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

Machine Learning Clustering for Collaborative Filtering Recommendation of Large-Scale E-commerce in Cloud Computing

Ling-Mei Han*

School of Business
Henan University of Science and Technology
Luoyang 471000, P. R. China
you1983892@163.com

Yan-Ping Gao

Software college
Henan University of Science and Technology
Luoyang 471000, P. R. China
gaoyp1203@163.com

Jian-Guo Liu

Institute of quantitative social sciences
Harvard University
Cambridge Massachusetts 02138, USA
jianguoliu@fast.harvard.edu

*Corresponding author: Ling-Mei Han
Received April 10, 2023, revised June 26, 2023, accepted August 1, 2023.

Abstract. *With the emergence of new technologies such as intelligent computing, artificial intelligence and e-commerce, the scale of electronic data of various commodities has become larger and larger. In order to deal with such large-scale data effectively, it is very important to study cloud computing system and artificial intelligence recommendation system. In order to achieve efficient and accurate recommendation on the cloud computing platform, this paper proposes an intelligent recommendation algorithm based on machine learning clustering model. In order to improve the performance of artificial intelligence recommendation system, an optimized K-means clustering algorithm is adopted to realize data classification. Then intelligent recommendation is completed through collaborative filtering. In order to improve the recommendation efficiency, clustering and recommendation are completed by using multiple nodes of Spark platform. Firstly, the K-means clustering model of users and resources is established. Then, the wolf pack algorithm(WPA) is used to optimize the initial category center points to improve the clustering accuracy. Secondly, according to the category attributes of users and resources, the user-resource scoring data is obtained, so as to establish a collaborative filtering intelligent recommendation model. Finally, the recommendation model is deployed to the Spark platform to realize the distributed operation of clustering and intelligent recommendation. The experimental results show that more accurate recommendation performance can be obtained by setting the number of clustering centers reasonably. Because of making full use of the parallel computing advantages of Spark platform, this work has high real-time performance.*
Keywords: Recommendation system; Artificial intelligence; Machine learning clustering; Collaborative filtering; Wolf pack algorithm

1. **Introduction.** With the emergence of new technologies such as intelligent computing, artificial intelligence and e-commerce, the scale of electronic data of various commodities has become larger and larger. In order to deal with such large-scale data effectively, it is very important to study the artificial intelligence recommendation system. Cloud computing has great advantages in storing and processing large-scale data. Cloud computing can provide storage resources and computing resources to users through services, and flexibly allocate service performance according to users' needs [1,2,3,4]. Artificial intelligence recommendation system can help users find their interesting content from a large amount of information. There is a natural interaction between artificial intelligence recommendation system and cloud computing [5,6]. Due to the increasing scale of data, it is more and more difficult for recommendation system to access these data. At the same time, how to quickly calculate the recommendation results has become an urgent problem to be solved.

We are in the age of data, where huge amounts of electronic data are being generated every moment of every day. Especially since the Internet has entered the Web 2.0 era [7,8], the Internet has made everyone a creator and user of data. Various types of electronic digital devices generate huge amounts of electronic data, such as digital cameras, smartphones, tablet computers, etc. The emergence of new projects such as "metaverse" has led to a rapid growth of electronic data on the Internet. In recent years, along with the rapid development of information technology such as cloud computing, big data, the Internet of Things and artificial intelligence and the transformation of traditional industries into digitalisation, the amount of data has grown geometrically. According to the Internet Data Center (IDC), the total amount of global data is expected to reach 44 ZB by 2020. how to obtain the content of our interest from large-scale data has been a hot issue for research [9]. An important research direction to solve this problem is recommendation systems. Recommender systems can help users select content that may be of interest to them from massive data based on their information and behaviours, such as gender, age, preferences and history [10,11]. With the continuous collection of user information and behavioural data, recommendation systems are constantly improving themselves to achieve accurate recommendations [12,13]. Recommender systems have important applications in e-commerce, social networks, search engines and Internet advertising and marketing.

Cloud computing platforms centralise computing processes in the cloud, reducing the load on user endpoints and allowing clients to focus on interactive technologies. Cloud computing applications have the ability to be deployed quickly and reduce hardware resource overheads. Equipping a recommendation system on a cloud computing platform has natural advantages.

The rest of the paper is organised as follows. Related works are presented in Section 1. In Section 2, the basic knowledge of cloud-based recommendation systems was studied in detail, while Section 3 provides the proposed recommended collaborative filtering method. Section 4 provides the experimental results and analysis. Finally, the paper is concluded in Section 5.

1.1. **Related Work.** The aim of this research is to build fast, large-scale recommender systems through emerging cloud computing platforms and use intelligent computing and clustering algorithms to achieve accurate recommendation results. Firstly, the distributed data storage on cloud computing platforms provides theoretically unlimited data storage capacity for recommendation systems. The virtualised storage management on the cloud provides efficient data throughput for the recommendation system. Both of these allow the recommendation system to read massive amounts of training data and thus provide quality recommendations. Secondly, the distributed computing power of the cloud computing platform provides the recommendation engine with fast responsiveness to satisfy the various personalised recommendation requirements of a large number of users.

Cloud computing is an emerging and popular concept, the core part of which is distributed computing [14,15]. Distributed computing can divide complex problems into many smaller problems and eventually aggregate the results to achieve high performance computing. Cloud computing allows storage and computing resources to be made available to users as a service, with service performance allocated as elastically as the user needs. Cloud computing platforms can provide dynamic provisioning, configuration, reconfigure and deprovisioning services. IDC defines cloud computing as [16]: Cloud computing is a new model of software development and application that provides customised services and solutions in real time over the Internet. Thus we can see from this that cloud computing provides an elastic IT service. Cloud computing obtains high computing power as well as massive storage capacity through large-scale clusters. The working principle of cloud computing is shown in Figure 1.

