Journal of Network Intelligence ©2023 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **8**, Number **4**, November **2023**

Research on Intelligent Recommendation of Employment Big Data with Affinity Propagation Clustering

Aiping Chen*

School of Marxism
Jiangxi Vocational College of Applied Technology
Ganzhou 341000, China
jxyy2602@126.com

Ting Liu

School of Marxism
Jiangxi Vocational College of Applied Technology
Ganzhou 341000, China
dailiping_1118@163.com

Liping Dai

School of Marxism
Jiangxi Vocational College of Applied Technology
Ganzhou 341000, China
tinaliut@163.com

*Corresponding author: Aiping Chen
Received February 23, 2023, revised April 26, 2023, accepted June 6, 2023.

Abstract. *The massive amount of data generated by the Internet has raised the problem of information overload. Through search engines, we can retrieve relevant content, but cannot satisfy our personalised data needs, and most users still find it difficult to find a suitable job quickly. Personalised recommendation systems have become a powerful tool for solving the information overload problem. To solve the problem of employment information overload in the big data environment, this work presents an intelligent recommendation model for employment big data with Affinity Propagation (AP) clustering algorithm. Firstly, under the control of association rule constraints, the amount of interest relevance features of user employment is collected. For adaptive matching of feature points of interest in employment recommendation, the Apriori algorithm is employed. Secondly, the users are clustered by applying the improved AP clustering algorithm, combined with the SimRank algorithm to find the similarity between users and companies, from which the final ranking results of recommended companies are achieved. The proposed improved AP clustering algorithm optimises the Preference parameters through Cuckoo Search (CS) and obtains stable clustering results. The simulation results show that the proposed intelligent recommendation model has a hit rate of 0.66 and a recommendation ranking index of 5.8, which can provide satisfactory personalised recommendation services in a massive data environment.*
Keywords: Employment recommendation; Apriori; adaptive matching; Affinity propagation; Cuckoo search optimization

---

1. **Introduction.** While the Internet has brought us great convenience since the 21st century, the massive amount of data it has generated has also raised the problem of information overload. Through search engines, we can retrieve relevant content, but cannot satisfy our personalised data needs [1,2]. Personalised recommendation systems have widely entered our horizon. It has become a powerful tool to solve the information overload problem.

However, the existing personalised recommendation system is still unable to provide efficient and fast personalised services in the face of massive data, so it is necessary to put in place a personalised recommendation system that can efficiently handle large-scale data in order to provide satisfactory personalised recommendation services in a massive data environment [3,4]. Through investigation and analysis, a large number of employment recruitment information is released on third-party employment information service websites. Freshly graduated university students are not very clear about their personal career pursuits because of the relatively narrow job search channels. With the support of the Internet, a large number of various employment information like spam advertisements are popping up, resulting in a difficult employment environment where it becomes very difficult for most students to find a job that meets their personal reality [5]. According to the latest research report, the overall satisfaction of college graduates with employment is low, where the number of satisfied people only accounts for 33.7% of the surveyed people. A party of students need to sift through the huge amount of employment information to meet their own employment wishes, a process that can cost a huge amount of job search costs, including time costs, energy costs, and opportunity costs.

Due to the popularity and development of the Internet and information technology, the volume of information faced by users has increased dramatically, resulting in a decrease in the efficiency of users in accessing information and the phenomenon of information overload (information overload) [6,7,8]. Recommendation systems have been proposed to solve this problem. This work takes data mining and recommendation system as the theoretical basis to build an accurate and effective employment big data intelligent recommendation model, which can provide graduates with scientific, individual and reliable employment recommendations in the process of career selection and employment. The Apriori algorithm is employed to effectively match and discover the best after user interest large data mining of university students' career is done. Then, we use the improved Affinity Propagation (AP) clustering algorithm to cluster users and combine it with the SimRank algorithm to find the similarity between users and companies (the ranking result of recommended companies) to realise university students' employment recommendations. Finally, the validity is concluded through the analysis of simulation experiments. The proposed model helps to achieve an increase in the employment rate of students and improve the quality of employment.

1.1. **Related work.** Recommendation systems need to recommend objects that fit the user's interests according to their preferences. Based on the core idea of personalised recommendation systems, they are now used not only in e-commerce, but also in a wide range of industries, including music, movies, news, emails, etc. [9].

Recommendation technology is a problem that the big data analytics industry must face. When users do not specifically input their needs, recommendation technology needs to analyse all aspects of user behaviour and other relevant information in order to intelligently filter what does not interest the user and thus recommend goods of potential interest to the user themselves. Currently, there are two main categories of recommendation systems [10], namely collaborative filtering-based recommendation systems and content-based recommendation systems. The main areas of application for recommendation systems are shown in Table 1.

Table 1. Key areas of application for recommendation systems.

Fields	**Referral System**
Mail	Tapestry
Web	FoxtrotM, EMOIRMETIOREW, assnt, Commtysearch, Fab
Film	Nakif, MovieLens, Moviefinder.com, Recommend Explorer, CBCF
E-commerce	FAIRWIS, Amazon.com, EFOL, entre, Dietorecs, Ghani, eBay
News	GroupLens, PHOAKS, P-Tango
Music	CDNOW, Ringo, CoCoA

Srifi et al. [11] apply collaborative filtering algorithms to recommender systems. First, the similarity between users is calculated using their historical information. Then, the ratings of neighbours who are more similar to the target user are used to predict the target user's preference for a particular product. The system makes recommendations to the target user based on this level of preference. The biggest drawback of collaborative filtering algorithms is that they rely too much on user reviews. The recommendation process based on collaborative filtering is shown in Figure 1

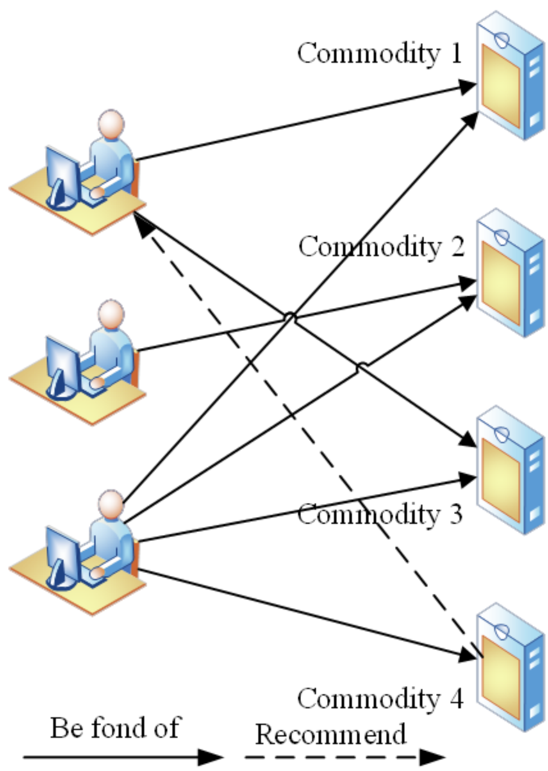

Figure 1. Recommendation process based on collaborative filtering

Content-based recommendation is a continuation of collaborative filtering, and Wang et al. [12] propose a content-based personalised recommendation model. This model does not rely on the user's opinion of the item, but rather calculates the similarity between users based on the product content information they have already selected, and makes recommendations accordingly. Kant et al. [13] extracted feature factors from industry big data and then used K-means clustering for personalised recommendations. Curiskis et al. [14] focused on the analysis of commercial competition and used clustered text mining methods to effectively extract user preferences for different products, whose main data source is the online review data of the products.

