Personalized Recommendation Algorithm Based on Data Mining and Multi-objective Immune Optimization

Zhigang Zhu
City Institute, Dalian University of Technology, Dalian 116600, China
E-mail: zhuzhigang19841011@126.com

Keywords: data mining, multi-objective optimization, individualization, recommendation, PCC

Received: July 3, 2024

To improve the accuracy of recommendation systems and user satisfaction, a personalized recommendation method combining data mining technology, convolutional neural network and multi-objective immune optimization algorithm is proposed in this paper. First, Pearson correlation coefficient is used to reduce the subjective bias of user ratings. Then, the convolutional neural network model is used to capture the long-term behavior pattern of users, extract deep interest features, and reduce the complexity of the model through ResNet connection. Finally, a multi-objective immune optimization algorithm is used to strike a balance between recommendation accuracy and diversity. The experiment was carried out on three datasets: MovieLens, Donation Dashboard, and Netflix. Compared with traditional algorithms, the average accuracy of the research algorithm on the three datasets was improved to 95.2%, and the root-mean-square error was less than 0.04. In addition, through multi-objective immune optimization, the algorithm significantly enhanced the recommendation diversity, with a hit rate of 0.3781 on the NetfAix dataset and a normalized discounted cumulative gain of 0.2349. The algorithm achieved stable performance in less iterations, and the recall rate was improved to 85%-95%, which was far better than other algorithms. The research method significantly improves the hit rate and normalized discounted cumulative gain value of recommendation results, providing users with more personalized resources

Povzetek: Razvit je personaliziran algoritem za priporočanje, ki združuje podatkovno rudarjenje in večciljno optimizacijo. Z uporabo Pearsonovega korelacijskega koeficienta, nevronskih mrež in imunskega algoritma izboljša kvaliteto in raznolikost priporočil, kar povečuje zadovoljstvo uporabnikov.

1 Introduction

With the rapid expansion of Internet information, the shortcomings of search engines are more obvious. This also makes it hard for users to quickly obtain the content they want, exacerbating the "information overload". The massive amount of news information has brought difficulties to users' choices. Effectively filtering out content that users are truly interested in is an urgent problem that needs to be solved [1-2]. In personalized recommendation systems, Collaborative Filtering (CF) algorithm is an important method, which includes two types: user-based CF and item-based CF. However, the former has low recommendation accuracy when data are sparse, while item-based CF may lead to overly single recommended content, ignoring the personalized characteristics of users [3-4]. In view of this, more researchers are paying attention to various new recommendation algorithms.

Ganesh and Velu built a movie recommendation system based on the Probabilistic Matrix Factorization. The experiment showed that the algorithm had good recommendation accuracy [5]. Bhaskaran and Marappan designed an enhanced vector space recommendation system. The system extracted information from the server through classification learning. The improved content-based filtering method was used to calculate similarity and generate more accurate recommendation lists. The results showed that the Mean Absolute Error (MAE) of the model was improved by 5.08%~25.26%, and the accuracy was 80%~93% [6]. Duan et al. proposed the ETBRec algorithm, which combined user trust differences and the influence of expert users to improve CF recommendations. The trust measure was divided into direct and indirect trust and took into account the direct impact of expert users on ratings. Experiments were conducted on bridge and Douban datasets. The results showed that ETBRec outperformed other recommended algorithms on indicators such as MAE and Root-Mean-Square Error (RMSE) [7]. To accurately predict flight delays, Shao et al. built a flight prediction model on the basis of trajectory mining technology through various vehicle trajectories and related sensor data on the airport apron. The simulation results showed that the error rate did not exceed 3% [8].

Wang et al. designed a graph neural network based on hyper-edges for cognitive radio to solve the course recommendation without understanding the correlation between learners [9]. Hasan and Ferdous proposed a hybrid movie recommendation system, which used Alternating Least Squares (ALS) algorithm to enhance recommendation accuracy by integrating text-to-number conversion and cosine similarity methods. The results showed that the RMSE of the system in the first experiment was 0.97613, and the RMSE was reduced to 0.8951 when expanded to 4800 movies in the second experiment [10]. To provide personalized recommendations, Chang et al. designed a recommendation mechanism that integrated multiple attributes and social network analysis methods to optimize online travel booking and meet the different needs of tourists. The K-means algorithm was used to identify specific tourism recommendation problems. Experiments showed that the research method could improve tourist booking satisfaction and make more accurate travel choices [11]. Ma et al. proposed a recommendation algorithm that combined kernel density estimation technology and multi-objective optimization to solve common problems such as low new user engagement, limited recommendations, and lack of data. Multiple objectives were considered to optimize the accuracy and richness of recommendations. The research method had a 5.6% improvement in accuracy compared with traditional methods [12]. In summary, the relevant research methods are shown in Table 1.

Table 1: Advantages and disadvantages of each research method

Year	Author	Research method	Advantage	Shortcoming
2022	Ganesh and Velu [5]	Comparison between support vector regression algorithm and matrix decomposition	The accuracy of the movie recommendation system is improved	The method is complicated and the calculation cost is high
2023	Bhaskaran and Marappan [6]	Enhanced vector space recommendation system with improved content-based filtering methods	The average absolute error is reduced, and the recommendation accuracy is high	The dependence on classification learning style is higher
2022	Duan et al. [7]	ETBRec algorithm combines user trust and expert user influence for collaborative filtering and recommendation	Excellent performance in trust measurement and MAE and RMSE indicators	It relies heavily on trust networks and expert users
2022	Shao et al. [8]	Flight prediction model based on trajectory mining technology	Low error rate and high prediction accuracy	The application scenarios are limited and may not apply to other types of data
2022	Wang et al. [9]	Graph neural networks based on SuperEdge are used for MOOC course recommendation	High accuracy, especially for MOOC course recommendations	Data sets are highly dependent and may have low applicability
2024	Hasan and Ferdous [10]	Alternating least squares combines text-to-number conversion and cosine similarity methods	The RMSE is reduced and the recommendation accuracy is significantly improved	The computational complexity is high in large-scale data processing
2022	Chang et al. [11]	Hybrid recommendation methods, combined with collaborative filtering and social network analysis	Improved recommendation accuracy and satisfaction for online travel bookings	Social network analysis relies on the integrity of user data and may lead to privacy issues

2023	Ma et al. [12]	Recommendation algorithm based on multi-objective optimization combined with kernel density estimation technique	Excellent performance in recommendation accuracy and richness	This can be challenging for new users or when data is scarce