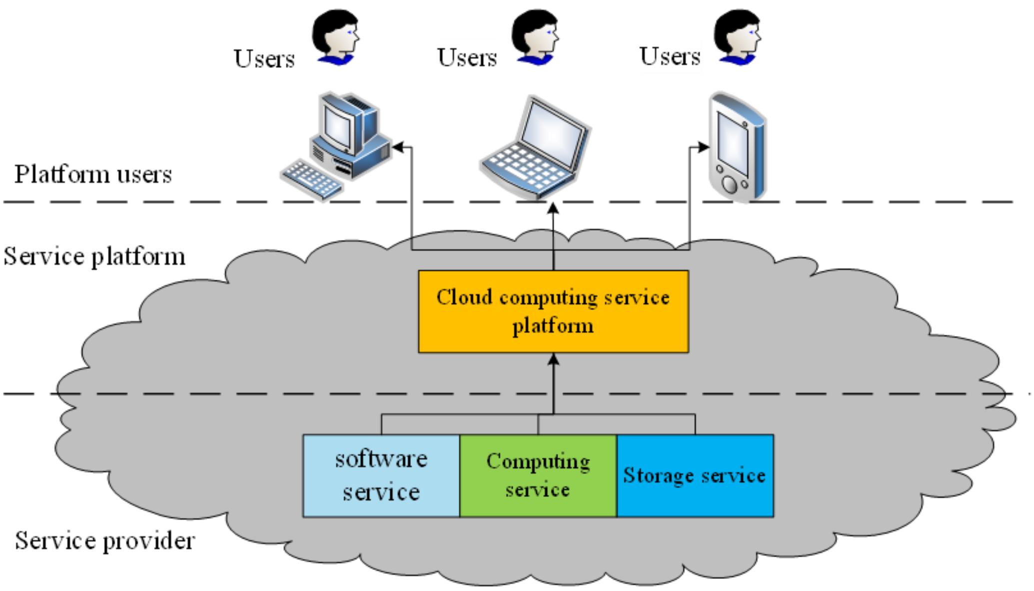

FIGURE 1. How cloud computing works

Figure 1. How cloud computing works

The rapid growth of data volumes on cloud computing platforms is leading to a gradual change in the way users access online services. Most web recommendation systems will obtain the similarity relationship between users and resources based on similarity calculation, so as to recommend the resources with the highest similarity for users. This kind of intelligent recommendation system based on data clustering is the mainstream model of current web recommendation systems [17,18]. Due to the clustering calculation involved, the accuracy and efficiency of recommendations are affected when there are more categories and the data size is large. Therefore, multi-class clustering and large-scale computing efficiency are the key problems to be solved in artificial intelligence recommendation system.

At present, the most typical machine learning clustering algorithm is K-means algorithm based on partition. Intelligent recommendation techniques based on the K-means algorithm have been studied more. Liu et al. [19] used a deep model to implement K-means clustering mining and used it for online learning resource recommendation. This work was able to make resource recommendations based on the user's behavioural habits over a short period of time. However, the accuracy of the recommendations was not high due to the limitations of the model. Liu et al. [20] used a neural network technology to improve the accuracy of K-means clustering in order to provide advertising pushes to online users of a website. However, limited by the initial value selection, mild misclassification also occurred in the recommended results. He et al. [21] achieved good results by using a clustering mining algorithm to achieve song recommendations for different users based on user history. However, the small amount of training data led to some limitations in the applicability of this recommendation system. Shu and Jin [22] proposed a piecewise parameterization method of control variables to obtain the optimal control parameters. Although all the above methods have improved the accuracy of clustering to a certain extent, they have not effectively used the cloud computing platform to improve the recommendation efficiency.

1.2. **Motivation and contribution.** In order to enable an efficient and accurate recommendation system on a cloud computing platform, this paper proposes a recommendation algorithm based on a machine learning clustering model. For the intelligent recommendation system to work better, an optimised K-means clustering method is employed to achieve data classification.

Wolf pack algorithm(WPA) is a new type of population intelligence optimisation algorithm recently proposed to solve the global optimisation and local extreme value problems. Compared to DE, PSO, ACO and GA, WPA has strong convergence performance, simple structure, few parameters to be adjusted and easy to realize. Most of the existing K-means clustering preprocessing needs to manually set the initial clustering center. However, the K-means algorithm based on WPA makes the k-means algorithm unaffected by the initial clustering center. Therefore, this paper attempts to introduce WPA to optimise the K-means clustering to enhance the precision of multi-category clustering, and to introduce the parallelization of Spark engine to improve the clustering efficiency.

The main innovations and contributions of this paper include:

(1) An in-depth analysis of the partition-based K-means algorithm is made, and several drawbacks of the K-means clustering are described. This work suggests using WPA to improve the original category cluster centers of the K-means method in order to address these drawbacks and increase clustering accuracy.

(2) In order to solve the problem that the traditional mode of clustering algorithm is difficult to handle large-scale data, this paper parallelizes the design of the K-means algorithm and collaborative filtering algorithm so that it can run properly in Spark's MapReduce framework.

2. Artificial intelligence recommendation system based on cloud computing.

2.1. **Recommender systems in a distributed environment.** The research on recommendation systems in distributed environments is mainly based on Mobile Agent [23], which is a computational entity that can function continuously and autonomously in a certain environment. The Mobile Agent technology makes the recommendation system suitable for complex distributed environments and has a high degree of flexibility.

In a Mobile Agent-based recommendation system, the Agent is divided into the following categories [24]: (1) the User Agent, which is responsible for the input and output interaction with the user; (2) the Control Agent, which is responsible for the coordination of the whole recommendation process; and (3) the Recommendation Agent, which is responsible for the discovery and invocation of the recommendation service. A typical mobile Agent-based recommendation system flow is shown in Figure 2.

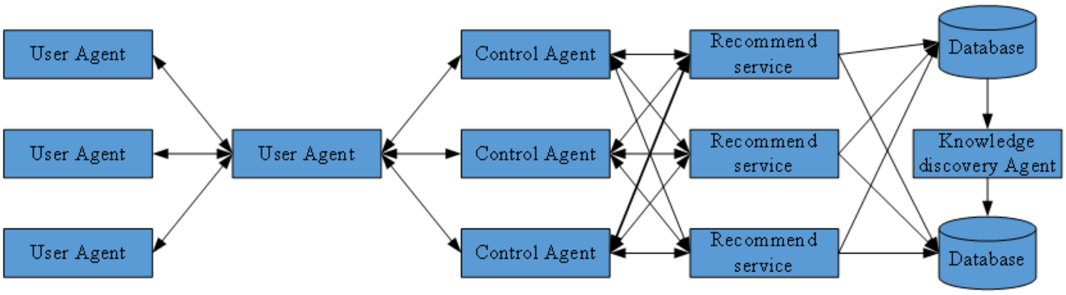

Figure 2. A typical process of recommender system based on Mobile Agent

Mobile Agent is an advanced artificial intelligence technology with good applicability in cloud computing. mobile agent embodies the social collaboration of computational intelligence and can effectively simulate the group intelligence of computers.

