International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Aug-2023)

http://dx.doi.org/10.12785/ijcds/140130

Agent-based Approach for the Recommendation and Unsupervised Classification of Enterprise Services in the Cloud

Djihene BOURENANE[1], Nawal SAD HOUARI[2] and Noria TAGHEZOUT[1]

[1]*Department of computer science, Laboratoire d'informatique Oran (LIO),Université ORAN1 AHMED BEN BELLA, Faculté des Science Exactes et Appliquées, Oran, ALGERIA*
[2]*Laboratoire d'Informatique Oran (LIO), Département du Vivant et de l'Environnement, Faculté des Sciences de la Nature et de la Vie, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, USTO-MB, Oran, ALGERIA*

Received 05 Jun. 2022, Revised 06 May. 2023, Accepted 09 May. 2023, Published 01 Aug. 2023

Abstract: The main objective of this study is to create a collaborative parallel environment, which supports the unsupervised classification and the filtration of an important volume of information. The proposed approach consists in the integration of an agent-based system composed of five reactive agents to assist the recommendation of the stored services in the cloud and to cluster these services through a new improved K-means as well. The conducted experiments and evaluations of the different approaches and measures, such as: Euclidean distance, Manhattan distance and Cosine similarity, show that the proposed approach of the unsupervised classification improve the within cluster sum of squares (WCSS), which facilitates the access to personalized and relevant services requested in a very improved response time, especially through migration to the cloud using agents.

Keywords: Cloud computing, Improved K-means, K-means, Multi-agent systems, Recommendation, Unsupervised classification

1. INTRODUCTION

The communication between companies that work in the same sector is primordial for sharing expertise and resources. As a result of the necessities of an industrial company, the request of an expert, a machine or a commercial premise can be initiated. The acquisition of the requested and personalized service must be achieved in an advanced and fast form, which implies that the search for this service must be performed with companies providing similar services. Therefore, it consists of a collaborative environment that supports the companies providing or requesting a service.

This work environment should ensure parallelism, flexibility and storage capacity for a clear and concise recommendation. It requires a high performance system to provide consistency, in order to recommend relevant and focused information in a short amount of time, which leads to the combination of multiple techniques and tools for the association of their strengths, in order to take advantage of their complementary effect.

The identified problem consists in guaranteeing a productive collaboration and communication between the involved companies based on the promotion of a personalized and fast filtration considering a large volume of services. To answer these issues, the main lines of the contribution process consist of:

- The use of natural language processing (NLP): the standardization of the language used by the collaborating companies for a meaningful comprehension. Since the service is introduced through a controlled natural language, the language normalization has been applied.
- Designing an agent-based architecture to promote parallelism and system performance.
- The proposition of a new clustering approach for generating an effective unsupervised classification of services.
- The migration to a cloud platform to guarantee the storage capacity and a large-scale passage with the integration of cloud services.
- The use of a recommendation tool to simplify the search process in a short time with guaranteeing successful recommendations.

bouranane.djihane@edu.univ-oran1.dz, nawal.sadhouari@univ-usto.dz, taghezout.nora@gmail.com

- The evaluation of the developed concepts' impact.

In order to conceive this process, an agent-based architecture has been developed to achieve the system balance and lightening, during the unsupervised classification (clustering) and recommendation, with processing a large amount of data in a cloud environment. For the sake of increasing the performance of the proposed system through introducing storage capacity, managing and allocating services in a flexible and reliable state and avoiding the problems of overload, these different paradigms have been combined.

This research paper organization comes as follows: Section II is dedicated to the state of the art on the different concepts and techniques, as well as the works, which fit in our thematic. Section III gives details about the architecture of the proposed approach. As for section IV, it reports the conducted experiments with the obtained results and their interpretations, whereas section V gives a general conclusion.

2. STATE OF THE ART

In order to have a powerful intelligent system to assist the involved process, the research has been conducted for different entities covering a wide variety of domains with the purpose of combining the strengths of each. This section provides a systematic overview of the integrated concepts, and it discusses some related researches to assimilate the work that will be undertaken in this manuscript.