In summary, recommendation systems tend to display content that is similar to the user's previous preferences, which may limit content diversity and make it difficult to access new information. Newly added users or products lack historical data, making it difficult to obtain initial customized recommendations. In addition, the operation of algorithms is often not clear enough for users, which may affect their trust in the pushed content. Furthermore, finding the appropriate balance between relevance and diversity in recommendations is a challenge faced by algorithm developers. In summary, the above methods have two problems. First, some recommendation systems, although improving recommendation accuracy, do not fully consider recommendation diversity, which may lower the user experience. In addition, some research have shown poor performance in handling new users or sparse data, which affects the widespread applicability of recommendations [13]. Therefore, this study innovatively introduces the Pearson Correlation Coefficient (PCC) method to reduce the subjective bias of user rating standards. TOP-N method is a recommendation method that solves the unreasonable recommendation content in traditional threshold methods. Then, the Convolutional Neural Network (CNN) is used to deeply explore the deep connections between data nodes, capturing long-term user behavior patterns. Finally, combining multi-objective immune optimization algorithms, the challenge of balancing accuracy and diversity in recommendation systems is solved. In addition, the study also employs binary encoding strategy and specific genetic operations to ensure the diversity of recommendation lists and the stability of population evolution. It aims to improve the personalized recommendation accuracy and satisfaction for users through the effective multi-objective evolutionary algorithm.

2 Methods and materials

2.1 Personalized recommendation algorithm based on data mining technology

Data mining is currently a hot topic in artificial intelligence and databases. Based on data mining techniques, the accuracy of user searches can be improved, which exerts a crucial role in personalized recommendation systems. To address the information overload, an intelligent cycle recommendation based on user information data mining is developed [14-15]. It deeply analyzes the historical data of users and predicts their behavior patterns, which convert users' active queries into intelligent recommendations of the system, thereby improving the user experience. The traditional cosine similarity method is greatly influenced by individual scoring standards and may affect the objectivity of similarity. Therefore, the study adopts the PCC method to reduce this subjective bias. The PCC calculation between dual-use users is shown in equation (1).

$$P(a,b)=\frac{\sum_{I_{aa'}}\left(S_{ab}-\overline{S_{ab}}\right)\left(S_{a'b}-\overline{S_{a'}}\right)}{\sqrt{\sum_{I_{aa'}}\left(S_{ab}-\overline{S_{a}}\right)^{2}\left(S_{a'b}-\overline{S_{a'}}\right)^{2}}} \tag{1}$$

In equation (1), $P(a,b)$ signifies the PCC between users a and b. S_{ab} signifies the joint rating of users a and b on a recommended content. $\overline{S_{ab}}$ signifies the average rating of users a and b on all common rating items. $\overline{S_{a}}$ signifies the average rating of user a on all rating items. S_b signifies the average rating of user b on all rating items. $I_{aa'}$ signifies the set of items jointly rated by a and b. Traditional CF typically outputs a complete list of items sorted by predicted scores, from which users need to filter out the content they are interested in. This reduces the efficiency of finding the content they are interested in. Therefore, in order to improve user experience and satisfaction, the study chooses the TOP-N method as the recommendation method to more directly and effectively meet the personalized needs. TOP-N analyzes the user's historical behavior and preferences, and sorts these data in descending order based on similarity. The items that the user may be interested in are predicted, and these items are presented to users in a list. The improved recommendation prediction $F(a,b)$ calculation expression is displayed in equation (2).

$$\begin{cases} F(a,b)=k\sum_{a'\in A}P(a,b) \\ k=\dfrac{1}{\left|\sum_{a'\in A}P(a,b)\right|} \end{cases} \tag{2}$$

To improve the efficiency of cycle intelligent recommendation systems, this study uses data mining to analyze existing results and optimize parameters to reveal implicit relationships within data. The commonly used mining methods perform poorly in terms of computational performance, mainly due to the algorithm's failure to fully consider the long-term dependence of cycle data. To capture these long-term associations and further refine recommendations, this study introduces CNN to deeply explore the deep connections between data nodes. The process of extracting recommendation content through convolution is shown in Figure 1.

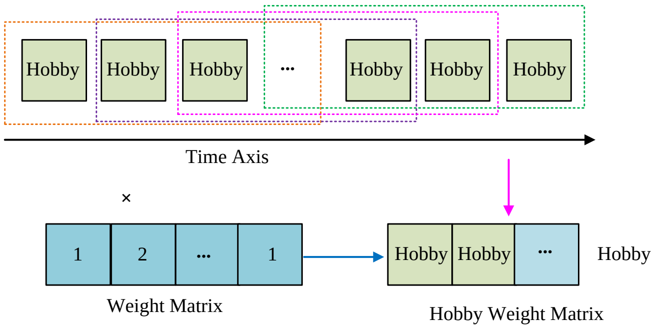


Figure 1: Recommended content for convolutional extraction

Figure 1 shows the process of generating recommendation content using CNN. Firstly, the “Hobby” data on the timeline represents user's activities or interests at different time points. Next, a 3x1 convolution kernel slides along the timeline and performs convolution operations to extract the features from this time point data. CNN uses 3×1 convolutional kernel, which is suitable for capturing short-term patterns of user behavior and avoiding overfitting problems caused by large convolutional kernel. The ReLU activation function is chosen to mitigate gradient disappearance, which is suitable for deep learning networks, helping them learn complex patterns through its nonlinear properties. ResNet connections are used to simplify model complexity and effectively solve disappearing gradients in deep networks. At this point, the convolution operation is displayed in equation (3).

$$h(i,j)=(I*K)(i,j)=\sum_m\sum_n I(m,n)K(i-m,j-n) \quad (3)$$

In equation (3), $h(i,j)$ represents the output value at position (i,j) after the convolution operation. I represents input data, namely, the “Hobby” data on the timeline, which is a two-dimensional matrix. $I(m,n)$ signifies the input data at position (m,n). K represents the convolution kernel. m and n are index variables for summation operations, used to traverse the corresponding elements of input data I and convolution kernel K. Then, ResNet is used to connect modules, as shown in equation (4).