2.2. **MapReduce.** MapReduce is a distributed programming model software architecture proposed by Google. Firstly, the programming model for cloud computing must be applicable to the distributed operating environment of a cluster. Secondly, in order to achieve the user requirements, we need to enable developers to directly observe distributed computing, task scheduling, and exception handling in the underlying cloud computing [25]. The MapReduce programming model requires the definition of two functions: Map and Reduce, as shown in Figure 3. The MapReduce process requires a TaskTracker, which

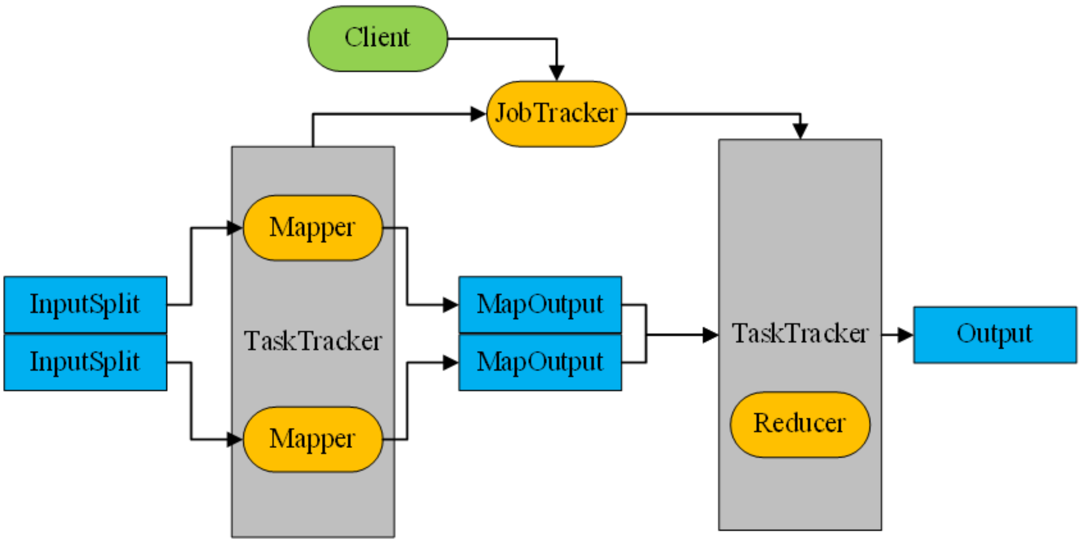

Figure 3. MapReduce process

stores the working state of all Maps and Reduces. TaskTracker assigns Map and Reduce tasks to the nodes in the system and monitors them for their work. When a Mapper job is complete, the intermediate results generated by that Mapper will be stored on the disk in this node and the location of the results will be sent to TaskTracker. TaskTracker notifies Reducer that need these intermediate results step by step. The required data is then read from Mapper' s disk by the Reducer.

The MapReduce programming model is highly fault tolerant, in that when a Mapper or Reducer fails, the TaskTracker will exclude that node from the system. The keys processed on the node will be reassigned to other Mappers or Reducers for execution. When the failed node is restored to normal, JobTracker reassigns new tasks to that node.

2.3. **Spark architecture for cloud computing.** Apache Spark is an open source distributed computing framework that was originally developed by the AMP Lab at the University of California, Berkeley [26,27]. Each step of Hadoop MapReduce must serialize data written to a distributed file system, resulting in a significant reduction in operational efficiency. However, Spark stores intermediate results on memory whenever possible, greatly increasing the speed of computation. Spark extends the MapReduce model to allow developers to develop complex multi-step data channels using Directed Acyclic Graph (DAG). In addition, Spark supports in-memory data sharing across directed acyclic graphs so that different jobs can work together on the same data.

Spark, as a common engine for large-scale data processing, uses a master-slave node management model to jointly perform data processing tasks. The main structure of Spark is shown in Figure 4. In addition to this master-slave model, the Spark platform has the

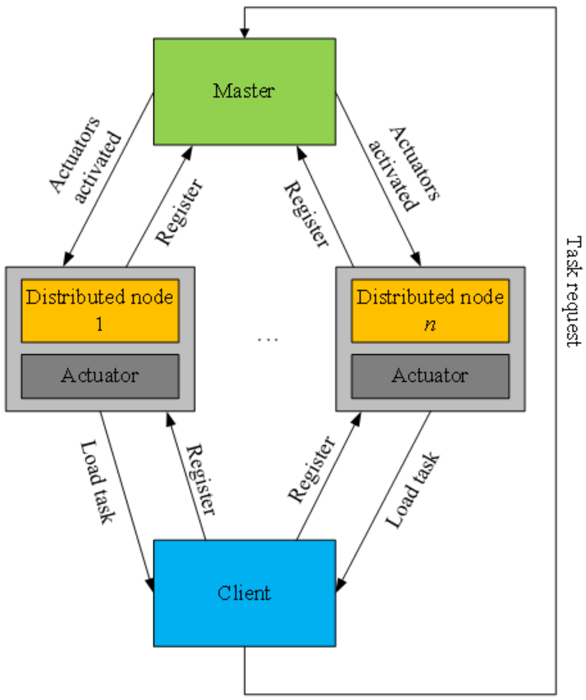

Figure 4. Spark structure

advantage that most of the data operations are done in Resilient Distributed Datasets (RDDs) in memory. This approach greatly improves the efficiency of data access. By using the Spark platform for clustering and recommendation of large-scale data, the problem of inefficient computing caused by frequent iterations in clustering is solved, and the problem of real-time intelligent recommendation is also solved.

3. Collaborative filtering recommendation based on machine learning clustering.

3.1. **Analysis of the K-means algorithm.** The K-means algorithm is a division-based machine learning clustering algorithm [28] that is capable of dividing a dataset into k sets specified by the user.

If the objects in the dataset can all be abstracted into points on an m-dimensional space (m being the number of object attributes considered in the clustering), then clustering using the K-means algorithm is well suited. The result of clustering is that the n (total number of objects in the dataset) objects on the m-dimensional space are divided into k different groupings c_j ,$C = \{c_j | j = 1, 2, ..., k\}$. The distance between any two points i and j in the K-means clustering space is S_{ij} .

$$S_{ij} = \begin{cases} d_{ij}, i \neq j \\ 0, i = j \end{cases} \tag{1}$$

Assuming that the centroid x_i contains n attributes, then x_i can be expressed as $(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in})$. Assuming that the point to be clustered is x_i, the distance between and x_j is d_{ij}.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{2}$$

The distance from the centroid of all sample points to be clustered can be calculated according to Equation (2). The distance d_{ij} is used to determine whether x_i and x_j belong to a class. The clustering objective function is then established based on the distances and the minimum value of the clustering objective function is solved.

$$e = \sum\nolimits_i \left\| x_i - \sum S_{ij} x_j \right\|^2 \tag{3}$$

The final objective function obtained is shown as follow.

$$\begin{aligned} &\min \varepsilon \\ &s.t. \quad \sum_j S_{ij} = 1, S_{ij} \geq 0 \end{aligned} \tag{4}$$

When executing the K-means algorithm, the number of initial groupings k and the initial clustering centres of each grouping need to be set artificially. However, in practical applications, there is no universal solution for the selection of the initial values of the K-means algorithm [29]. the efficiency of K-means clustering depends on the dimensionality of the points to be clustered and the amount of sample data. In general, the accuracy and efficiency of clustering decreases as the number and dimensionality of the samples to be clustered increases, so that using only the K-means algorithm is not feasible when dealing with large-scale data. Because the selection of the initial centroid of the K-means clustering method is also important and it affects the efficiency of the clustering, some improvements to the K-means algorithm are needed. Therefore, this paper proposes an idea of pre-processing the data set to derive the k-value, the initial clustering centre.