A. Overview

1) Clustering

Clustering is an unsupervised learning approach for classifying items automatically, in order to build a prediction model. This permits the optimal partitioning of the initial data set [1]. It is a technique allowing the determination of the relationship between the items in a data set [2]. Clustering techniques can improve the performance of the recommendations and response time by dimensional reduction [3], [4]. The idea behind this is to form clusters with a reduced number of data allowing the manipulation of the targets in a short time with fewer calculations.
Centroid-based algorithms are based on the elaboration of a number of clusters, each with a representative called centroid, which is calculated from the average of the items belonging to its cluster. K-means is the most commonly used method in the context of recommendation systems [3], providing a significant improvement [5]. Based on the carried-out study [6] analyzing a number of literature articles looking for the most partitioning clustering algorithm used with recommendation systems.
This clustering technique, associated with an appropriate similarity measure, presents efficient results while having a principle easily assimilated and being simple to implement. In addition to that, it manipulates all types of data, and increases productivity using a large corpus [7], and has a complexity equals to O (n) for n items [8]. On the other hand, the major limitation of clustering algorithms is usually the initial partitioning [7]. More precisely, the problem with this K-means algorithm is the random choice of the number k of clusters and centroids [9], [8]. In order to overcome the issue of determining the number of clusters, different approaches have been proposed in [10] indicating that they are not really meaningful.

2) Recommendation systems

The proliferation of the indexed data on the web has caused a series of problems related to information overload, which means an excess of choice against users, who are prevented from distinguishing and selecting relevant information in a rapid reliable manner. In the literature, several solutions have been proposed including the systems filtering information and providing personalized recommendations. These systems reduce search efforts and response time to make a decision about an alternative. Recommendation systems have been defined as an alliance that protects users from information overload [11].
According to [12], the recommendation-based algorithms learn about users with the aim to provide them with relevant information meeting their expectations. This information is given in the form of a list of recommendations that are mostly generated, according to three types of filtering: collaborative filtering, content-based (cognitive) filtering and hybrid filtering that combines different types of recommendations to aggregate their strengths. This classification has been adopted, according to the type of the followed filtering mechanism [13].
For filtering, these algorithms are based on similarity measures. Therefore, depending on the mechanism of the adopted algorithm, similarity measures are used in different ways.
These similarity measures are chosen in accordance with the type of data and their representation. They play a crucial role in determining the performance of prediction results in the context of recommendation or unsupervised classification in the context of clustering. According to [14], the choice of the similarity measure has a direct influence on the quality of the recommendations in terms of accuracy. The cosine similarity measure is the most widely used in clustering [15], [16], [3], [17] and information retrieval techniques [18], [19], [20]. Several comparisons have been established approving the quality of its performance [16], [21], [17], [22], [23].
The evaluation of recommender systems has always been the focus of several researchers, resulting in different performance evaluation measures in addition to the criterion of the user's satisfaction.
To summarize, recommendation systems are a means of selecting relevant information from a large amount of data that provide meaningful decision support. This has led to their widespread use, integrated in most cases with other methods and technologies to overcome certain anomalies.

3) Multiagent systems

Multi-agent systems are an optimal solution that not only reduces response time but improves system performance by using the notion of parallelism and task distribution. Multi-agent systems (MAS) are defined in [24] as systems that promote fault tolerance and scalability through the multitude of agents working in the same distributed environment.

The combination of these definitions presents a multi-agent system (MAS) as a flexible system (adding new agents [24]), reliable (distributed problem resolution), fault tolerant (task assigned to another agent), effective (distribution of tasks), and inexpensive (allocation of overheads avoiding the need for a powerful entity [25]). This means that, these systems are used in a wide variety of domains to solve real problems: Smart cities [26], Urbanism [27], Smart Grid [28], [29], Robotics [30], Internet of things [31], health care [32], cloud [33], [34] etc.