$$[y_i]=[w_{ij}]\times[x_i]+[b_i] \quad (4)$$

In equation (4), y_i represents the output of the i-th module. w_{ij} represents the weight connecting the j-th module to the i-th module, used to control the information flow between different modules. x_i signifies the input of the i-th module. b_i signifies the bias term of the i-th module. The bias term is used to adjust the baseline level of the output, which is also learned through training. Afterwards, a max pooling operation is performed to aggregate the convolved features and select the maximum value within each region, thereby reducing dimensionality and preserving key features. The ReLU activation function is adopted, as shown in equation (5).

$$\mathrm{Re}LU(x)=\begin{cases}x, x>0\\0, x\le 0\end{cases} \quad (5)$$

Finally, based on these features, multiple fully connected network modules are combined to form a Multi-connected and Fully Connected Network (MFCR) module to output recommended content. To reduce the impact of gradient explosion, ResNet connection is introduced in the study, as shown in equation (6).

$$F(x)=f(x)+x \quad (6)$$

In equation (6), $f(x)$ represents the nonlinear transformation part in the ResNet module, which is used to extract the features of the input data. x signifies the input of the ResNet module, which is the output of the previous module. The MFCR model is shown in Figure 2.

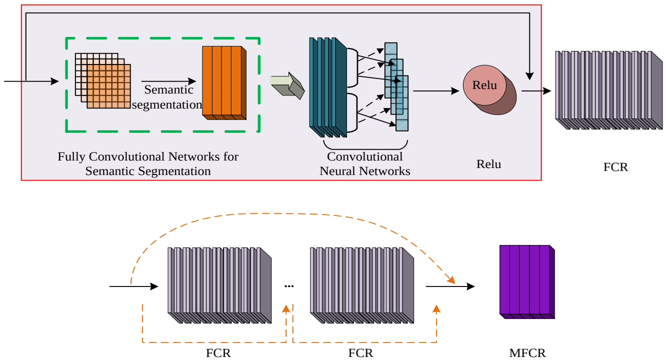


Figure 2: MFCR model

In Figure 2, a cycle recommendation strategy on the basis of user big data is proposed, called the MFCR-Big dataed Cycle Intelligent Recommendation (MFCR-BCIR) model. This model utilizes the data nodes explored by the MFCR to optimized the performance of the cycle intelligent recommendation. The steps are shown in Figure 3.

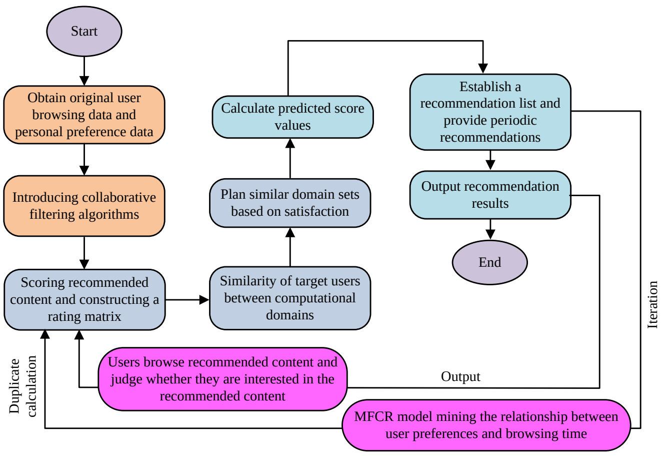


Figure 3: MFCR-BCIR algorithm flow

The first step is to obtain raw user browsing data and personal preference data. This is the foundation of recommendation systems, which understand user preferences by analyzing their historical behaviors. The second is to utilize the similarity between users and the interaction data between users and items to predict the content that users may be interested in. The similarity calculation is optimized by grey balance algorithm, as shown in equation (7).

$$B_i = \sum_{k=1}^{m} B_{ki} \tag{7}$$

In equation (7), B_{ki} represents the equilibrium closeness between data k and i. The second is to rate the recommended content and construct a rating matrix. This matrix records the user's ratings for different content, providing a quantitative basis for subsequent recommendations. The fourth is to compute the similarity of the target users in the computational domain. By analyzing user behavior patterns, user groups with similar preferences are identified. The fifth is to plan a similar domain set based on user satisfaction to provide more personalized recommendations. Then, the attention is fed back to the user, as displayed in equation (8).

$$w_{pq} = \sum_{i=1}^{n} \frac{B_i}{len(u_i)} \frac{e_{ij} \times U_i}{len(u_i)} \tag{8}$$

In equation (8), w_{pq} represents the sum of the similarity or balanced proximity between data p and q, which is used as a part of the total weight. n signifies the user or the data point considered. B_i represents a value of Boolean type. $len(u_i)$ represents the number of behaviors or the length of a certain feature of user i, used for normalization. e_{ij} represents the measurement of user i 's behavior or preference towards project i. U_i represents the activity level of user i, used to calculate attention feedback. The sixth is to calculate the predicted score value. Based on user similarity and rating matrix, how users may rate unrated content is predicted. Next, a recommendation list is established, and the cycle recommendation is provided. Based on the predicted score, a recommendation list for users is generated, which is regularly updated to reflect their latest preferences. The cycle intelligent recommendation calculation process is shown in equation (9).

$$P_{ui}=\frac{\sum sim(u,v)\times(r_{vi}-\overline{r_v})}{\sum sim(u,v)} \quad (9)$$

In equation (9), r_{vi} represents the predicted rating for the current recommendation. $\overline{r_v}$ represents the mean score. The eighth step outputs recommendation results. The optimized recommendation list is displayed to users.

2.2 Personalized recommendation method combining data mining and multi-objective immune optimization

Based on the above data mining techniques, rich user data can uncover potential user behavior patterns and preferences. To address the dilemma of balancing accuracy and diversity in recommendation results, the recommendation model is continuously optimized to maintain the timeliness and relevance of the recommended content [16]. Figure 4 shows the framework of the recommendation algorithm based on multi-objective immune optimization.

