, 2022. https://doi.org/10.1109/TKDE.2022.3145690

[12] Jianrui Chen, Bo Wang, Zhi** Ouyang, and Zhihui Wang. Dynamic clustering collaborative filtering recommendation algorithm based on double-layer network. International journal of machine learning and cybernetics, 12(4):1097-1113, 2021. https://doi.org/10.1007/s13042-020-01223-2

[13] S. Manochandar, and M. Punniyamoorthy. A new user similarity measure in a new prediction model for collaborative filtering. Applied intelligence, 51(1):586-615, 2021. https://doi.org/10.1007/s10489-020-01811-3

[14] Wei Liang, Songyou Xie, Jiahong Cai, Jianbo Xu, Yupeng Hu, Yang Xu, and Meikang Qiu. Deep neural network security collaborative filtering scheme for service recommendation in intelligent cyber-physical systems. IEEE internet of things journal, 9(22):22123-22132, 2021. https://doi.org/10.1109/JIOT.2021.3086845

[15] Taushif Anwar, V. Uma, Md. Imran Hussain, and Muralidhar Pantula. Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: A comparative analysis. Multimedia tools and applications, 81(25):35693-35711, 2022. https://doi.org/10.1007/s11042-021-11883-z

[16] Armielle Noulapeu Ngaffo, and Zièd Choukair. A deep neural network-based collaborative filtering using a matrix factorization with a twofold regularization. Neural computing and applications, 34(9):6991-7003, 2022. https://doi.org/10.1007/s00521-021-06831-9

[17] Gopal Behera, and Neeta Nain. DeepNNMF: Deep nonlinear non-negative matrix factorization to address sparsity problem of collaborative recommender system. International journal of information technology, 14(7):3637-3645, 2022. https://doi.org/10.1007/s41870-022-00982-1

[18] Jiajia Chen, Xin Xin, Xianfeng Liang, Xiangnan He, and Jun Liu. GDSRec: Graph-based decentralized collaborative filtering for social recommendation. IEEE transactions on knowledge and data engineering, 35(5):4813-4824, 2022. https://doi.org/10.1109/TKDE.2022.3153284

[19] Urvish Thakker, Ruhi Patel, and Manan Shah. A comprehensive analysis on movie recommendation system employing collaborative filtering. Multimedia tools and applications, 80(19):28647-28672, 2021. https://doi.org/10.1007/s11042-021-10965-2

[20] Fan Wang, Haibin Zhu, Gautam Srivastava, Shancang Li, Mohammad R. Khosravi, and Lianyong Qi. Robust collaborative filtering recommendation with user-item-trust records. IEEE transactions on computational social systems, 9(4):986-996, 2021. https://doi.org/10.1109/TCSS.2021.3064213

[21] Sandipan Choudhuri, Suli Adeniye, and Arunabha Sen. Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation. Artificial intelligence and applications, 1(1):43-51, 2023. https://doi.org/10.47852/bonviewAIA2202524