These systems are used for a better learning and a good, intelligent, dynamic and especially parallel reflection to alleviate the systems and improve their performance.

4) Cloud computing

Dynamic systems that interact with a large number of users and data require better performance in terms of storage capacity, calculations and fast processing. This comes in the context of good elasticity and great availability. In recent years, cloud computing has proven to be the best paradigm of virtualization that provides well automatic management and an allocation of abstract resources on demand.

Cloud computing is a set of services and hardware that deliver these services [25], [35], [36]. It is a large-scale distributed technology that enables the supply of resources in a dynamic and parallel manner. Virtualization in this paradigm allows for failure tolerance by isolating applications from each other, thus avoiding the failure of the entire system [37].

Cloud Computing has experienced an exceptional evolution from its advantageous impact, which can be represented in the support of intensive calculations [37], [38] and a large number of users [35], the provision of significant storage capacity [37], elastic and advanced services [35], reduced user costs [25], [37], [39], scalability and reliability that reduce response time and increase resource availability [40], [25] and in overall performance improvement [35].

However, with all these advantages, this technology still has challenges to overcome, related to data security, consistency of replications, etc. This prompted researchers to combine cloud computing with other technologies such as: multi-agent systems (for conflict resolution [41]) and the recommendation [38] for large data filtering.

In this same context, multi-agent systems (MAS) refer to another decentralized paradigm, which involves a multitude of agents to solve a problem intelligently [37].

As a result, these two distributed models have often been combined from the point of view that they are complementary. Cloud computing provides an environment with high performance and a large storage capacity for the scalable execution of an MAS. As for the MAS, it provides the cloud platform with intelligence, autonomy and reasoning to improve its flexibility and interaction besides a level of confidentiality [42], [37], [36]. Combined with the cloud paradigm, compared to other environments, a MAS system is more powerful taking advantage of its elasticity for scalable execution [37].

B. Literary review

In the literature, several investigations have focused on improving recommendations to foster collaboration between different entities. This research has been based on the integration of different techniques.

Preparation and pre-processing of the dataset prior to filtering and recommendation has also been proven to be a technique improving the relevance of recommendations. Two studies [40], [43] have focused on the proposal of an algorithm using data mining techniques for the identification of frequent sequential access patterns on the web. The principle is based on the generation of a graph from a preprocessed exploration, which helps to extract the implicit user's behavior based on his navigation. This identification leads to the recommendation of relevant web page links. The proposed contributions differ in the steps of the algorithm. In [40], it is about the removal of infrequent web pages before the graph creation, followed by the removal of infrequent edges, then, the generation of frequent sequential patterns, according to their frequency. Finally, the web page recommendation rules are generated. However, in [43], the creation of the graph precedes the elimination of the nodes and edges. These two methods are an improvement of an older algorithm allowing the generation of the graph without considering the recommendation. Comparing these two approaches, the first one reduces the graph generation time and requires less memory space.

Focusing on problems that are related to the reduction of recommendations' relevance and response time, some works have introduced clustering method. A recommendation system based on collaborative filtering and clustering has been proposed in [4] to provide better guidelines and decision support in the context of cardiovascular disease. Concerning the random choice of centroids, it suggests choosing only one centroid randomly as long as the others are determined, according to the standard deviation. The items belonging to the same cluster are those that have the smallest standard deviation, according to the means. This approach avoids the problems of sparsity and scalability, and improves relevance and response time.

Another approach discussed in [7], which is based on collaborative filtering for the efficient recommendation of TED talks using the K-means clustering method for the construction of the predictive user model. The obtained results show that this approach avails effective recommendations and predictions.

Another aspect has been highlighted in [11], which is a work that ensures the recommendations performance for big data through proposing two collaborative filtering approaches. The first one is based on an improved K-means clustering algorithm, and the second uses the same algorithm with a covariance-based dimension reduction method (principal component analysis). In terms of error prediction computed by MAE and RMSE measures, these proposed approaches have the lowest value compared to the classical collaborative filtering algorithm. With the specification that the second proposed approach, including dimension reduction, decreases the margin of error significantly. In addition to the work abrogated in [45], which consists in using two hybrids clustering algorithms based on sequences and hierarchy, for recommendation of e-commerce web pages. This recommendation method based on clustering has shown significant results. However, the aspect of computing power (deployment in the cloud) has not been addressed.