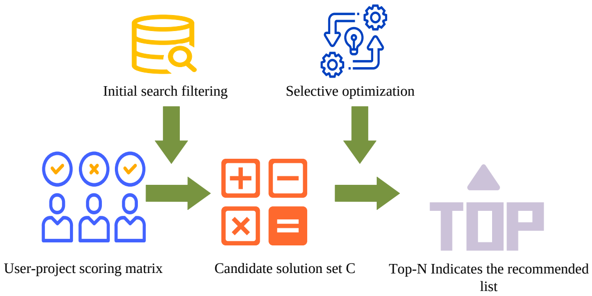


Figure 4: Multi-objective immune optimization sketch map

In Figure 4, this algorithm starts with extracting the user item rating matrix from the initial data and outputs it to the user in a recommendation list. The first step is to apply CF to estimate the score of user unrated items and generate a candidate list $|C|$ that exceeds the length of N items. The length of the candidate list satisfies $|C|>N$. The Non-dominated Neighbor Immune Algorithm (NNIA) selects the optimal Top-N recommendation item set from candidate items that meet both similarity and diversity requirements. The existing CF methods contains two types: model-based and memory-based [17]. The research focuses on the latter, whose core is to find neighboring users or item sets that are similar to the target user's behavior or preferences in all rating data, and predict user preferences. In this process, similarity measurement is a key step, taking the similarity between items to predict ratings [18-19]. To optimize the prediction accuracy, the equation (10) is obtained.

$$P_{u,i}=b_{u,i}+\frac{\sum_{j\in F} sim(i,j)(r_{u,j}-b_{u,j})}{\sum_{j\in F}(|sim(i,j)|)} \quad (10)$$

In equation (10), $b_{u,i}$ and $b_{u,j}$ represent the user's baseline prediction scores for items i and j. To optimize prediction accuracy and ensure that recommended content is highly correlated with user preferences by measuring the similarity between projects, the study uses a similarity matching function as a target, defined as equation (11).

$$f_M(P_u,R)=\frac{1}{N}\sum_{i\in R} g_m(i,P_u) \quad (11)$$

In equation (11), $f_M(P_u,R)$ is a similarity matching function used to calculate the overall similarity between the preference vector P_u of target user u and the items in the recommendation set R. Among them, N is the quantity of items in the recommended set R. $g_m(i,P_u)$ is a similarity function for a single item, which is used to measure the similarity between the preference vector P_u of item i and user u. In addition, a second function is designed to measure the diversity of recommendation lists. The specific process is shown in equation (12).

$$f_D(R)=\frac{1}{Z(Z-1)}\sum_{i\in R}\sum_{j\in R, j\neq i} d(i,j) \quad (12)$$

In equation (12), $d(i,j)$ represents a symmetric distance function, taken as $d(i,j)=1-sim(i,j)$. Z signifies the recommendation list length. Figure 5 shows the flow diagram of the NNIA.

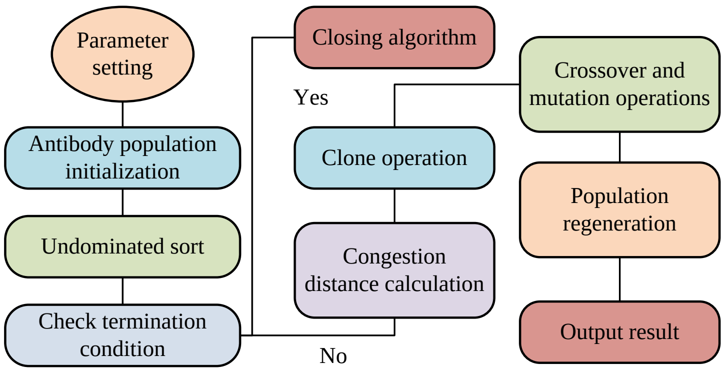


Figure 5: The flow diagram of the Non-dominated neighbor immune algorithm

In Figure 5, The process of NNIA is as follows: First, parameters (such as the maximum number of iterations, population size, etc.) are set to initialize the antibody population. The first NM individuals are then selected for non-dominated sorting. Whether the maximum number of iterations is reached is checked. If not, the crowding distance is calculated and the first NM individuals are selected. Cloning, crossover, and mutation are then performed to update the population and continue to iterate. After the termination condition is reached, the optimal solution is output. The NNIA framework is applied to address the multi-objective optimization proposed in the previous section. The specific steps are shown in Figure 6.

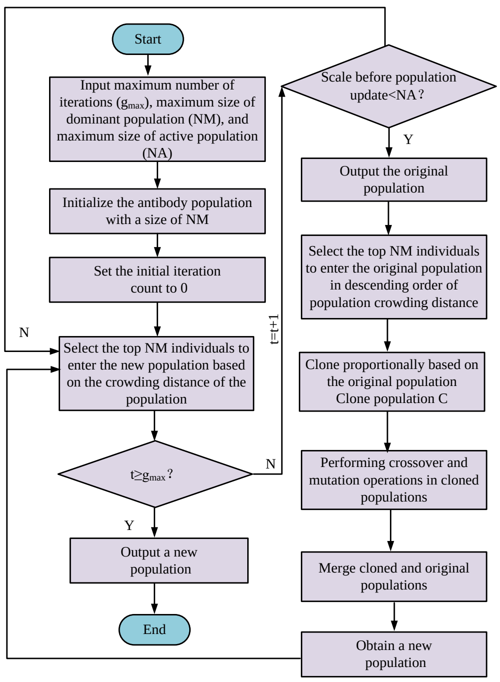


Figure 6: Multi-objective optimization solution steps based on NNIA

In Figure 6, first, these parameters are set, including the maximum iterations, the maximum size of the active population, etc. Second, the antibody population is initialized, and the population size is NM. In the multi-objective immune optimization algorithm, the cross probability is set to 0.8 to increase the exploration ability of solution space and prevent premature convergence. The variation probability is 0.1, and randomness is introduced appropriately to prevent falling into the local optimal solution. 200 iterations ensure high accuracy and diversity of models within a reasonable time. The parameter selection is experimentally tuned to ensure the best balance between performance, complexity and adaptability. Assuming the length of antibodies in the population is C, the antibody constraints for the initial and updated antibody populations are shown in equation (13).

$$\sum_{i=1}^{C} x_i = Z \qquad (13)$$

In equation (13), x_i represents a certain attribute or characteristic value of the i-th antibody. Z represents the sum of all antibody attribute values that should be satisfied after accumulation. Third, the iteration counter is set to 0, which marks the starting point of the algorithm's iteration process. Fourth, the top NM individuals are selected to enter the new population on the basis of the crowding distance of the population. Fifth, whether the current iteration count t has reached the maximum iteration count gmax is judged. If it is, the current population is output as a new population and the algorithm ends. If not, the next step is is to continue. Sixth, before updating the population, whether the current population size is below or equal to the maximum size NA of the active population is determined. If not, the algorithm goes back to step four and selects the individual again. Seventh, if it is, the original population is output. Then, depending to the descending order of population crowding distance, the top NM individuals are selected to enter the original population. The crowding distance is shown in equation (14).

$$D_{i,m} = \frac{f_{i+1,m} - f_{i-1,m}^{N}}{\max_{j-1}^{N}\left(f_{j,m}\right) - \min_{j-1}^{N}\left(f_{j,m}\right)} \qquad (14)$$

In equation (14), $f_i(x)$ represents the value of the i-th individual on the m-th objective function. N is the size of the current population. Eighth, the original population is cloned in proportion to form the cloned population C. This step helps to explore the solution space and increase population diversity. Ninth, crossover and mutation operations are performed in the cloned population C. These genetic algorithm operations help introduce new genetic mutations and promote population evolution. Finally, the cloned population is merged with the original population to form a new population. This step combines exploration (cloning and cross mutation) and development (selection) strategies [20]. The following operator based on uniform crossover is designed, as shown in Figure 7.