1.2. **Motivation and contribution.** Affinity Propagation clustering algorithm [15,16,17], as a relatively novel clustering method, does not require a pre-given number of clusters and has better clustering performance and efficiency compared to traditional clustering methods. Therefore, this work uses the AP clustering algorithm to apply to intelligent recommendation of employment big data.

The main innovations and contributions of this study are shown below.

(1) In relevant sections such as user attribute selection, the association rule algorithm in data mining techniques was used for filtering. The analysis of the results of the Apriori algorithm resulted in attributes that were more relevant to the employment problem and improved the accuracy of the recommendations.

(2) The traditional AP clustering algorithm uses Euclidean distance to measure the similarity between objects, but the similarity matrix represented by Euclidean distance is difficult to achieve global consistency for sparse data, thus seriously affecting the clustering performance. Therefore, this work proposes to combine the similarity obtained by the SimRank algorithm with AP clustering to address the effectiveness of the clustering distance.

(3) To address the problem that the performance of the traditional AP clustering algorithm is highly dependent on the Preference parameters, this work also proposes to optimize the Preference parameters of AP clustering using the Cuckoo Search algorithm to improve the applicability of the AP algorithm in employment recommendation analysis.

2. Characteristic variable sampling and adaptive matching.

2.1. **Interest relevance feature quantity collection and solution.** Prior to using the association rules mining approach to meet the interest feature points in the job recommender system, user interest data is first collected in order to achieve the employment suggestion for college students.

Information sampling for employment personalised recommendations is done using a label recognition method, while the output is a sequence of sampled data with personalised interest features $x = \{x_1, x_2, \cdots, x_m\}$. $I = \{i_1, i_2, \cdots, i_n\}$ denotes the set of items of user's employment intention, where m denotes the number of users and n denotes the number of employment positions. After sampling the user interest information, the interesting date fusion feature quantity *p(x)* for employment personalised recommendation is obtained.

$$p(x) = \frac{x_m}{\sum_{i=1}^{n} I_i \cdot u_m} \tag{1}$$

where u_m is the user's interesting feature index. The interesting distribution feature sampling method is then built.

$$P(k) = \frac{p(x)}{\sum_{i=1}^{n} I_i(l(k) \cdot q(k))} \tag{2}$$

Based on information fusion features and user interest distribution features, the employment personalised recommendation information is jointly identified and matched to jobs under a fuzzy association rule scheduling model. Use I_a and I_b to show the evaluation sets of users u_a and u_b , respectively. The similarity between users u_a and u_b is

$$\operatorname{sim}(u_a, u_b) = \frac{\sum_{i \in I_a \wedge i \in I_b} p(x)P(k)}{\sqrt{\sum_{i \in I_a} \left(r_{u_a,i} - \bar{r}_{u_a}\right)^2 \cdot \sum_{i \in I_b} \left(r_{u_b,i} - \bar{r}_{u_b}\right)^2}} r \tag{3}$$

where r denotes the associative directionality coefficient of employment personalised recommendations and *u(t)* is the effective component of the interest feature.

Distributed reorganisation of interest features based on the sum of the sum of the individual users. The similarity attributes between two two users in the group are calculated to obtain the standard quantitative set of college student employment recommendations. In order to actualize the sampling of user interest aspects for university student employment, a big data evolutionary game method for customized recommendation of university student employment is created utilizing the approach of fuzzy rule mining.

Analyse the absolute value of the difference between item ratings and construct a relevance feature extraction model for personalised recommendations for employment.

$$y = F(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \tag{4}$$

where y denotes the set of nodes for job assignment between two two users, and $f_m(x)$ denotes the recommended job for the mth user.

Under the distributed recommendation model, the big data of interest features of college students' employment recommendations are optimally fused and processed to obtain the weights for adaptive learning.

$$W_k(\mathbf{U}) = \alpha \left(\frac{1}{m} \sum_{i=1}^{m} \frac{\sum_{j \in \mathrm{Item}_i} r_{i,j}}{\sum_{j \in \mathrm{Item}_i} \hat{r}_{i,j} + r_k} \right) \tag{5}$$

$$W_k(\mathbf{V}) = \alpha \left(\frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{j \in \text{ User }_i} r_{i,j}}{\sum_{j \in \text{ User }_i} \hat{r}_{i,j} + r_k} \right) \tag{6}$$

where $\mathbf{U}$ and $\mathbf{V}$ are both fuzzy clustering feature vectors. $Item_i$ is the number of employment intention weights; $User_i$ is the number of user association weights.

User interest preference analysis in employment based on user interest characteristics. Apriori learning method is used for group size classification. The adaptive learning function is *W(k)*.

$$W(k) = W_k(\mathbf{U})[1 - W_k(\mathbf{V})]^{k-1} \tag{7}$$

The quantified search fuzziness function $E(k)$ between employment personalisation needs and interest features is obtained in the fuzzy domain of Apriori learning [18].

$$E(k) = \sum_{k=0}^{\infty} [1 - W(k)]^k \tag{8}$$

The average number of time slots for Apriori learning [19] is

$$T_{l\text{ - ary}} = E(k)n_i = \frac{L}{(1 - 1/n)^{m-1}} \tag{9}$$

2.2. **Adaptive matching of interest feature points based on Apriori algorithm.** Based on the above analysis, the interest relevance feature quantity is constructed and the large data of interest features for employment recommendations are optimally fused.

The level of adaptability of employment recommendations is improved by calculating the joint information entropy of user interests. The Apriori method is utilized for flexible matching of feature points to obtain the matching relationship between employment recommendations and user interests.

$$E^{cv}(c_1, c_2) = \mu\cdot \text{ Length } (C)+\delta\cdot \text{ Area } (\text{inside } (C))+ \lambda_1 \int_{\text{inside } (C)} |I - c_1|^2+\lambda_2 \int_{\text{outside } (C)} |I - c_2|^2 \tag{10}$$

where c_1 and c_2 denote interest preferences with more similar data attributes, respectively, Length(C) denotes the length of the distribution of employment job attributes, Area(inside(C)) denotes the set of area distributions of nearest neighbours, and μ ,σ, λ_1 and λ_2 all denote the semantic autocorrelation coefficients (all constants greater than 0) of personalised recommendations for employment.