As for the research conducted in [44], the objective is to make it easier for students to access suitable courses according to their interests, and to facilitate a collaborative and efficient working environment. On the one hand, the adopted approach enables unsupervised classification to divide students into clusters based on term frequency and semantic feature extraction algorithm using an improved K-means algorithm, which aims to improve the selection of initial cluster centers. On the other hand, it allows the recommendation of a limited number of well-targeted suitable courses to a trained student population. The recommendation process uses semantics, which leads to solve the cold start problem that may be encountered. By reverting to previous clustering methods, the obtained recommendation results are stable and improved.

Still in the field of education, a system for recommending books along with their descriptions and metadata based on an RDF knowledge graph is designed in [45]. The initially followed approach focuses on forming book clusters via the K-means algorithm, and the next step involves predicting and scoring books before making collaborative filter-based recommendations. The discussed results report the relevance of the recommendations and predictions.

In order to provide the points of interest recommendations in the field of urban tourism for tourist groups with similar preferences and opinions, the work presented in [46] enables the development of recommender systems based on clustering and fuzzy best and worst methods. The used clustering algorithm is a modified K-means algorithm implemented by the evolved Euclidean distance and elbow-based method to select the number of clusters. The experimental results show the recommendations' relevance, which is further enhanced by using fuzzy methods.

According to these works, clustering brings more relevance to the recommendations.

Among the conducted researches, several have migrated to the cloud computing paradigm. As in [47], a hybrid approach has been implemented for recommending banking products in addition to recommending solutions for the banking entity based on intelligent agents and case-based reasoning in the cloud to facilitate sharing.

On the Basis of the recommendation systems deployed in the cloud, the integration of agent-based architectures has been promising. The closest work to our contribution [39] proposes a multi-agent system to dynamically process and analyze the operation of a user's application running in a public cloud environment, in order to provide adequate resources. This system uses three agents, which apply the recorded resource predictions (deductive reasoning) with inference rules to dynamically choose the best parameters to use for the execution of the application in the public cloud. The evaluation of the proposed system shows good prediction results and a good balance between CPU usage, application execution time and cost.

C. Contribution

The main objective of our contribution is the elaboration of a high-performance system with a great flexibility, which guarantees the acquisition of a personalized service in a short time with a significant relevance. The idea behind this study is to combine different methods and paradigms by matching their strengths to achieve a cost-efficient combination and the proposal of an improved clustering technique for the generation of targeted clusters in order to increase the relevance of recommendations.

Our contribution addresses two important phases that lead to the delivery of a service, which can be an expert, a machine or a premise. The first phase consists of data preprocessing, information extraction and clustering. This step has been developed as follows:

- The extraction of information from the collected data considering semantics based on conceived domain ontology.
- The representation of data (the chosen model is the vector representation).
- The implementation of the K-means clustering algorithm and comparison with a new improved K-means algorithm that we have proposed for improving the assignment of data in suitable clusters using Euclidean distance, Manhattan distance and cosine similarity for the comparison.

The second phase allows processing and personalizing the request for a recommendation process. It has been conceived by:

- the implementation of the recommendation algorithm based on content filtering and cosine similarity measure.
- the conception of a user feedback for the evaluation of the recommendations.

The execution environment of this approach is implemented in an agent-based architecture composed of 5 agents deployed in a cloud public infrastructure (Iaas). In order to place our contribution among the related works, a comparative table is given in Table I.