Data pre-processing allows simple analysis and operations to be performed on the dataset to extract the clustering grouping information implied by the dataset. By pre-processing the data, we can derive the k-values and k initial cluster centres required by the K-means algorithm. There are various ways of pre-processing the data, one of which is to select a population intelligence optimisation algorithm to first compute the dataset. The result of the calculation is used to determine the k-value. Based on this solution, the Wolf pack algorithm is ideally suited to be used for data pre-processing due to its powerful global optimisation seeking capability.

3.2. **Wolf pack algorithm.** The principle of the Wolf pack algorithm (WPA) is shown in Figure 5. Let the total number of wolves in a given pack be N and the data dimension be D , then the position of the i-th wolf is $X_i = (x_{x1}, x_{x2}, \ldots, x_{id}, \ldots, x_{iD})$, where $1 \leq i \leq N$, $1 \leq d \leq D$.

$$x_{id} = x_{\min} + rand * (x_{\max} - x_{\min}) \tag{5}$$

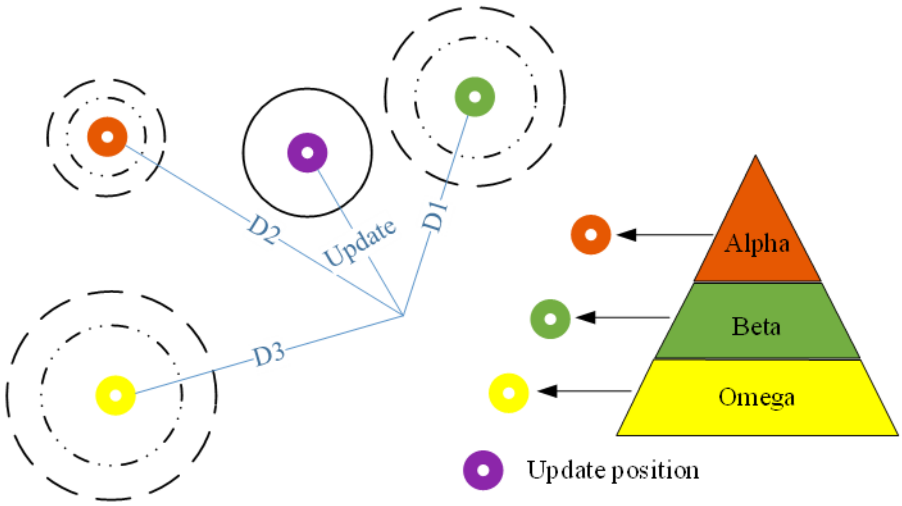

Figure 5. Principle of the WPA

where $x_{\max}$ and $x_{\min}$ are the upper and lower bounds of the d-dimensional space, respectively. $rand$ is a random value in the range [0,1].

The wolf with the highest fitness value was selected as the Alpha wolf and the wolves around it were the Beta wolves. The number of Beta wolves was T_{num} and there was a limit to the number of steps they could move.

$$Step_G(d) = \frac{|\max_d - \min_d|}{S} \tag{6}$$

where $1 \leq d \leq D$, S are adjustable weight constants. The Beta wolf's position is updated in the manner shown as follow.

$$x_{i,d}^G = x_{i,d} + \sin(\frac{2\pi}{h}) \times Step_G(d) \tag{7}$$

where $i = 1, 2, .., T_{num}$, h is the number of directions of movement.

The remaining wolves in the pack are Omega wolves, which also have some limitations on their movement paces.

$$Step_B(d) = \frac{2 \times |\max_d - \min_d|}{S} \tag{8}$$

Omega's position is updated in the manner shown as follow.

$$x_{i,d}^B = x_{i,d} + Step_B(d) \cdot \frac{[s_{i,d} - x_{i,d}]}{|s_{i,d} - x_{i,d}|} \tag{9}$$

where $i = 1, 2, .., N - T_{num} - 1$. $[x]$ means the largest integer that does not exceed the real number x. $s_{i,d}$ is the distance between the i-th wolf and the Alpha wolf in d-dimensional space, $s_{i,d} \in D_d$.

$$D_d = \frac{1}{D \times \omega} \times \sum_{d=1}^{D} |\max_d - \min_d| \tag{10}$$

where ω is the distance factor constant. When the pack finds its prey and the Alpha wolf gives the command to surround, the Alpha wolf's movement step is updated in the way shown as follow.

$$Step_W(d) = \frac{|\max_d - \min_d|}{2 \times S} \tag{11}$$

The Alpha Wolf's position update is calculated as follows.

$$x_{i,d}^{W} = x_{i,d} + \lambda \cdot Step_W(k) \cdot |s_d - x_{i,d}| \tag{12}$$

The overall flow of WPA is shown in Figure 6.