Based on the above analysis, the Apriori learning model for user employment is

$$C = \text{Min}\left\{\max\left(C_i\right)\right\} \tag{11}$$

$$\sum_{j=1}^{n} Z_j = 1, \forall i \in (1, n), \forall j \in (1, n_i) \tag{12}$$

where: C_i denotes the fuzzy correlation coefficient of university students' employment, Z_j denotes the overall satisfaction level.

Let the set of association rule distributions for employment personalized recommendations be $S = \overline{X_1}, \overline{X_2}, \cdots, \overline{X_k},$, and the set of employment satisfaction and job matching features be $T_1, T_2, \cdots, T_K$, then the optimal solution feature vector for Apriori learning is

$$\widetilde{\mathbf{w}}_k^i = \widetilde{\mathbf{w}}_{k-1}^i \frac{l\left(z_j/\tilde{x}_k^i\right) l\left(\tilde{x}_k^i/x_{k-1}^i\right)}{q\left(\tilde{x}_k^i/x_{k-1}^i\right)} \tag{13}$$

where $\tilde{x}_k^i$ is the user's interest preference for employment and $\widetilde{\mathbf{w}}_{k-1}^i$ is the weight.

The Apriori algorithm was used to perform adaptive matching of interest feature points for user employment recommendations, and the optimized matching iterative process was obtained as shown below.

$$x_i(k+1) = x_i(k) + \alpha\left(\frac{x_j(k) - x_i(k)}{\|x_j(k) - x_i(k)\|}\right) \tag{14}$$

where α is the standard parameter recommended for employment.

The results of the final employment satisfaction scores are shown below.

$$F\left(Z_j, i\right) = \widetilde{\mathbf{w}}_k^i \cdot \alpha\left(Z_j, i\right) + \widetilde{\mathbf{w}}_{k-1}^i \cdot \left(1 - \alpha\left(Z_j, i\right)\right) \tag{15}$$

3. An employment recommendation model based on an improved AP clustering algorithm.

3.1. **SimRank algorithm to calculate inter-user similarity.** Traditional AP algorithms usually use Euclidean distance to measure the similarity between objects, but Euclidean distance indicates a high degree of similarity between spatially adjacent data points.

As a result, it is difficult to achieve global consistency with sparse data using the similarity matrix of the Euclidean distance metric AP algorithm, which in turn severely affects clustering performance. To further illustrate the problem, this work was analysed using Compound data, as shown in Figure 2. It can be intuitively seen that the similarity

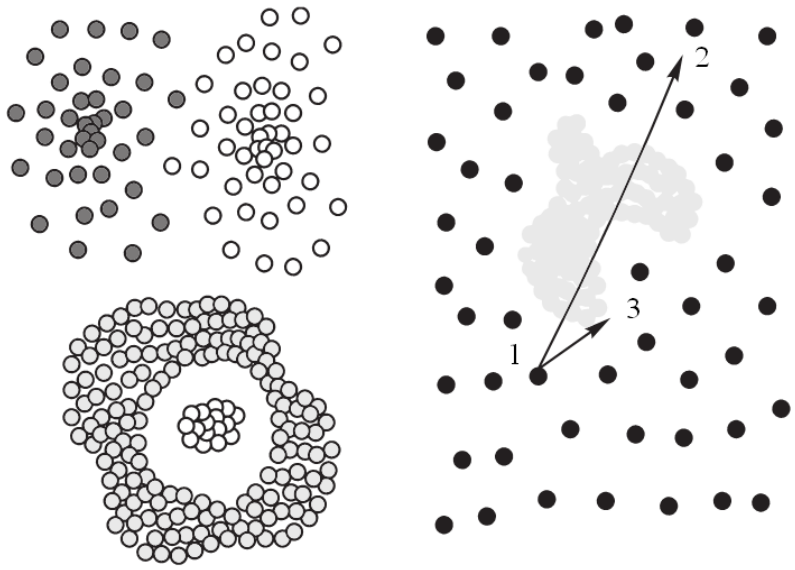

Figure 2. Euclidean distance

between data point 1 and data point 2 is greater than the similarity between data point 1 and data point 3. However, when using Euclidean distance as a similarity measure, the straight line distance between data point 1 and data point 3 is clearly smaller than the distance between data point 1 and data point 2. The probability of data point 1 and data point 3 being classified as the same class will be greater than that of data point 2. Therefore, the global consistency cannot be reflected when using Euclidean distance as the data point similarity measure. For sparse data sets, if the AP algorithm simply uses Euclidean distance to calculate the similarity between data points, the performance of the clustering algorithm will be seriously affected.

Therefore, this work uses the SimRank algorithm [20] to compute the similarity of any two of all users. Let the set of students be D. Users D_1 and D_2 belong to D. $S(D_1, D_2)$ denotes the inter-user similarity. Let the set of user feature attributes be E. $S(E_1, E_2)$ denotes the inter-user feature attribute similarity. *L(X)* denotes the set of nodes directly connected to point $X.|L(X)|$ denotes the number of nodes directly connected to point $X.L_i(X)$ denotes the i-th node directly connected to point X.

$$S(D_1, D_2) = \left\{ \begin{array}{cc} 1 & , D_1 = D_2 \\ \frac{C}{|L(\mathrm{D}_1)||L(\mathrm{D}_2)|} \sum\limits_{i=1}^{|L(\mathrm{D}_1)||L(\mathrm{D}_2)|} \sum\limits_{j=1} S(L_i(D_1), L_j(D_2)) & ,\text{others} \end{array} \right\} \tag{16}$$

$$S(E_1, E_2) = \left\{ \begin{array}{cc} 1 & , E_1 = E_2 \\ \frac{C}{|L(\mathrm{E}_1)||L(\mathrm{E}_2)|} \sum\limits_{i=1}^{|L(\mathrm{E}_1)||L(\mathrm{E}_2)|} \sum\limits_{j=1} S(L_i(E_1), L_j(E_2)) & ,\text{others} \end{array} \right\} \tag{17}$$

where C is an adjustment parameter that can be used to normalise the effect of the size of the result interval on the distribution of results.

3.2. **AP clustering principle.** The basic idea of the AP clustering algorithm is to consider all data points as potential clustering centres [21,22,23]. the AP clustering algorithm connects two pairs of data points to each other to form a network (similarity matrix) and passes messages (attractiveness and affiliation) at each end of the network to calculate the clustering centres for each sample.

Let $S(i,j)$ be the similarity between the samples i and j.

$$S(i,j) = -\|x_i - x_j\|^2 \tag{18}$$

After obtaining the similarity matrix for all data points, the diagonal elements can be referred to as the Preference parameter P. In practice, the value of P has a large influence on the results of the clustering and must therefore be set appropriately when operating.

Define $r(i,j)$ to represent the attractiveness function, $a(i,j)$ to represent the affiliation function, $R = [r(i,j)]_{N\times N}$ to represent the composition matrix, and $A = [a(i,j)]_{N\times N}$ to represent the affiliation matrix. If $r(i,j) + a(i,j)$ is larger, it means that the points i and j are more similar.