TABLE I. A COMPARATIVE TABLE BETWEEN THE RELATED WORKS AND THE PROPOSED APPROACH

Works	P	Rep	Clust	Rec	Sim	Items	MAS	Cloud	Eval
[11]	-	(V)	Improved K-means + PCA	(CF)	Pearson correlation	Movies	-	-	MAE, RMSE
[4]	+	(V)	K-means, KNN	(CF)	Pearson, Cosine, Euclidean, Weighted proposed similarity	Treatment for cardiovascular disease	-	-	Response time, MAE, recall, precision
[7]	-	(V)	K-means, KNN	(CF)	Pearson correlation	Ted Talks	-	-	RMSE, recall, precision, F1
[39]	-	-	Linear regression	-	-	-	+	+	MAE, RMSE Bias, MAPE
[38]	+	(V)	-	(CF), (S)	Not announced	e-commerce	-	+	-
[48]	-	-	(Ph)	Hybrid	(Dps)	e-commerce websites	-	-	Response time, precision, recall, F-measure
[44]	+	(V)	K-means, Improved K-means	(CF)	Cosine	Courses for grouped students	-	-	precision, recall, RSA, Popularity average
[45]	-	-	K-means	(CF)	-	Books	-	-	MAE, recall precision, F1
[46]	-	(V)	K-means, Improved K-means	(CF)	Developed euclidean distance	Points of interest	-	-	User satisfaction, precision, recall, F1,
Our work	+ (O)	(V)	K-means, Improved K-means	(Cb)	Cosine, Euclidean, Manhattan	enterprise services	+	+	User satisfaction, MAE, NRMSE, RMSE, recall, precision, F-measure, Response time

Where: P. Preprocessing (semantics), Rep: Representation model, Clust: Clustering (machine learning), Rec: Recommendation type, Sim: Similarity measures, MAS: Multi-agent system, Eval: Evaluation method, (V): Vector model, (CF): Collaborative filtering, (Cb): Content based filtering, (S): sentiment analysis, (O): Ontology, (Ph): Proposed hybrid clustering (HSC: K-medoid + DBSCAN, TSC: B-trees + BIRCH), (Dps): Dynamic programming based sequence alignment method