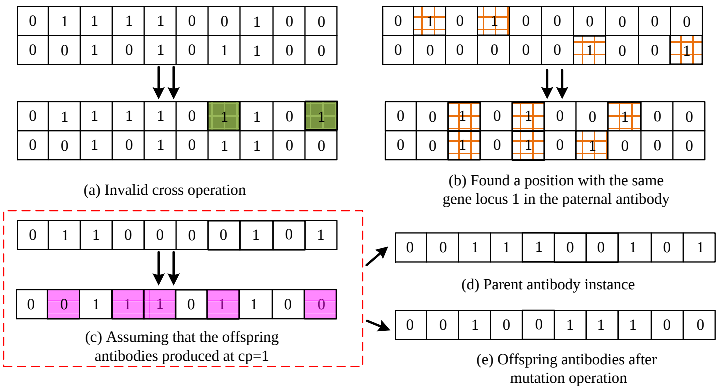


Figure 7: Mutation operation

In Figure 7, considering two paternal antibodies p_1 and p_2, all genes of the offspring are set antibody to 0 during the crossover process. When the genes of two parents in the same position are both set to 1, the corresponding offspring antibodies are also set to 1 at this position, and the gene position of the parents is 0. s is the total number of positions that meet this condition, resulting in the difference in $d = k - s$. As shown in Figure 7 (c), the same gene locus is deleted and found in the paternal parent. Then a positive integer cp is randomly generated between [1, d-1]. For p_1 and offspring antibodies c_1 and c_2, the gene loci with the first cp being 1 is selected. The corresponding c_1 position is set to 1. For p_2, the remaining $d - cp$ gene loci with a value of 1 is selected, and the corresponding c_2 position is set to 1. Two gene positions are selected, and the middle gene position is randomly rotated. As shown in Figure 7 (d), the parent generation mutated at positions 4 and 7 to produce a new antibody, as displayed in Figure 7 (e).

3 Results

On PyCharm 2021.1.3, Python 3.7 is used for the experiment. The study uses three datasets: MovieLens, Donation Dashboard, and Netflix, where the MovieLens (https://grouplens.org/datasets/movielens/latest/) and Netflix data (https://tianchi.aliyun.com/dataset/146311) contain user ratings on a scale of 1-5 stars and 1-10, respectively. In order to ensure the comparability of different scoring scales, the scoring data are standardized and the missing values are processed by means of average filling or nearest neighbor interpolation. The user-project scoring matrix is constructed based on the user's scoring history, and the data set is divided into training set and testing set according to 7:3 ratio for the training and evaluation of the model.

3.1 Simulation experiment analysis of MFCR-

BCIR algorithm

To exhibit the superiority of the MFCR-BCIR and test its performance, the experiment uses 1000 user data publicly available on the website, including Webpage browsing and interaction information, and obtains 2 million evaluation data. The experiment selects Linear Discriminant Analysis (LDA) and ARMD model as references to compare the recommendation effectiveness. Figure 8 shows the accuracy results of similarity calculation models.

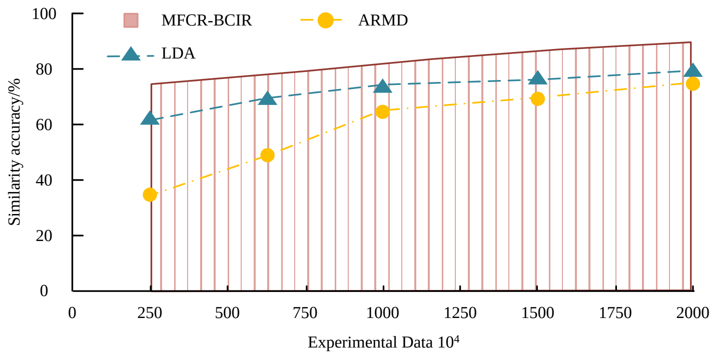


Figure 8: Comparison results of similarity calculation accuracy of three recommendation models

As shown in Figure 8, the traditional LDA algorithm performed poorly in terms of similarity accuracy, especially when the data size was small. In contrast, the ARMD algorithm showed higher stability and accuracy, but the error rate was still about 20%. The MFCR-BCIR algorithm quickly reached a steady state, and its similarity accuracy was close to 91%. The advantage of this algorithm was obvious, at least 6.9% higher in accuracy than other algorithms, greatly enhancing the recommendation accuracy. Figure 9 shows the recommendation precision and recall results of three recommendation models.

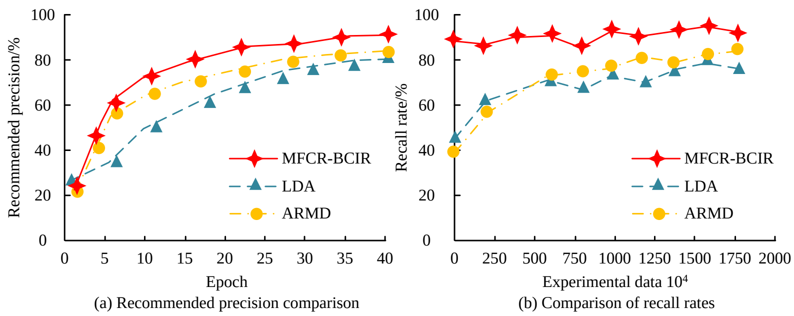


Figure 9: Results of recommendation precision and recall rate of three recommendation models

As shown in Figure 9 (a), although the recommendation precision from the three algorithms increased with epoch, the MFCR-BCIR algorithm still showed the highest prediction precision. This algorithm only required a small number of epochs to achieve relatively stable results. For the MFCR-BCIR algorithm, the precision was improved by at least 6%. As shown in Figure 9 (b), the recall rates of LDA and ARMD algorithms did not yet exceed the threshold of 80% and required a large amount of data to achieve this goal. In contrast, the recall rate of the MFCR-BCIR algorithm ranged from 85% to 95%, with a recall rate improvement of at least 7.9%.