The update process for $a(i,j)$ is shown below.

$$r(i,j) = s(i,j) - \max_{j' \text{ s.t. } j' \neq j} \{a(i,j') + s(i,j')\} \tag{19}$$

$$a(i,j) = \min\{0, r(j,j) + \sum_{i' \text{ s.t. } i' \notin \{i,j\}} \max\{0, r(i',j)\}\} \tag{20}$$

where $r(j,j)$ is the self-attraction of the node j .

When $i = j$, the calculation of $a(i,j)$ changes[24,25].

$$a(j,j) = \sum_{i' \text{ s.t. } i' \neq j} \max\{0, r(i',j)\} \tag{21}$$

$$r(i,j) + a(i,j) = s(i,j) + a(i,j) - \max_{j' \text{ s.t. } j' \neq j} \{a(i,j') + s(i,j')\} \tag{22}$$

Define $E = \Gamma[\tau(i,j)]_{N\times N}[s(i,j) + a(i,j)]_{N\times N}$, then Γ is the potential array.

$$e(i,j) = \tau(i,j) - \max_{j' \text{ s.t. } j' \neq j} \{\tau(i,j')\} \tag{23}$$

The damping factor effectively balances the elimination of oscillations with the speed of convergence.

$$R_T = (1-\phi)R_T + \phi R_{T-1} \tag{24}$$

$$A_T = (1-\phi)A_T + \phi A_{T-1} \tag{25}$$

The AP clustering Silhouette assessment metric for sample t is *Silt(t)*.

$$Sil(t) = \frac{b(t) - a(t)}{\max\{a(t), b(t)\}} \tag{26}$$

where $a(t)$ is the mean value of the distance between t and other points in the same category, *Silt(t)* takes values in the range [-1, 1].

3.3. **SCS-AP clustering algorithm.** After solving the global consistency problem of sparse data using the SimRank algorithm, in order to reduce the influence of the Preference parameter on the effect of AP clustering, as well as to reduce the sample points of misclassification within clusters, this work further optimizes AP clustering using the CS algorithm, referred to as the SCS-AP clustering algorithm.

In order to solve the problem that AP is not applicable to sparse data, reduce the influence of the Preference parameter value on the algorithm and reduce the sample points of misclassification within clusters, the SCS-AP algorithm is proposed in this paper. The algorithm uses the SimRank algorithm to calculate the inter-user similarity on the basis of the traditional AP algorithm, which effectively solves the problem that AP is not applicable to sparse data. At the same time, the CS algorithm is used to adjust the Preference parameter adaptively and reduce the intra-cluster misclassification points to improve the clustering effect of the AP algorithm

Firstly, SCS-AP clustering uses the CS algorithm to calculate the sample points for the incorrect and correct clusters within each cluster from the previous iteration and obtains these sample point weights, which is shown as following:

Let the flock contain cuckoos [26] and the initial location is $X_0 = (x_1^0, x_2^0, x_3^0, \cdots x_N^0)$. The probability of a nest being found by a host is P_{a}, and the optimal nest and optimal fitness are x_{best}^0 and f_{best}^0 , respectively. The sample features are then updated using the above weights so that the appropriate Preference parameter values are reacquired for the next iteration.

Cuckoo flights obey certain distribution conditions.

$$L(s, \lambda) = s^{-\lambda}, \lambda \in (1, 3] \tag{27}$$

where s indicates the flight step.

The cuckoo location update method is shown below.

$$X^{t+1} = X^t + \alpha \cdot Levy(\lambda) \tag{28}$$

where $t = 1, 2, 3, \cdots n$, α are the movement step sizes. $Levy(\lambda)$ is the Lévy distribution.

$$Levy(\lambda) = \frac{\phi \times u}{|v|^{1/\lambda}} \tag{29}$$

$$\phi = \left\{ \frac{\zeta(1+\lambda) \times \sin(\pi \times \frac{\lambda}{2})}{\zeta\left[(\frac{1+\lambda}{2}) \times \lambda \times 2^{\frac{\lambda-1}{2}}\right]} \right\}^{1/\lambda} \tag{30}$$

where ζ is the Gamma function.

Let the adaptation optimal solution be $X_t = (x_1^t, x_2^t, x_3^t, \cdots x_N^t)$ after the t-th flight , where *1¡t¡T*. Let $r \in [0, 1]$ and no position update is performed when the condition is satisfied. Continue the flight to perform the nest position update under the condition that $r > P_a$ [27,28].

$$X_i^{t+1} = X_i^t + rand \times (X_j^t - X_k^t) \tag{31}$$

where X_j^t and X_k^t are the j nest position and k nest position after the tth flight, respectively, and $rand$ is a random number in the range (0,1). When the operation is stable, x_{best} and f_{best} are output.

The fitness function is a generic term for a population intelligence optimisation algorithm for finding the best, and needs to be designed specifically when oriented towards different problems. The proposed SCS-AP clustering uses the contour metric as the fitness function. Taking P as the nest location, *Silt(t)* is set as the fitness function. First, the sample matrix $S(i, j)$ is calculated after obtaining a sample of user interest data. Let the number of cuckoo nests be N , the maximum number of iterations T_{max} , initialize $r(i, j) = 0$ and $a(i, j) = 0$, and then perform SCS-AP clustering.

3.4. **Recommendation steps based on SCS-AP clustering.** By reasonably setting parameters such as discovery probability, movement step and damping factor of cuckoo hosts, SCS-AP clustering can obtain better clustering results. Compared with other commonly used clustering algorithms, the SCS-AP clustering algorithm can obtain higher profile index values and the shortest Euclidean distance performance.

The specific steps for the implementation of an intelligent recommendation model based on SCS-AP clustering are as follows:

Step 1: Determine the K values as well as the initial clustering centres, and select K initial clustering centroids as the centres of the K categories that you want to form. Since AP clustering considers all data points as potential cluster centres, K cluster centres are set according to the number of samples, the attributes of the users (gender, age, profession, etc.).

Step 2: The SimRank algorithm is called to calculate the similarity results between each user and the K central values. The K shape similarities of each user are compared to arrive at the corresponding category with the least similarity, thus forming the K initial categories.

Step 3: Recalculate the centroids of the K clusters, i.e. determine the clusters for this iteration based on the results of the previous iteration. First, the Preference parameter is initialised and set to the bird's nest in the CS algorithm. The contour indicator value is set as the fitness function of the CS algorithm. Next, the optimised Preference parameter is updated with the bird's nest position, thus continuously updating the decision and potential array for AP clustering.

Step 4: Repeat Step 2 and Step 3 until there is no significant change in the K centroids, thus obtaining a stable recommendation.

4. **Simulation results and analysis of employment referrals.**

4.1. **Experimental environment and dataset.**

To validate the performance of SCS-AP clustering in the application of intelligent recommendations for big data, example tests were conducted.

The experimental environment was a desktop computer with the open source linux series ubuntu 21.04 (Hirsute Hippo), an Intel I7 CPU, 8G of RAM, a GTX970 graphics card and MATLAB R2013b software. the test dataset was a database of 1000 university student profiles from the class of 2020 at a university. The number of companies recommended for employment was 100.