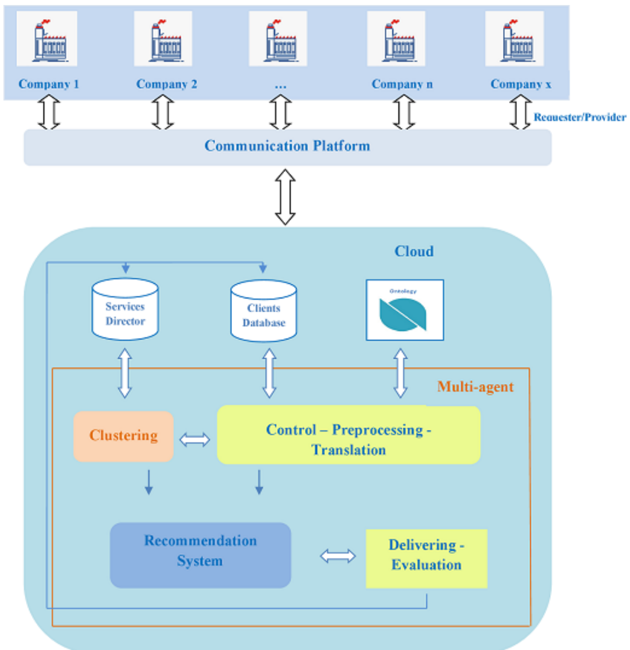


Figure 1. Architecture of the proposed system

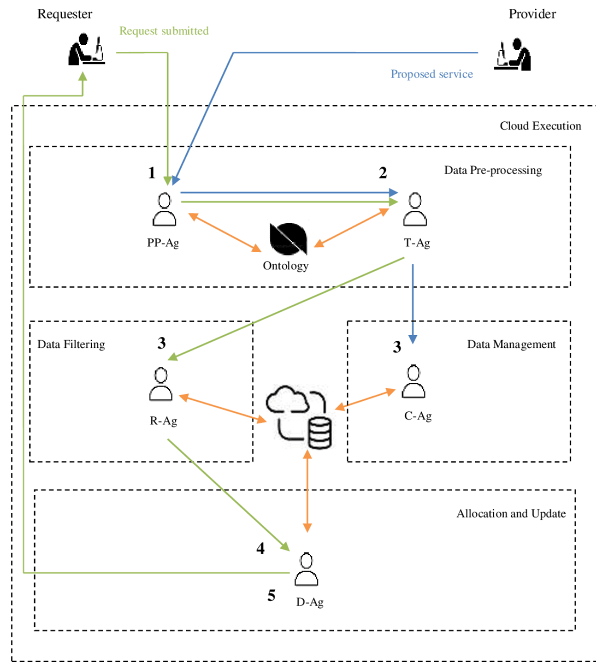


Figure 2. Structure of the agent-based architecture

3. PROPOSED APPROACH

The proposed approach consists of establishing a collaborative system dedicated to companies to assist them in requesting or providing a service. The proposed architecture is illustrated in Figure 1.

Within this architecture, several modules of different concepts are related to the elaboration and the acquisition of a model that meets the identified objectives.
Our architecture is dedicated to two types of profiles:

- **A provider** for adding a new service: this part involves adding a service so that the service directory is updated. First, the natural language of the query is translated. Then, its vector representation is elaborated. The following process leads to a similarity calculation with the centroids of the clusters for the assignment of this proposed service.

- **A requester** for requesting a service: this space selects a list of suitable services considering the centroids of the clusters to select the most similar one so that the list is searched among the members of the selected cluster.

For this purpose, we have used a multi-agent system, which is composed of 5 agents. The structure of the proposed system is shown in Figure 2.

As shown in Figure 2, several modules are included. The following section provides the details of these modules.

A. The cloud computing

Cloud computing has been used as an execution infrastructure platform and storage support to enable large-scale passage and increase system performance. The idea behind the deployment of our system into the cloud is to enable: the handling of a large number of services, guaranteeing the system accessibility and the increase of its availability and flexibility by taking advantage of the cloud's flexibility feature.
The deployment of our system has been performed on a public cloud to take advantage of its unlimited number of resources allocated on demand and its open access. With Iaas, as a type of service, to benefit from an infrastructure that allows the deployment of our system with a certain degree of control.

B. Agent-based modeling

The use of multiple agents contributes to reduce the system load through the notion of parallelism. As different steps are predicted, it is necessary to distribute the tasks and assign them to different agents to reduce the response time and workload.
The proposed multi-agent architecture consists of five reactive agents: Preprocessing, Translation, Clustering, Recommendation and Delivering Agents.

- **Preprocessing Agent (PP-Ag):** This agent is dedicated to the preprocessing of the request by presenting the requested or the proposed service. This pre-processing consists in eliminating empty words and extracting information in the form of keywords,