3.2 Experimental analysis of recommendation algorithms with multi-objective immune optimization

To analyze its effectiveness, tests are conducted on three major datasets: MovieLens, Donation Dashboard, and Netflix. The MovieLens data are collected by the GroupLens team, but the experiment only uses rating data, with a rating range of 1-5. The Donation Dashboard dataset records over 59000 ratings from 3908 users for 70 products. The rating range of original dataset is [-10, 10]. The Netflix dataset is a publicly available movie and TV drama rating dataset that contains approximately 5 million rating records, involving approximately 17770 movies and 480189 users. User ratings are based on a five-star rating system, ranging from 1 star (lowest) to 5 stars (highest). The training and testing set are divided into a 7:3 ratio. Table 2 displays the settings.

Table 2: Experimental settings

Argument	Explain	Parameter value	Argument	Explain	Parameter value
NM	The dominant population is the largest	40	*pc*	Crossover probability	0.8
N	Final recommended list length	10	*pm*	Variation probability	0.1
K	Nearest neighbor number	20	*g* max	Number of iterations	200
NA	Largest active population	20	*CS*	Clonal population size	100

To optimize the reliability, the study introduces traditional Genetic Algorithm (GA), CF, and Particle Swarm Optimization (PSO) for comparison. They are tested on the training and testing sets respectively, as displayed in Figure 10.

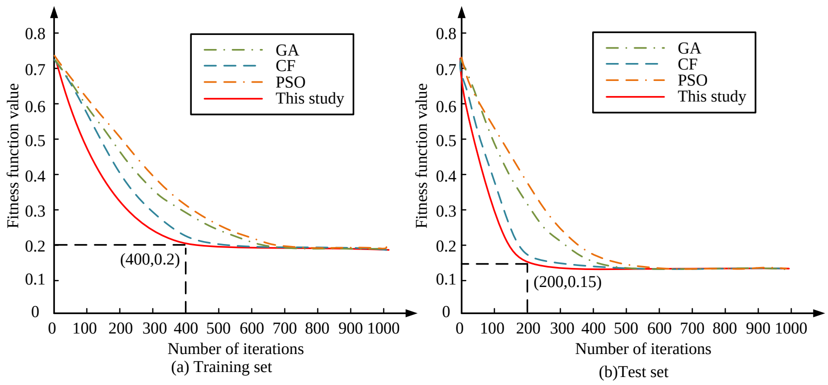
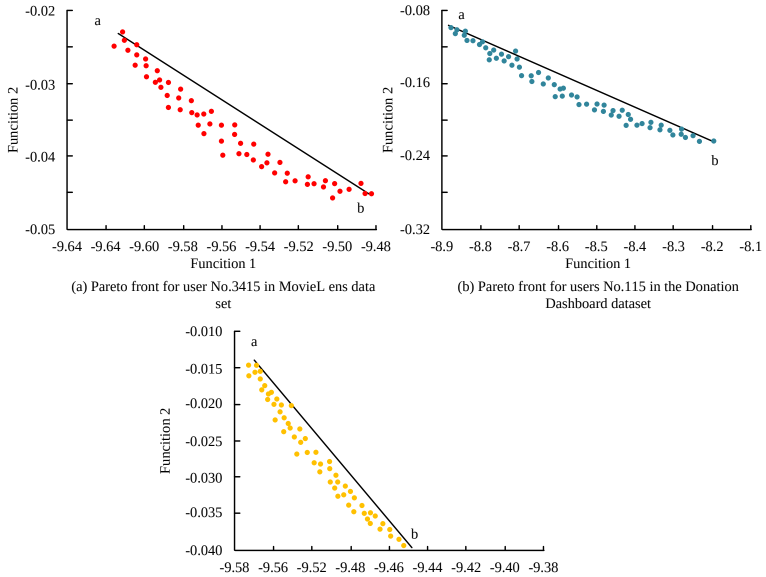


Figure 10: Iteration performance of the algorithm on testing and training datasets

Figure 10 (a) displays the iteration efficiency of four different optimizers on the training dataset. From the training data, as the iteration increased, the fitness function values of all four methods gradually decreased and eventually stabilized. Figure 10 (b) displays the iterative performance test results on the testing set. The proposed new algorithm only required at least 200 iterations to achieve a fitness function value of 0.15. The proposed method is used to generate Pareto frontiers, as displayed in Figure 11.



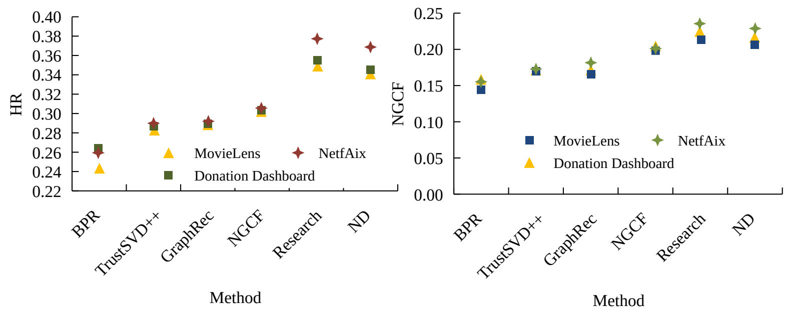
(a) Pareto front for user No.3415 in MovieL ens data set

(b) Pareto front for users No.115 in the Donation Dashboard dataset

(c) Pareto front for No.2155 users in the Netflix dataset

Figure 11: Results of pareto frontier generated by the algorithm for different users

As shown in Figure 11, Function1 is a matching function used to determine similarity, while Function2 measures the diversity of the recommendation list. Each punctuation mark in the diagram represents a recommendation sequence. Figure 11(a) had the highest similarity among all recommendations, but its diversity was the lowest. Figure 11(b) showed the highest diversity among all generated recommendations, but performed poorly in the similarity. Considering that user preferences are fundamentally influenced by individual subjective feelings, there is currently no unified quantitative criterion to determine the best recommendation. Figure 12 shows the results of the hit rate @10 and normalized discounted cumulative gain @10 for the research model and five other advanced recommendation models on the MovieLens, Donation Dashboard, and NetfAix datasets.