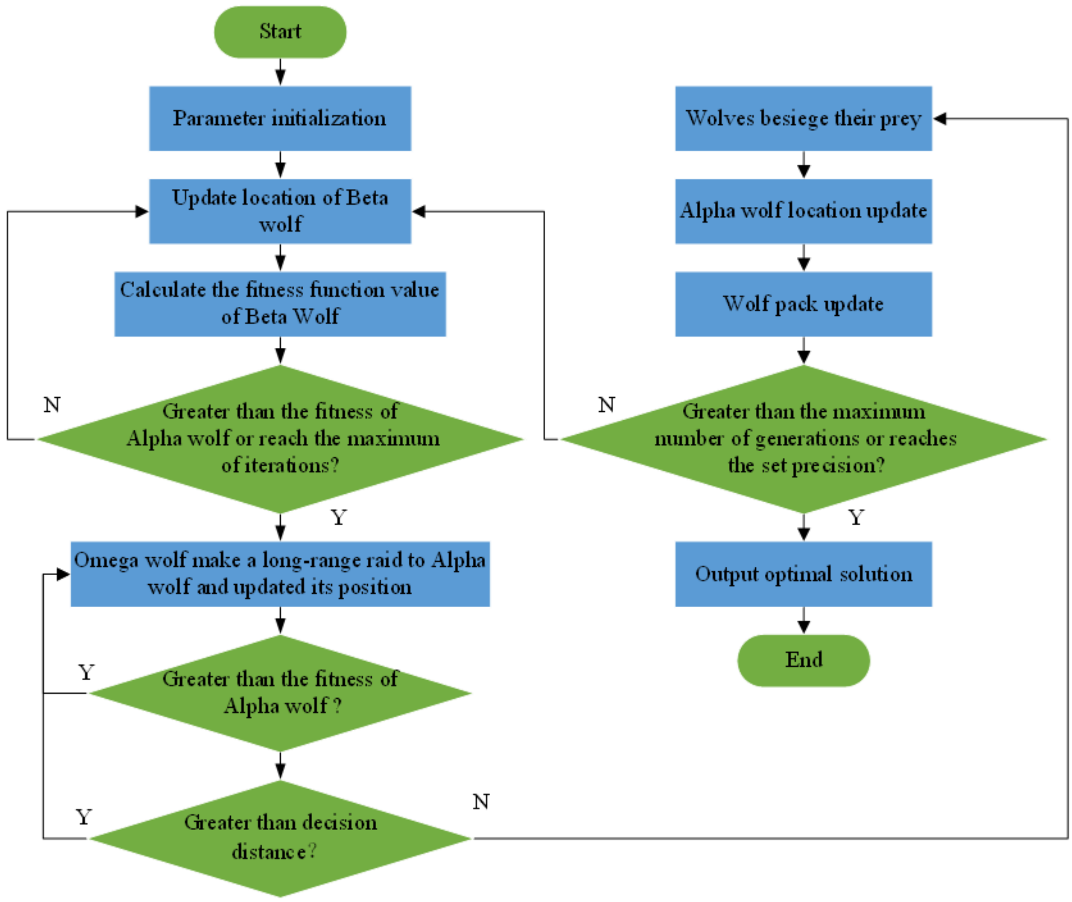

Figure 6. Flow of the WPA

3.3. **Collaborative filtering recommendations.** It is possible to get the user's ranking of each resource or service to be recommended after clustering using the WPA-K-means method. Then, practical recommendations are generated using the collaborative filtering process. Collaborative filtering-based recommendations are the most widely used recommendation method in recommendation systems. The basic principle of collaborative filtering is to find the correlation between users based on their recommended behaviour towards the set of recommended objects, and make recommendations accordingly. An example recommendation process is shown in Figure 7. The set of recommended users is $U = (u_1, u_2, \ldots, u_m)$, the set of resources to be recommended is $I = (i_1, i_2, \ldots, i_n)$, and $r_{m,n}$ denotes the rating of the n-th resource by the m-th user, then the similarity relation between users a and b is $sim(a,b)$.

$$sim(a,b) = \frac{\sum_{p=1}^{n} (r_{a,p} - \bar{r_a})(r_{b,p} - \bar{r_b})}{\sqrt{\sum_{p=1}^{n} (r_{a,p} - \bar{r_b})^2 (r_{b,p} - \bar{r_b})^2}} \tag{13}$$

where $\bar{r_a}$ and $\bar{r_b}$ represent the average rating of all resources by users a and b respectively.

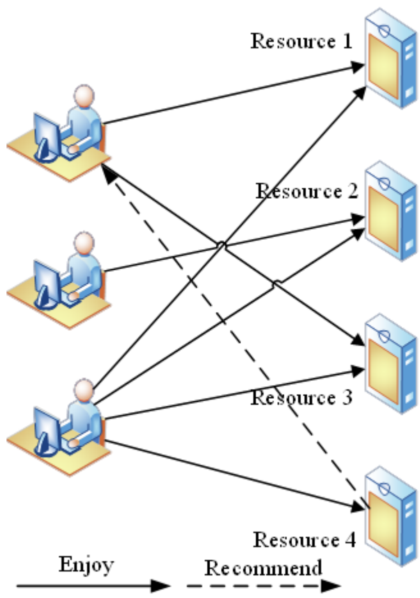

Figure 7. Example of a recommendation based on collaborative filtering

In the process of collaborative filtering recommendations, in addition to the analysis of similarities between users, the most important thing is the need to solve for the user's rating of the resource. The rating of resource k by user j is *score(j,k)*.

$$score(j,k) = \bar{r_a} + \frac{\sum_{p=1}^{n} sim(j,k)^2 (r_{b,p} - \bar{r_b})}{\sum_{p=1}^{n} sim(j,k)} \tag{14}$$

Based on the rating of resources, we will recommend resources with high ratings for users, thus completing intelligent recommendations. The intelligence of the work in this paper is reflected in the fact that instead of all users rating all resources one by one, WPA-K-means clustering is used to predict the users' rating values for resources from the behavioural data of users accessing the network.

3.4. **Improvements to MapReduce-based Recommender Systems.** The most important computational elements in the recommendation system proposed in this paper are the collaborative filtering algorithm and the WPA-K-means clustering algorithm. However, both of these algorithms are serially computed in a sequential order. Therefore, MapReduce-based improvements to these two algorithms are needed.

Firstly, the two most important steps in the collaborative filtering algorithm are the calculation of item similar degree and the calculation of predicted ratings for the target user's unrated items based on item similar degree. These two most important steps are serial. To perform parallel computation on the MapReduce-based Spark platform, we use a MapReduce job to compute the similarity of items. When the results are obtained, a MapReduce job is then used to compute the predicted rating values for the user's unrated items. When the target user needs a recommendation, we calculate the predicted score of each item in the map phase based on the user's recommendation score and the similarity list of each item. The results are collected in the reduce phase and the Top N recommended items are returned, as shown in Table 1 and Table 2.

Table 1. Map phase.

Input date	Output date
<element 1, ((element 2, similarity), (element 3, similarity))>	<NA, (element 1, predicted score)>
<element 2, ((element 1, similarity), (element 3, similarity))>	<NA, (element 2, predicted score)>
<element 3, ((element 1, similarity), (element 2, similarity))>	<NA, (element 3, predicted score)>

Table 2. Reduce phase.

Input date	Output date
<NA, (element 1, predicted score)> <NA, (element 2, predicted score)> <NA, (element 3, predicted score)>	<NA, sorted list of recommended elements

The collaborative filtering algorithm on the Spark platform goes through a total of four MapReduce processes. Each MapReduce process is a parallel computational process. Since the mathematical basis of the collaborative filtering algorithm is not changed, the MapReduce-based collaborative filtering algorithm has the same recommendation effect compared to traditional collaborative filtering algorithms. However, MapReduce-based collaborative filtering algorithms have the ability to run in large clusters and can efficiently perform recommendation analysis on large data sets.