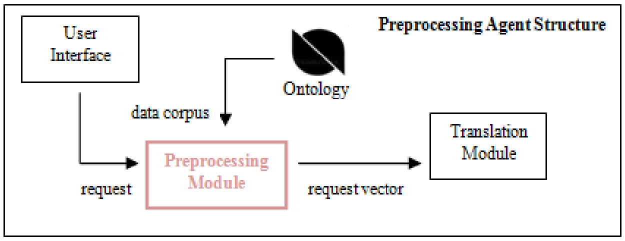


Figure 3. Structure of the preprocessing agent

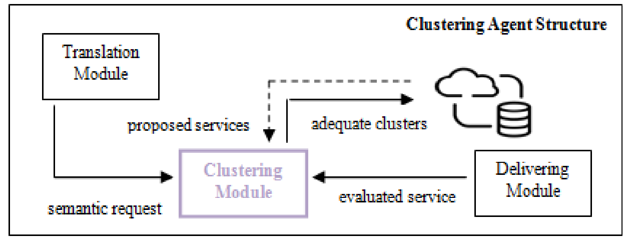


Figure 5. Structure of the clustering agent

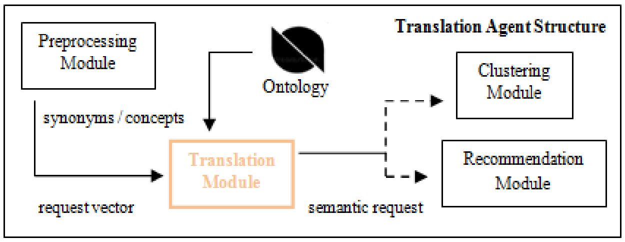


Figure 4. Structure of the translation agent

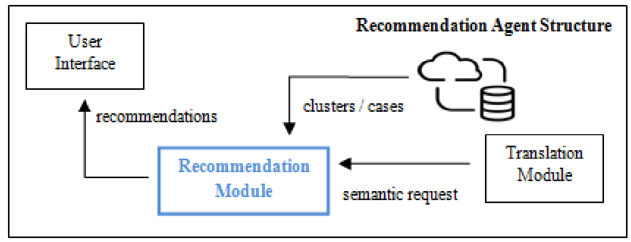


Figure 6. Structure of the recommendation agent

which are then indexed as a vector (Figure 3). Since the considered data is semi-structured, this extraction step is necessary to convert the semi-structured data into a structured data. Moreover, as long as the items used in this study can be described by a common and a known set of attributes (representative keywords), and as long as the implementation of the vectors is simple, and meets our work requirements, the vector representation has been adapted.

- **Translation Agent (T-Ag):** The expansion of vectors is very important; it brings meaning and semantics to the required information. In our case, we have developed domain ontology in order to resolve the problem of polysemy and synonymy, by creating a concept that reflects the meaning of all these synonyms. Based on this ontology, which plays the role of a dictionary (Stemming Step), this agent allows the standardization of the vectors of the provider's or requester's queries (Figure 4).

- **Clustering Agent (C-Ag):** With the objective of facilitating the extraction of relevant and targeted information in a shorter time, clustering techniques are used to form categories. This step is accomplished by Clustering Agent (Figure 5), which runs a clustering algorithm to produce clusters for an unsupervised classification of services using similarity criteria. This step allows assigning the provider's request to the appropriate cluster.
On the basis of a comparison with the classical K-means, the clustering module follows a new improved unsupervised classification approach of the K-means algorithm applied with the cosine similarity measure. This similarity measure has been chosen among two other measures (Euclidean distance and Manhattan distance).
The proposed approach of the improved K-means (Algorithm 1) relies on a new idea, which allows to select the most distinct centroids and then to merge them with similar items for the generation of suitable clusters.
For solving the K-means problem related to the choice of the number of clusters, we have previously defined the latter by the elbow method also used in [49], [50].
In order to reflect the users' evaluations, the clustering is recalculated following an evaluation initiated by the user after the recommendation of an inadequate service.

- **Recommendation Agent (R-Ag):** In the case of a service request, this agent performs filtering to generate relevant service recommendations (Figure 6). First, the similarity rates are calculated with the centroids of the clusters using the Cosine similarity measure. Then, the most similar ones are selected to perform further similarity calculations with its members to generate recommendations. These calculations are simultaneously conducted (in parallel), which reduces response time, and lightens the system in a multi-agent architecture.
Recommendations are determined based on content-based filtering to alleviate the problem of a newly requested service or a similar service that has recently been provided.
The choice of the Cosine similarity measure has been made on the basis of a comparative study in a previous work [51].

- **Delivering Agent (D-Ag):** The coherence of our knowledge bases requires updates on the availability

Algorithm 1 Improved-K-means;

Input: Services data set serD, k
Output: Cluters

i = 1; $sim1$ = []; $sim2$ = []; Centroids set Ctr [k], threshold t
if *serD* ≠ ∅ **then**
  id ← *random(serD);*
  Ctr[0] ←*id*;
  serD←*serD* -{*Ctr[0]*} ; *//remove Ctr[0] from serD*
end
for each $x \in serD$ ***do***
  sim1[x]= cos(*Ctr[0]*,*x*);
  end for