(a) Hit rate @10

(b) Normalized discounted cumulative gain @10

Figure 12: Experimental results under different data sets and different dimensions

As shown in Figure 12, the proposed method showed better performance on all three datasets. On the dataset NetfAix, the research method had the highest hit rate @10, which was 0.3781. The maximum normalized discounted cumulative gain @10 was 0.2349. Similarly, regardless of the dataset, the evaluation index of the research method is consistently higher than the other five recommendation models. Figure 13 displays the comparison of recommendation accuracy between the research model and the recommendation algorithms in literature [4], [9], and [12].

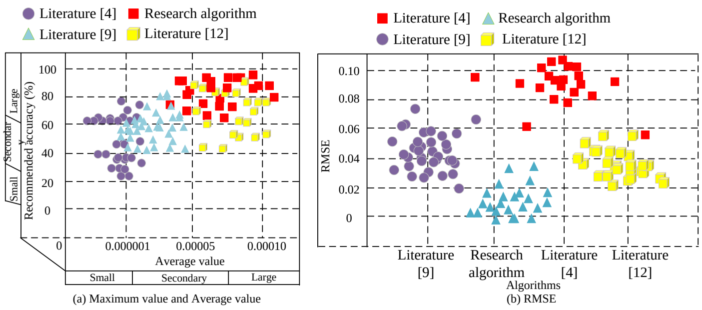


Figure 13: Comparison results of recommendation performance of four recommendation models

As shown in Figure 13 (a), the recommendation accuracy range of the designed recommendation algorithm was over 60%, with an average recommendation accuracy of approximately 95.2%. The average accuracy of the three recommendation algorithms in literature [4], [9], and [12] were 56.4%, 72.8%, and 80.1%, respectively. As shown in Figure 13 (b), the RMSE of the designed recommendation algorithm was below 0.04, and its mean RMSE was approximately 0.015. The mean RMSE of the three recommendation algorithms in literature [4], [9], and [12] were 0.094, 0.037, and 0.026, respectively. To further verify the performance of the research algorithm in personalized recommendation, the algorithms in literature [6], [7] and [10] are selected for comparison. Table 3 shows the comparison results of the personalized recommendation performance indicators of each method.

Table 3: Comparison results of data processing performance indicators of each method

Index	Literature [6]	Literature [7]	Literature [10]	Designed algorithm
Error rate (%)	1.37	2.19	3.54	0.21
Coverage (%)	82.27	86.27	89.10	93.74
Accuracy (%)	88.71	90.52	91.24	98.48
Novelty (%)	82.36	85.74	88.67	90.25
Diversity (%)	85.47	87.38	89.92	94.67

From Table 3, the designed algorithm performed well on several metrics. The error rate was only 0.21%, which was significantly lower than 1.37%, 2.19% and 3.54% in literature [6], [7] and [10]. The coverage rate reached 93.74%, which was significantly higher than other methods, indicating a wider range of recommendations. In terms of accuracy, the designed algorithm reached 98.48%, which was far higher than other literature methods, showing high recommendation accuracy. Although the novelty was slightly lower than 88.67% of the literature [10], it was still better than other literature. The diversity performance of designed algorithm was the best, reaching 94.67%, which was significantly ahead of other methods, showing excellent performance in diversified recommended content.

4 Discussion

Although the SVR algorithm proposed by Ganesh and Velu et al. [5] improved the recommendation accuracy, its complexity limited its scalability. The research reduced the complexity and enhanced the scalability based on ResNet connections. Compared with the method proposed by Bhaskaran et al. [6], this study achieved a better balance between recommendation accuracy and diversity, avoided the dependence on a single user feature, and improved adaptability. Duan et al. [7] relied on trust network, while the study reduced rating bias through PCC and combined with CNN to capture long-term user behavior, which maintained high accuracy even under the support of trustless network. Although not directly applied to flight forecasting, the study demonstrated a wider range of application scenarios and data processing capabilities through CNN and data mining techniques. Compared with the method proposed by Wang et al. [9], this study not only improved the accuracy, but also enhanced the diversity of recommendations through multi-objective immune optimization, which has a wider application range. The method designed by Hasan and Ferdous [10] was not as stable as the research on large-scale data processing, while Chang et al. [11] relied on external data. The research reduced privacy risks through internal data optimization. Ma et al. [12] did not perform well when data was scarce. However, their research method could still maintain efficient recommendation in the case of new users or sparse data through PCC and multi-objective immune optimization.

5 Conclusion

To solve the low accuracy and slow recommendation speed in current intelligent recommendation algorithms, a user big data cycle intelligent recommendation based on mining algorithms and a personalized recommendation optimization algorithm combined with multi-objective immune optimization were proposed. Compared with LDA and ARMD, the proposed MFCR-BCIR algorithm exhibited higher stability and accuracy. The similarity accuracy of its calculation was close to 91%, at least 6.9% higher than other algorithms in accuracy. This algorithm only required a small number of epochs to achieve relatively stable results. Its outstanding performance is due to its ability to capture the time patterns of user software usage through neural network mining algorithms, revealing the deep connections between user browsing habits at different time periods. For the MFCR-BCIR algorithm, the precision was improved by at least 6%. The recall rate of LDA and ARMD algorithms did not exceed the threshold of 80%, while the recall rate of MFCR-BCIR algorithm ranged from 85% to 95%, which can be achieved based on limited data volume. The proposed algorithm has a significant advantage in recall rate, which can make more accurate recommendations with less data, and its recall rate has increased by at least 7.9%. The MFCR-BCIR algorithm only required 200 iterations in testing analysis to obtain a fitness function value of 0.15. The research algorithm outperformed other models on the MovieLens, Donation Dashboard, and NetfAix datasets, especially on the NetfAix dataset, with hit rates and normalized discounted cumulative gains reaching 0.3781 and 0.2349, respectively. This algorithm outperformed the other five recommendation systems on various datasets, with an accuracy rate of over 60% and an average of 95.2%. The RMSE of the recommendation algorithm was less than 0.04, with an average value of about 0.015. The mean values of the other three algorithms were 0.094, 0.037, and 0.026, respectively. Although the research method has greatly enhanced the personalized recommendation accuracy, it can be optimized. The existing research on recommendation algorithms is mainly based on a single user behavior. The key in the future will be to integrate the diverse behaviors exhibited by users in life and social media, such as labeling and commenting, to more accurately predict preferences. At the same time, the time dimension will be incorporated to adjust recommendations in real-time to improve user satisfaction.