The initial user attributes contain 19 items. The tool used was the Apriori correlation analysis module of Microsoft's SQL Server 2005 data mining system. The data mining association analysis technique was used to find student attributes that were relatively more relevant to the employment enterprise problem and also to remove attributes that were not relevant to the problem. The purpose of filtering student data attributes is ultimately achieved. The training and testing ratio of the dataset was 8:2, and the filtered user attributes were nine items (Specialty, Language, ArtsScience, Score, Research, Professional-ability and Social-practice, etc.). The scale set of statistical feature measures for university employment was $Q = 200$, the fuzzy feature matching coefficients $c_1 = 0.34$ and $c_2 = 0.32$, the personalised preference feature assignment coefficient $c_r = 2$, and the recommendation model prediction coefficient $\mu_1 = \mu_2 = 0.01$. The number of Apriori iterations was 200 and the root mean square error $\delta = 0.8$.

4.2. **Pre-processing of the data.**

L1 parametric regularization was used to complete the feature normalization.

Firstly, the L1 parametric score value was calculated and the threshold value of the fitness was set to 0.6 in order to reduce the data dimensionality. The samples after feature selection were then normalised by a simple and quick mean-variance normalisation method, which was calculated as follows.

$$X_{scale} = \frac{X - min}{max - min} \tag{32}$$

where min represents the lower bound value, max represents the upper bound value, X represents the input feature value and X_{scale} represents the normalised feature value.

4.3. **Analysis of the results of SCS-AP clustering.** In the SCS-AP clustering calculation process, the profile evaluation index was selected as the clustering end condition.

The main parameters of the CS are $P_a = 0.25, \lambda = 1.5$ and $\alpha = 1$. The main parameter of the AP clustering algorithm is $\phi = 0.7$. When the clustering was completed, the sample of firms was divided into four classes from high to low performance as shown in Table 2. To further validate the optimisation performance of SCS-AP clustering, the clustering test was conducted on the student samples using AP clustering and SCS-AP clustering respectively, and the Euclidean distance of each student sample to the respective cluster centre was calculated, and the results are shown in Figure 3.

Table 2. Clustering of company recommendation indices.

Category	**Sample companies**
1	3, 11, 21, 22, 29, 40, 62, 68, 70
2	1, 5, 10, 16, 17, 18, 23, 27, 30, 31, 33, 34, 39, 45, 46, 47, 48, 53, 54, 55, 56, 61, 63, 65, 66, 67, 69, 73, 74, 75, 76
3	2, 6, 9, 12, 13, 15, 24, 25, 32, 35, 38, 41, 42, 43, 49, 50, 52, 57, 59, 60, 64, 72, 77, 78, 79, 80
4	4, 7, 8, 14, 19, 20, 26, 28, 36, 37, 44, 51, 58, 71

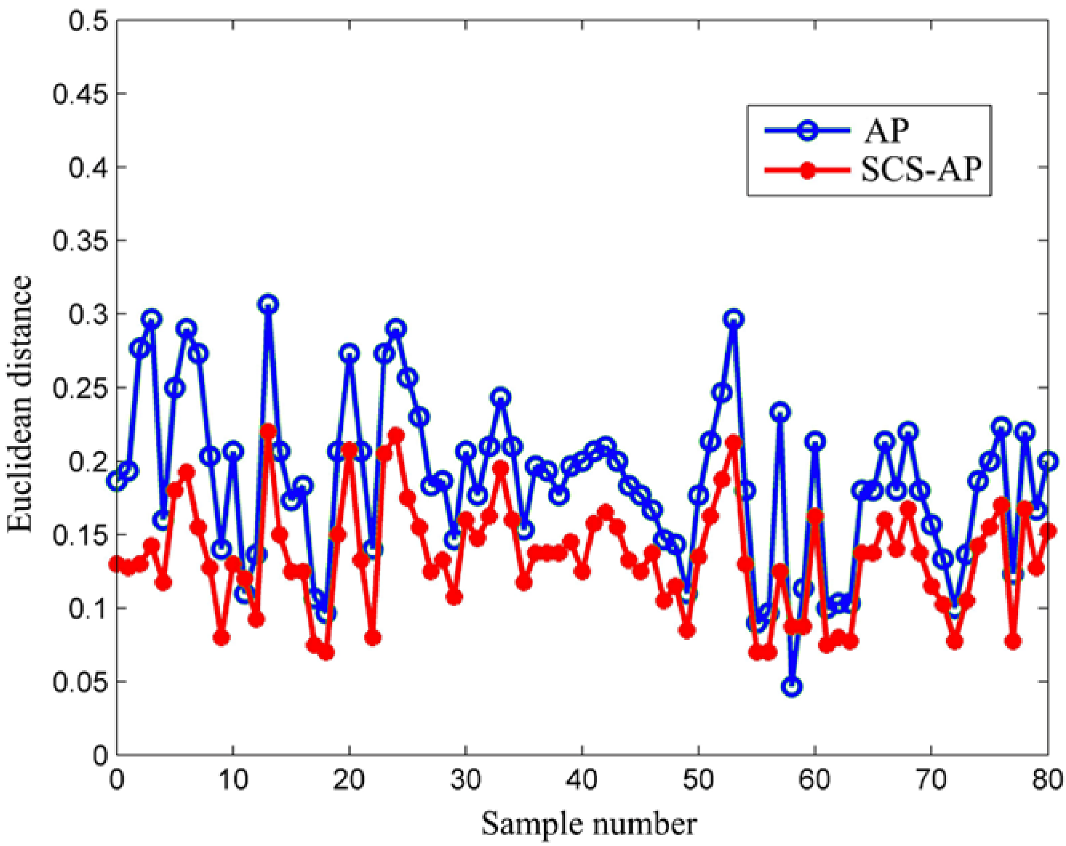

Figure 3. Euclidean distances for AP and SCS-AP algorithms

It can be obtained that the distance from each sample point to the class centroid of the AP algorithm basically fluctuates around 0.2, while the Euclidean distance of the SCS-AP algorithm basically fluctuates around 0.13, which indicates that SCS-AP is superior to AP. On the other hand, the distance fluctuation of the AP algorithm is greater, while the distance fluctuation of the SCS-AP algorithm is smaller, which indicates that the inter-class distance of the SCS-AP algorithm is smaller and the clustering This indicates that the SCS-AP algorithm has a smaller inter-class distance and a better choice of centroids. This is because the number of clustering classes and the number of clustering centroids are more reasonably chosen after the CS algorithm is used to optimise the Preference parameter P, resulting in better clustering results for the AP algorithm.

The contour performance *Silt(t)* was then tested on 80 samples using the AP and SCS-AP algorithms and the results are shown in Tabe 3. It can be seen that the mean

Table 3. Silhouette performance of AP and SCS-AP algorithms.