Secondly, in order to realize the parallelization operation of WPA-K-means clustering, a new Map Reduce function is designed in this paper, as shown in Table 3. In the Map phase, the nearest cluster centre is calculated based on the input data points, and the data points are given a new group identity. In the Reduce phase, a new cluster centre is calculated based on the output of the Map function.

3.5. **WPA-K-means clustering and recommendation process.** First, analyse the client's intelligent recommendation task requirements. Then, the Spark platform is built and distributed nodes of appropriate size are deployed. Then, a machine learning clustering model is built and the initial clustering centroids of the K-means algorithm are optimised by WPA. The clustering results are used to obtain user-attribute score data. Finally, collaborative filtering is used to complete the intelligent recommendation, and the intelligent recommendation process under the Spark platform is shown in Figure 8.

4. **Experimental results and analysis.** In order to verify the recommendation performance of machine learning clustering under Spark platform, the test is carried out on public data sets. The public data set is ZARAUR uniqlo clothing sales data set (https://www.heywhale.com/mw/dataset), as shown in table 4. This data set contains 23,000 online shopping comments and evaluations, which can well verify the recommendation performance of clustering mining. Spark platform consists of one Master node and nine Work nodes, all of which have the same hardware parameters. WPA-K-means algorithm is used to complete clustering, and then product recommendation is completed through collaborative filtering. The simulation environment for all algorithms in this paper is Matlab7.0. The experimental platform is Windows 10 64-bit operating system, and the CPU is i5-4570 processor with 4 GB of memory. The experimental parameter settings are shown in Table 5.

Table 3. Map Reduce functions for WPA-K-means clustering.

Map Reduce functions	Pseudo-code

Map phase	Map_Class{
	map(key, value){
	Sort = 0.
	Dis = Max Value.
	for (int i=1; i<k; i++) {
	tempDis = dis(i, pointer);
	if(tempDis <minDis) {
	Sort =i. minDis = tempDis.
	}
	}
	produce<"Sort", value>.
	}

Reduce phase	Reduce_Class{
	Reduce(key,value){
	Rows = 0.
	Sort = k.
	Records = new Double[Col];
	for(int i = 0; i<Col; i++){
	for(int Rows = 0;Rows<value.length; Rows++){
	Records[i] = value[Rows][i].
	}
	}
	for (int j = 0; j <Col; j++)
	pointer += Records[j].
	produce<key, pointer>.
	}

Table 4. ZARAUR experimental data set.

Sample set	User data	Records of consumption	Sample size
Data1	1956	1683	5 GB
Data2	6042	3680	25 GB
Data3	72001	10000	150 GB
Data4	805543	100000	1 T

4.1. **Recommended performance for different number of clustering centres.** The Root-mean-square error (RMSE) is used to evaluate the performance of the recommendation system. The smaller the RMSE, the higher the accuracy of the recommendation system. The product rating matrix is very sensitive to the number of clustering centres k, which affects the stability of the collaborative filtering recommendation algorithm. Therefore, different values of k were chosen during the testing process to find the optimal number of clustering centres, as shown in Figure 9. It is clear that as the value of k rises, the RMSE value of the recommendation system first lowers and then rises. When the value of k is small, the recommended goods have a large deviation from the actual rating value of users. Data1 and Data2 obtained the smallest RMSE values at k=16, while Data3 and Data4 obtained the smallest RMSE values at k=18. As the

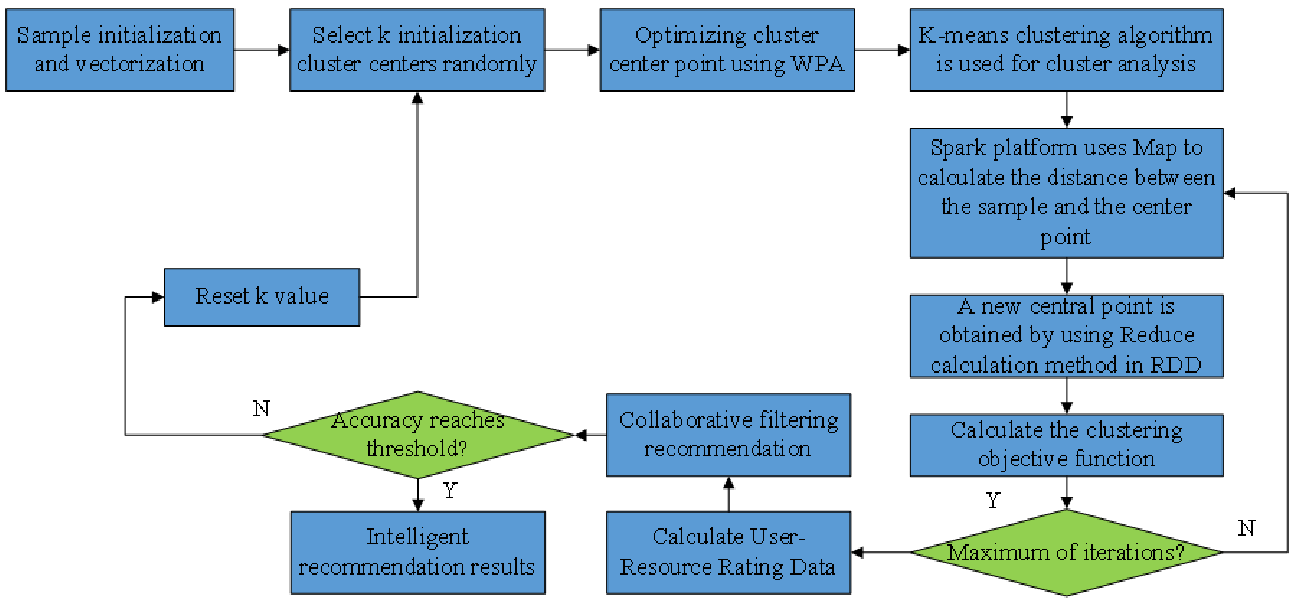

Figure 8. Flow chart of intelligent recommendation under spark platform

Table 5. Experimental parameters.

Parameter	numerical value
x_{max}	0.4
x_{min}	0.2
S	0.6
N	40
D	20
ω	0.2
The maximum number of iterations	2000

value of k continued to increase, the RMSE values gradually increased and the stability of the recommendations became worse. This suggests that the optimal k value for the recommender system is [16,18] for the ZARAUR dataset. Therefore, in the subsequent tests for the ZARAUR dataset, the range of k values was set to [16, 18].

4.2. **Recommended performance of public datasets.** First, k=16 was set and the WPA-K-means method was used. Then, the ZARAUR dataset was trained using the collaborative filtering recommendation algorithm, and the training process was performed on a Spark platform consisting of 10 nodes. The recommendation performance of the public dataset is shown in Table 6.