Funding

The research is supported by: Basic Scientific Research Projects of Liaoning Provincial Department of Education, Research on personalized recommendation algorithm based on python data mining (No. LJKMZ20222047).

Reference

[1] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," Big Data Mining and Analytics, vol. 4, no. 3, pp. 139-154, 2021. https://doi.org/10.26599/BDMA.2020.9020026

[2] J. Wang, K. Yue, and L. Duan, "Models and techniques for domain relation extraction: A survey," Journal of Data Science and Intelligent Systems, vol. 3, no. 1, pp. 16-25, 2023. https://doi.org/10.47852/bonviewJDSIS3202973

[3] N. Mou, Q. Jiang, L. Zhang, J. Niu, Y. Zhang, Y. Wang, and T. Yang, "Personalized tourist route recommendation model with a trajectory understanding via neural networks," International Journal of Digital Earth, vol. 15, no. 1, pp. 1738-1759, 2022. https://doi.org/10.1080/17538947.2022.2130456

[4] J. Sun, M. Zhu, Y. Jiang, Y. Liu, and L. Wu, "Hierarchical attention model for personalized tag recommendation," Journal of the Association for Information Science and Technology, vol. 72, no. 3, pp. 173-189, 2021. https://doi.org/10.1002/asi.24400

[5] R. Ganesh, and C. M. Velu, "A peculiar approach of movie rating prediction using Svr-Algorithm comparing to matrix decomposition with accurate rating," ECS Transactions, vol. 107, no. 1, pp. 14233-14246, 2022. https://doi.org/10.1149/10701.14233ecst

[6] S. Bhaskaran, and R. Marappan, "Enhanced personalized recommendation system for machine learning public datasets: generalized modeling, simulation, significant results and analysis," International Journal of Information Technology, vol. 15, no. 3, pp. 1583-1595, 2023. https://doi.org/10.1007/s41870-023-01165-2

[7] Z. Duan, W. Xu, Y. Chen, and L. Ding, "ETBRec: a novel recommendation algorithm combining the double influence of trust relationship and expert users," Applied Intelligence, vol. 52, no. 1, pp. 282-294, 2022. https://doi.org/10.1007/s10489-021-02419-x

[8] W. Shao, A. Prabowo, S. Zhao, P. Koniusz, and F. D. Salim, "Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map," Neurocomputing, vol. 472, no. 2, pp. 4280-293, 2022. https://doi.org/10.1016/j.neucom.2021.04.136

[9] X. Wang, W. Ma, L. Guo, H. Jiang, F. Liu, and C. Xu, "HGNN: Hyperedge-based graph neural network for MOOC course recommendation," Information Processing & Management: Libraries and Information Retrieval Systems and Communication Networks: An International Journal, vol. 59, no. 3, pp. 1-18, 2022. https://doi.org/10.1016/j.ipm.2022.102938

[10] M. D. R. Hasan, and J. Ferdous, "Dominance of AI and machine learning techniques in hybrid movie recommendation system applying text-to-number

conversion and cosine similarity approaches," Journal of Computer Science and Technology Studies, vol. 6, no. 1, pp. 94-102, 2024. https://doi.org/10.32996/jcsts.2024.6.1.10

[11] J. L. Chang, H. Li, and J. W. Bi, "Personalized travel recommendation: a hybrid method with collaborative filtering and social network analysis," Current Issues in Tourism, vol. 25, no. 14, pp. 2338-2356, 2022. https://doi.org/10.1080/13683500.2021.2014792

[12] T. Ma, X. Wang, F. Zhou, and S. Wang, "Research on diversity and accuracy of the recommendation system based on multi-objective optimization," Neural Computing and Applications, vol. 35, no. 7, pp. 5155-5163, 2023. https://doi.org/10.1007/s00521-020-05438-w

[13] E. Koc, and A. Y. Ayyildiz, "An overview of tourism and hospitality scales: Discussion and recommendations," Journal of Hospitality and Tourism Insights, vol. 5, no. 5, pp. 927-949, 2022. https://doi.org/10.1108/JHTI-06-2021-0147

[14] C. Sardianos, I. Varlamis, C. Chronis, G. Dimitrakopoulos, A. Alsalemi, Y. Himeur, F. Bensaali, and A. Amira, "The emergence of explainability of intelligent systems: Delivering explainable and personalized recommendations for energy efficiency," International Journal of Intelligent Systems, vol. 36, no. 2, pp. 656-680, 2021. https://doi.org/10.1002/INT.22314

[15] M. A. Raji, H. B. Olodo, T. T. Oke, W. A. Addy, O. C. Ofodile, and A. T. Oyewole, "E-commerce and consumer behavior: A review of AI-powered personalization and market trends," GSC Advanced Research and Reviews, vol. 18, no. 3, pp. 066-077, 2024. https://doi.org/10.30574/gscarr.2024.18.3.0090

[16] S. Bhaskaran, and R. Marappan, "Design and analysis of an efficient machine learning based hybrid recommendation system with enhanced density-based spatial clustering for digital e-learning applications," Complex & Intelligent Systems, vol. 9, no. 4, pp. 3517-3533, 2023. https://doi.org/10.1007/s40747-021-00509-4

[17] O. A. Odejide, and T. E. Edunjobi, "AI in project management: exploring theoretical models for decision-making and risk management," Engineering Science & Technology Journal, vol. 5, no. 3, pp. 1072-1085, 2024. https://doi.org/10.51594/estj.v5i3.959

[18] Z. Yao, Y. Wang, F. Yang, Y. Chen, and B. Ran, "Map matching for travel route identification based on earth mover's distance algorithm using wireless cell trajectory data," Journal of Intelligent Transportation Systems, vol. 25, no. 1, pp. 644-656, 2021. https://doi.org/10.1080/15472450.2021.1955209

[19] F. Huang, J. Xu, and J. Weng, "Multi-task travel route planning with a flexible deep learning framework," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 3907-3918, 2020. https://doi.org/10.1109/TITS.2020.2987645

[20] S. Arslan, "Exploring the potential of chat GPT in personalized obesity treatment," Annals of Biomedical Engineering, vol. 51, no. 9, pp. 1887-1888, 2023. https://doi.org/10.1007/s10439-023-03227-9