Algorithms	**Number of samples**	**Clustering categories**	**Silt(t)**	**Standard deviation**
AP	80	9	0.7316	0.5027
SCS-AP	80	4	0.8295	0.2439

of the 80 samples of SCS-AP is 0.8295, which is significantly better than the 0.7316 of the AP algorithm, and the standard deviation is better. After the CS optimisation, the clustering effect is significantly improved, and the samples are more closely distributed between classes closer to the centre of the clusters. The convergence performance of both is shown in Figure 4.

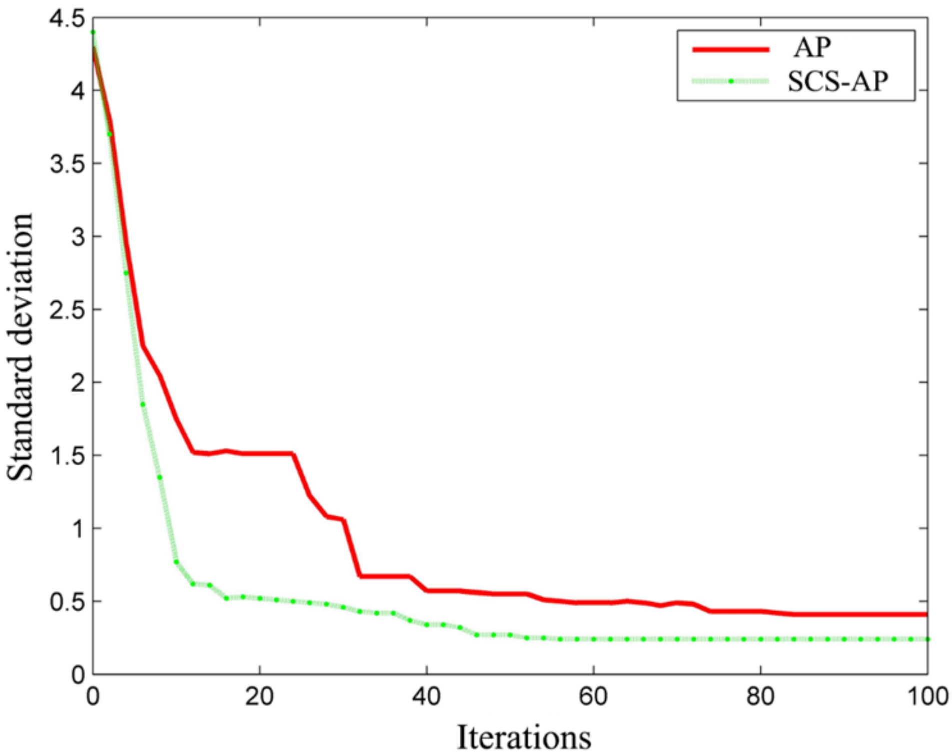

Figure 4. Convergence of AP and SCS-AP algorithms

It can be obtained that the standard deviation of both algorithms decreases rapidly until it is stable. However, the comparison reveals that the AP algorithm shows an artifact of local convergence of the standard deviation several times during the iterations, at stages such as [18,22], [37,39], etc. At the above 2 stages, the standard deviation hardly changed, while the standard deviation of the SCS-AP algorithm kept decreasing. After 50 iterations, the SCS-AP algorithm reached stability and converged to about 0.25, whereas the AP algorithm only converged to 0.5 after 80 iterations, so the convergence performance of the SCS-AP algorithm was better. This is mainly because after CS optimisation, the AP algorithm is able to obtain a better Preference parameter P, which saves time for the subsequent clustering iterations and achieves better standard deviation values.

4.4. **Profile performance of different clustering algorithms.** In the following, SCS-AP clustering is compared with other commonly used recommendation clustering algorithms, including hierarchical clustering [29], K-means clustering [30] and PSO-K-means clustering [31], and the test results are shown in Figure 5. A higher mean value of *Silt(t)*

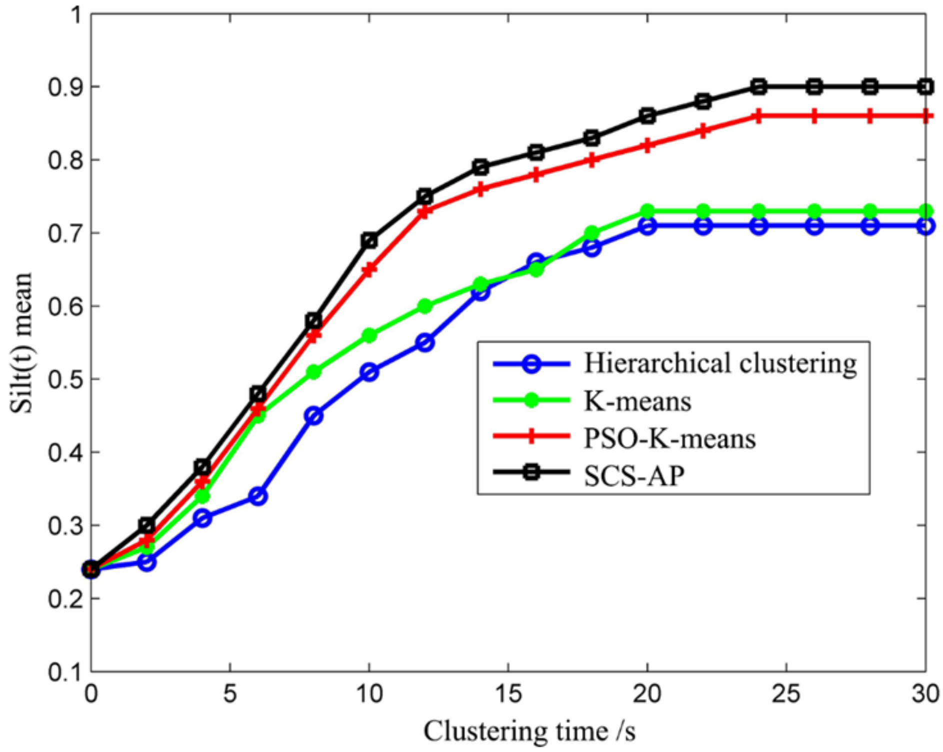

Figure 5. Silhouette performance of different clustering algorithms

for the samples indicates a higher concentration of nodes within the cluster and better clustering. It can be seen that the SCS-AP clustering algorithm has the best profile performance, with a value of 0.9 at stability. PSO-K-means clustering is slightly lower than SCS-AP clustering. Hierarchical clustering was the worst at around 0.7. In terms of clustering time, hierarchical and K-means clustering were the best, obtaining stable clustering results within 20 s. Both PSO-K-means and SCS-AP took 24 s to converge, mainly because both clustering processes required multiple iterations. As shown in Figure 6, for point queries, each of the three indexing methods was used and the Hilbert-R-tree index was found to be the best and the most efficient for queries.

4.5. **Test results of the recommendation model.** The comprehensive performance of the recommendation model is evaluated using the recommendation hit rate P and the recommendation ranking index F.