In terms of recommendation time performance, the difference in capacity is not significant in terms of recommendation time, mainly due to the Spark platform being adopted for parallel computing. For the 10 nodes, the variation in recommendation time due to the difference in capacity of the 4 sample sets is very small.

4.3. **Acceleration Performance of the Spark Platform.** To estimate the impact of the Spark framework on the efficiency of intelligent recommendations, the speed-up factor of parallel recommendations versus serial recommendations was calculated. the

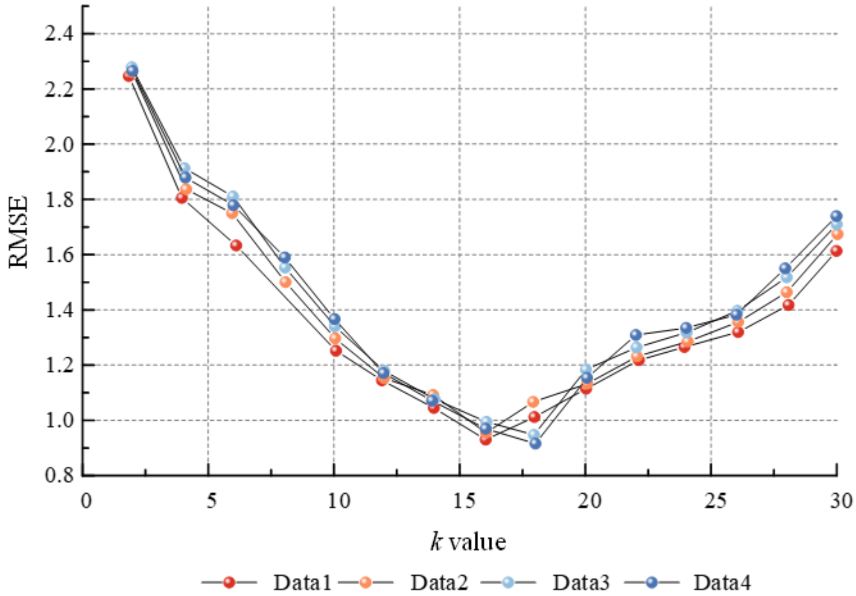

Figure 9. Precision of recommendations for various k values

Table 6. Recommended performance.

Sample set	Accuracy % Minimum value	Average value	Maximum value	Recommended time/s
Data1	86.913	88.539	90.062	1.156
Data2	86.752	88.613	89.435	1.197
Data3	86.913	88.784	89.391	1.213
Data4	86.865	88.728	90.506	1.228

Spark speed-up performance is shown in Table 7.

$$S = \frac{T_a}{T_s} \tag{15}$$

where T_a is the serial recommended time and T_s is the parallel recommended time. It

Table 7. Effect of Spark on intelligent recommendation efficiency.

Data set	Capacity	Number of nodes	Speed-up factor
		1	1.000
Data1	5 GB	5	1.001
		10	1.001
		1	1.000
Data2	25 GB	5	1.013
		10	1.024
		1	1.000
Data3	150 GB	5	6.473
		10	13.139
		1	1.000
Data4	1 T	5	22.287
		10	43.907

is evident that the Spark speed-up factor grows as the number of worker nodes rises. The impact of the number of Worker nodes on the speed-up factor becomes increasingly important as the data set capacity increases. When the sample size of Data1 is 5 GB and the worker nodes reach 10, the speedup ratio only increases by 0.001. When the sample size is 1 T, the parallel recommendation time is reduced by about 43 times compared to the serial recommendation time. As a result, the recommendation effectiveness for big data sets is enhanced by collaborative filtering recommendation based on machine learning clustering on Spark platform.

4.4. **Performance simulation of different recommendation algorithms.** In order to continue to verify the superiority of the algorithms in this paper in intelligent recommendation systems, the SVM algorithm, deep neural network (DNN), XGBoost algorithm and the algorithms in this paper were compared on Data 4 of the ZARAUR dataset, respectively, as shown in Figure 10. It can be seen that the recommendation accuracy of

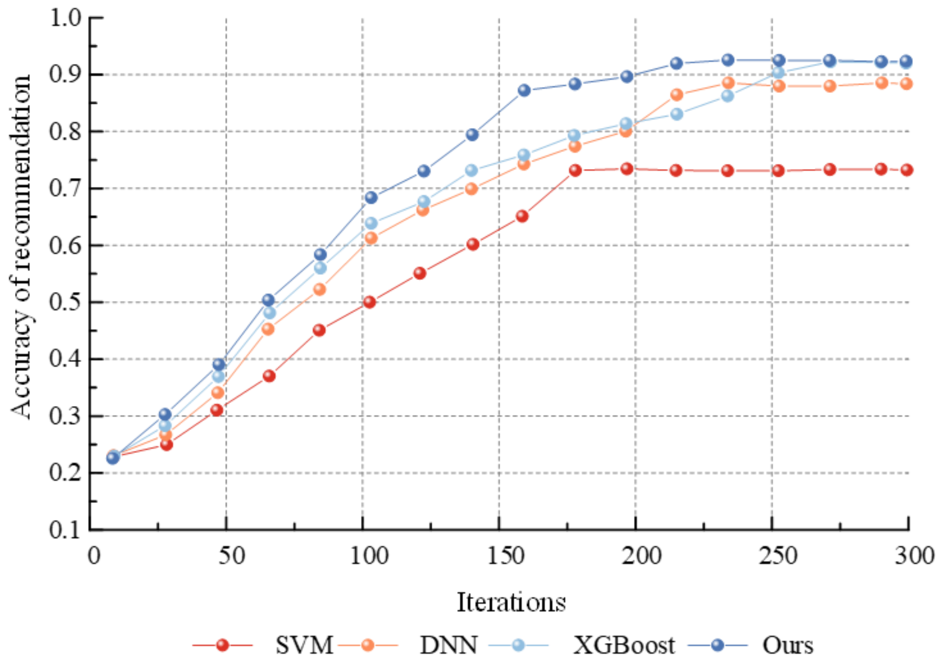

Figure 10. Recommended performance of different algorithms

the four different algorithms increases with the number of iterations at the beginning. When the number of iterations is greater than 200, the recommendation accuracy of the four different algorithms stabilises. Recommendation performance consists of two aspects: accuracy and convergence speed, so it is necessary to analyse these two aspects together. Firstly, in terms of recommendation accuracy, our algorithm ranked first after convergence, with an accuracy rate of over 0.9. The SVM algorithm had the worst recommendation accuracy rate, which was only about 0.7. Secondly, in terms of convergence speed, the SVM ranked first with 190 times, while our algorithm ranked second with about 230 times, and the XGBoost algorithm had the slowest convergence speed. Based on the overall recommendation performance comparison of the four algorithms, this paper ranked high in both accuracy and convergence speed.