All companies will be recommended to students according to the weighted ranking results. Since the more inferior ranked companies are not very meaningful for students to recommend, only the top N companies should be recommended. There is a problem with the choice of the N value. If the N value is too large, it will defeat the original purpose of the recommendation model. If the N value is too small, the recommended companies may fall outside the recommendation interval, resulting in a failed recommendation (reduced accuracy). Therefore, it is necessary to choose a reasonable N value according to the accuracy of the recommendation.

A larger value of P indicates higher accuracy. The smaller the value of the recommendation ranking index F, the higher the accuracy. The recommendation ranking index F is calculated as shown below.

$$F = \frac{\sum_{i=1} f_i}{M}(i = 1, 2, 3 \ldots) \tag{33}$$

The variation curve of the recommended hit rate P is shown in Figure 6. The test shows that the highest hit rate P is achieved at the damping factor $\phi = 0.86$. When N is 20, 25 and 30, the hit rate P is 0.57, 0.66 and 0.65 respectively. At the same time, we find that the hit rate P does not change much when N is 25 and 30, while it decreases significantly when N is 15. Therefore, $N = 25$ is a more reasonable choice. The change

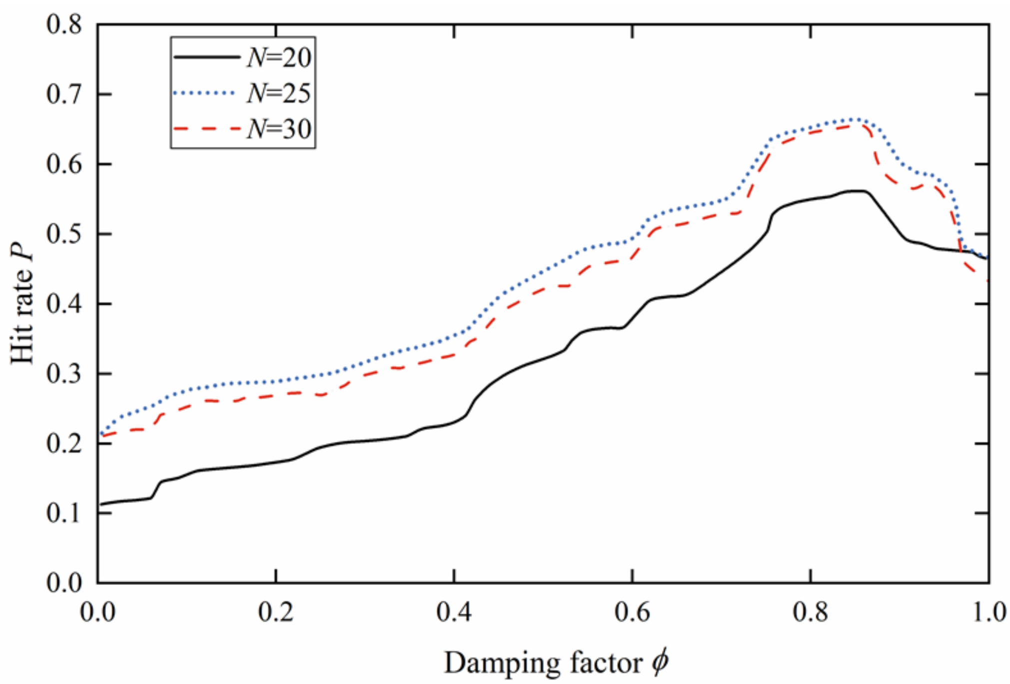

Figure 6. Variation curve of the recommended hit rate P

curve of the recommendation ranking index F is shown in Figure 7. The test shows that the recommended ranking index F does not change much when the recommended number N is 20, 25 and 30 respectively, so the ideal N value is finally determined to be 25. When the damping factor $\phi = 0.79$, the recommended ranking index F is at least 5.4. When the parameter $\phi = 0.86$, the recommended ranking index F is 5.8. In these two cases, the difference in recommended ranking is not significant, and considering that the hit rate P should be as large as possible. is as large as possible, so this work sets the damping factor ϕ to 0.86, which satisfies a hit rate of 0.66 and a recommended ranking of 5.8.

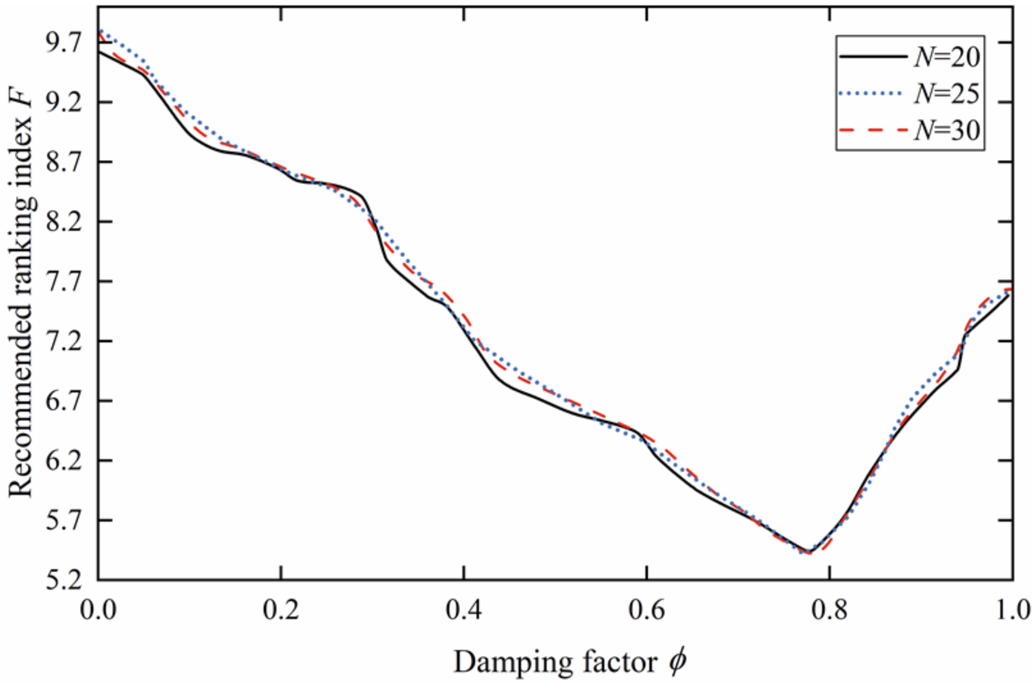

Figure 7. Curve of change in recommendation ranking index F

5. **Conclusion.** An employment recommendation model based on SCS-AP clustering was proposed. By analyzing the shortcomings of the traditional AP clustering algorithm, it is proposed to combine the similarity obtained by the SimRank algorithm with AP clustering to solve the effectiveness of the clustering distance. To address the problem that the performance of the traditional AP clustering algorithm is highly dependent on the Preference parameters, this work also proposes to adopt the Cuckoo Search algorithm to optimise the Preference parameters of AP clustering in order to improve the applicability of the AP algorithm in employment recommendation analysis. The simulation results show that the proposed intelligent recommendation model has a hit rate of 0.66 and a recommendation ranking index of 5.8, which can provide satisfactory personalised recommendation services in a massive data environment. Subsequent research will focus on how to speed up the iteration of the SCS-AP algorithm to further improve the operational efficiency of the recommendation model.