5. **Conclusion.** In order to effectively process large-scale electronic data on the cloud computing platform and realize an efficient and accurate artificial intelligence recommendation system, this paper proposes a recommendation algorithm based on machine

learning clustering model. In order to improve the performance of artificial intelligence recommendation system, an optimized K-means clustering algorithm is adopted to realize data classification. Then intelligent recommendation is completed through collaborative filtering. This paper attempts to introduce WPA to optimize K-means clustering algorithm to improve the accuracy of multi-class clustering, and at the same time, introduce multi-node parallel computing on Spark platform to improve clustering efficiency. Because of making full use of the parallel computing advantages of Spark platform, the work in this paper has high real-time performance, so as to obtain the content we are interested in from large-scale data more quickly. However, when WPA algorithm is faced with complex high-dimensional problems, the iterative calculation process has a large amount of calculation, which will be further studied in the future.

Acknowledgement. This work was supported by the Social Practice First-class Curriculum Engineering of Henan Province, China, Cross-border e-commerce school, government and enterprise collaborative innovation and education social practice (No. YuJiao[2021]21980) and the Social Science Planning Annual Foundation of Henan Province, China,Research on the collaborative transformation of digital and green manufacturing of Henan Province (No. 2022BJJ141).

REFERENCES

[1] P. Srivastava and R. Khan, "A Review Paper on Cloud Computing," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 6, pp. 17–20, 2018.

[2] A. Rashid and A. Chaturvedi, "Cloud Computing Characteristics and Services a Brief Review," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 421–426, 2019.

[3] M. N. Birje, P. S. Challagidad, R. H. Goudar, and M. T. Tapale, "Cloud computing review: concepts, technology, challenges and security," *International Journal of Cloud Computing*, vol. 6, no. 1, 32, 2017.

[4] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.

[5] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.

[6] T.-Y. Wu, F. Kong, Q. Meng, S. Kumari, and C.-M. Chen, "Rotating behind security: an enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, no. 1, 36, 2023.

[7] T.-Y. Wu, L. Wang, X. Guo, Y.-C. Chen, and S.-C. Chu, "SAKAP: SGX-Based Authentication Key Agreement Protocol in IoT-Enabled Cloud Computing," *Sustainability*, vol. 14, no. 17, 11054, 2022.

[8] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and Cloud Computing," *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.

[9] H. Xiong, Y. Wang, W. Li, and C.-M. Chen, "Flexible, Efficient, and Secure Access Delegation in Cloud Computing," *ACM Transactions on Management Information Systems*, vol. 10, no. 1, pp. 1-20, 2019.

[10] Q. Mei, H. Xiong, Y.-C. Chen, and C.-M. Chen, "Blockchain-Enabled Privacy-Preserving Authentication Mechanism for Transportation CPS with Cloud-Edge Computing," *IEEE Transactions on Engineering Management*, 2022. [Online]. Available: https://10.1109/TEM.2022.3159311

[11] C.-M. Chen, Z. Tie, E. K. Wang, M. K. Khan, S. Kumar, and S. Kumari, "Verifiable dynamic ranked search with forward privacy over encrypted cloud data," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 2977-2991, 2021.

[12] C. He, D. Parra, and K. Verbert, "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities," *Expert Systems with Applications*, vol. 56, pp. 9–27, 2016.

[13] P. Kumar and R. S. Thakur, "Recommendation system techniques and related issues: a survey," *International Journal of Information Technology*, vol. 10, no. 4, pp. 495–501, 2018.

[14] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 1, pp. 57–74, 2007.

[15] R. Mishra, P. Kumar, and B. Bhasker, "A web recommendation system considering sequential information," *Decision Support Systems*, vol. 75, pp. 1–10, 2015.

[16] H. Ko, S. Lee, Y. Park, and A. Choi, "A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields," *Electronics*, vol. 11, no. 1, 141, 2022.

[17] Y. Tian, B. Zheng, Y. Wang, Y. Zhang, and Q. Wu, "College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm," *Procedia CIRP*, vol. 83, pp. 490–494, 2019.

[18] Y. Cao and Y. Li, "An intelligent fuzzy-based recommendation system for consumer electronic products," *Expert Systems with Applications*, vol. 33, no. 1, pp. 230–240, 2007.

[19] J. Liu, H. Zhang, and Z. Liu, "Research on Online Learning Resource Recommendation Method Based on Wide & Deep and Elmo Model," *Journal of Physics: Conference Series*, vol. 1437, 012015, 2020.

[20] D.-R. Liu, Y.-S. Liao, Y.-H. Chung, and K.-Y. Chen, "Advertisement recommendation based on personal interests and ad push fairness," *Kybernetes*, vol. 48, no. 8, pp. 1586–1605, 2019.

[21] M. He, H. Guo, G. Lv, L. Wu, and H. Ma, "Leveraging proficiency and preference for online Karaoke recommendation," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 273–290, 2019.

[22] Y. Shu and T. Jin, "Stability in measure and asymptotic stability of uncertain nonlinear switched systems with a practical application," *International Journal of Control*, vol. 8, pp. 1–11, 2022.

[23] G. Mehmood, M. Zahid Khan, M. Fayaz, and M. Faisal, "An Energy-Efficient Mobile Agent-Based Data Aggregation Scheme for Wireless Body Area Networks," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 5929–5948, 2022.

[24] D. Shehada, C. Y. Yeun, M. Jamal Zemerly, M. Al-Qutayri, Y. Al-Hammadi, and J. Hu, "A new adaptive trust and reputation model for Mobile Agent Systems," *Journal of Network and Computer Applications*, vol. 124, pp. 33–43, 2018.

[25] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "Big data analytics on Apache Spark," *International Journal of Data Science and Analytics*, vol. 1, no. 3–4, pp. 145–164, 2016.

[26] M. Zaharia, M. J. Franklin, A. Ghodsi, J. Gonzalez, and S. Shenker, "Apache Spark," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[27] D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, "A comparison on scalability for batch big data processing on Apache Spark and Apache Flink," *Big Data Analytics*, vol. 2, no. 1, pp. 43–46, 2017.

[28] M. Belouch, S. El Hadaj, and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using Apache Spark," *Procedia Computer Science*, vol. 127, pp. 1–6, 2018.

[29] R. Shyam, B. Ganesh, S. Kumar, P. Poornachandran, and K. P. Soman, "Apache Spark a Big Data Analytics Platform for Smart Grid," *Procedia Technology*, vol. 21, pp. 171–178, 2015.