REFERENCES

[1] X. M. Qian, H. Feng, G. S. Zhao, and T. Mei, "Personalized Recommendation Combining User Interest and Social Circle," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763-1777, 2014.

[2] T.-Y. Wu, J. C.-W. Lin, U. Yun, C.-H. Chen, G. Srivastava, X. Lv, "An Efficient Algorithm for Fuzzy Frequent, Itemset Mining," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 5787-5797, 2020.

[3] Y. H. Tian, B. Zheng, and Y. F. Wang, "College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm," *Procedia CIRP*, vol. 83, pp. 490-494, 2019.

[4] T.-Y. Wu, C.-W. Lin, Y. Zhang, C.-H. Chen, "The Grid-Based Swarm Intelligence Algorithm for Privacy-Preserving Data Mining", *Applied Science*, vol. 9, no. 4, 774, 2019.

[5] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 21-48, 2017.

[6] A. L. H. P. Shaik, M. K. Manoharan, A. K. Pani, R. R. Avala, C.-M. Chen, “Gaussian Mutation–Spider Monkey Optimization (GM-SMO) Model for Remote Sensing Scene Classification,” *Remote Sensing*, vol. 14, no. 24, 6279, 2022.

[7] C.-M. Chen, S. Lv, J. Ning, J. M.-T. Wu, “A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem,” *Symmetry*, vol. 15, no. 1, 124, 2023.

[8] J. L. Katzman, U. Shaham, and A. Cloninger, “DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network,” *BMC Medical Research Methodology*, vol. 18, no. 1, pp. 1-12, 2018.

[9] N. Helberger, K. Karppinen, and L. D’Acunto, “Exposure diversity as a design principle for recommender systems,” *Information, Communication & Society*, vol. 21, no. 2, pp. 191-207, 2016.

[10] J. Möller, D. Trilling, N. Helberger, and B. van Es, “Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity,” *Information, Communication & Society*, vol. 21, no. 7, pp. 959-977, 2018.

[11] M. Srifi, A. Oussous, A. Ait Lahcen, and S. Mouline, “Recommender Systems Based on Collaborative Filtering Using Review Texts -A Survey,” *Information*, vol. 11, no. 6, 317, 2020.

[12] D. H. Wang, Y. C. Liang, D. Xu, and X. Y. Feng, “A content-based recommender system for computer science publications,” *Knowledge-Based Systems*, vol. 157, pp. 1-9, 2018.

[13] S. Kant, T. Mahara, V. Kumar Jain, D. Kumar Jain, and A. Kumar Sangaiah, “LeaderRank based k-means clustering initialization method for collaborative filtering,” *Computers & Electrical Engineering*, vol. 69, pp. 598-609, 2018.

[14] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy, “An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit,” *Information Processing & Management*, vol. 75, no. 2, 102034, 2019.

[15] P. A. Karegar, “Wireless fingerprinting indoor positioning using affinity propagation clustering methods,” *Wireless Networks*, vol. 24, no. 8, pp. 2825-2833, 2017.

[16] J. Wang, Y. Gao, K. Wang, A. Sangaiah, and S.-J. Lim, “An Affinity Propagation-Based Self-Adaptive Clustering Method for Wireless Sensor Networks,” *Sensors*, vol. 19, no. 11, 2579, 2019.

[17] C. Gherbi, Z. Aliouat, and M. Benmohammed, “An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks,” *Energy*, vol. 114, pp. 647-662, 2016.

[18] G.-H. Jung, S. Yoo, and Q-Han. Park, “Measuring the optical permittivity of two-dimensional materials without a priori knowledge of electronic transitions,” *Nanophotonics*, vol. 8, no. 2, pp. 263-270, 2018.

[19] A. Varghese, S. Narasimhan, and N. Bhatt, “A Priori Parameter Identifiability in Complex Reaction Networks,” *IFAC- PapersOnLine*, vol. 51, no. 15, pp. 760-765, 2018.

[20] P. Symeonidis, L. Kirjackaja, and M. Zanker, “Session-based news recommendations using SimRank on multi-modal graphs, ” *Expert Systems with Applications*, vol. 180, 115028, 2021.

[21] Y. M. Han, C. Long, and Z. Q. Geng, “A novel DEACM integrating affinity propagation for performance evaluation and energy optimization modeling: application to complex petrochemical industries,” *Energy Conversion and Management*, vol. 183, pp. 349-359, 2019.

[22] S. Subedi, H.-S. Gang, N. Y. Ko, S.-S. Hwang, and J.-Y. Pyun, “Improving Indoor Fingerprinting Positioning with Affinity Propagation Clustering and Weighted Centroid Fingerprint,” *IEEE Access*, vol. 7, pp. 31738-31750, 2019.

[23] T. Koshimizu, S. Gengtian, H. Wang, and Z. Pan, “Multi-Dimensional Affinity Propagation Clustering Applying a Machine Learning in 5G -Cellular V2X,” *IEEE Access*, vol. 8, pp. 94560-94574, 2020.

[24] C. Wang, ”A sample entropy inspired affinity propagation method for bearing fault signal classification,” *Digital Signal Processing*, vol. 102, 102740, 2020.

[25] M. Fiaschetti, M. Graziano, and B. W. Heumann, ”A data-based approach to identifying regional typologies and exemplars across the urban-rural gradient in Europe using affinity propagation,” *Regional Studies*, vol. 55, no. 12, pp. 1939-1954 , 2021.

[26] H. Rakhshani and A. Rahati, ”Snap-drift cuckoo search: a novel cuckoo search optimization algorithm,” *Applied Soft Computing*, vol. 52, pp. 771-794, 2017.

[27] M. Mareli and B. Twala, ”An adaptive Cuckoo search algorithm for optimisation,” *Applied Computing and Informatics*, vol. 14, no. 2, pp. 107-115, 2018.

[28] D. Chitara, K. R. Niazi, A. Swarnkar, and N. Gupta, ”Cuckoo Search Optimization Algorithm for Designing of a Multimachine Power System Stabilizer,” *IEEE Transactions on Industry Applications*, vol. 54, no. 4, pp. 3056-3065, 2018.

[29] P. K. Kimes, Y. Liu, D. Neil Hayes, and J. S. Marron, "Statistical significance for hierarchical clustering," *Biometrics*, vol. 73, no. 3, pp. 811-821, 2017.

[30] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: a Comprehensive Survey and Performance Evaluation," *Electronics*, vol. 9, no. 8, 1295, 2020.

[31] M. Mahajan, S. Kumar, and B. Pant, "Prediction of Environmental Pollution Using Hybrid PSO-K-Means Approach," *International Journal of E-Health and Medical Communications*, vol. 12, no. 2, pp. 65-76, 2021.