*sim*1[] ← *ascending – sort(sim1);*
while *(i <k)* ***do***
  Ctr[*i*] ←*sim1[i]* ;
  serD ←*serD* -{*Ctr[i]*} ;
  //select first k-i items from sim1 and remove them from ser-D
  end while

i = 1 ;
  while *(i <k+1)* ***do***
  for each $x \in Ctr - \{i\}$ ***do***
  sim2[x] = cos(*Ctr*[*i*], *x*);

if *sim2[x]* ≥ *t* **then**
  place *x* ***in*** *cluster i and* ***remove*** *x* ***from*** *Ctr;*
  select *first item* ***from*** *sim1 and* ***remove*** *it* ***from*** *serD and* ***place*** *it* ***in*** *Ctr;*
end
end for
  i =i+1 ;
  end while

i=0 ;
  for each $x \in serD$ ***do***
  while(*i* ¡ *k*)***do***
  sim[i]= cos(Ctr[i],x) ;
  i + + ;
  end while
  select *most similar i and* ***place*** *x* ***in*** *cluster i;*
  end for

recalculate *centroids* ***for each*** *cluster and* ***place*** *them* ***in*** *Ctr;*

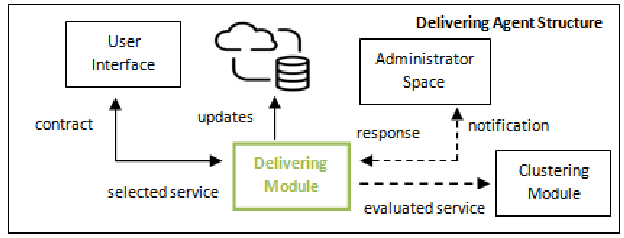


Figure 7. Structure of the delivering agent

of services and traceability. This agent is responsible for these updates and for recording assessments (Figure 7).
In order to present a more detailed and concise explanation of the process behind the proposed approach, we have presented the main points of the two phases (clustering and recommendation) through a pseudo-code (Algorithm 2 and Algorithm 3).

Algorithm 2 Service-Clustering;

Input: Request req, Ontoloy o, Centroids Ctr
Output: adequate cluter

sim = [];
if *authorized-user* **then**
  Apply preprocessing module **by** PP-Ag;
  req ← *preprocessing*(req);
  Apply *translationmodule***by** $T - Ag$;
  req← *translation*(req);

  for each $x \in Ctr$ **do**
  sim[x]= *cos*(x,req);
  end for

  select the most similar centroid and **place** *req* **in** cluster *x* ;
  recalculate the centroid **for** cluster *x* and **modifie** it **in** *Ctr* ;
end

The Users' requests are generated from a semi-natural language through sending a small paragraph expressing their needs. After the preprocessing phase and the generation of a semantic vector, a verification of the presence of domain concepts is performed using the ontology to accept or reject the request. The transformation of the request into a vector is also evaluated by the user himself.

An overall view of the system process has been modeled through an activity diagram in figure (Figure 8).

4. IMPLEMENTATION AND RESULTS

A. Development tools

The system implementation has been realized using the programming language Java associated with the agent

Algorithm 3 Service-Recommendation;

Input: Request req, Ontoloy o, Centroids Ctr
Output: recommendation list of services

```
sim = [];
if authorized-user then
    Apply preprocessing module by PP-Ag;
     req ← preprocessing(req);
     Apply translationmoduleby T − Ag;
     req← translation(req);

    if accepted-req then
        for each x ∈ Ctrdo
         sim[x]= cos(x,req);
         end for

        select most similar centroid x from sim ;

        for each s ∈x do
         sim[s]= cos(s,req);
         end for

        select top-k most similar services from sim and
         recommend them;
    end
    if user-satisfaction then
        Assign service with highest score to user;

        else

        Notify experts for reallocation in appropriate clus-
         ters using Service-Clustering() or request pre-
         processing baed on user' scores;
         Apply Service-Recommendation() from 3 or 7;
    end
end
```

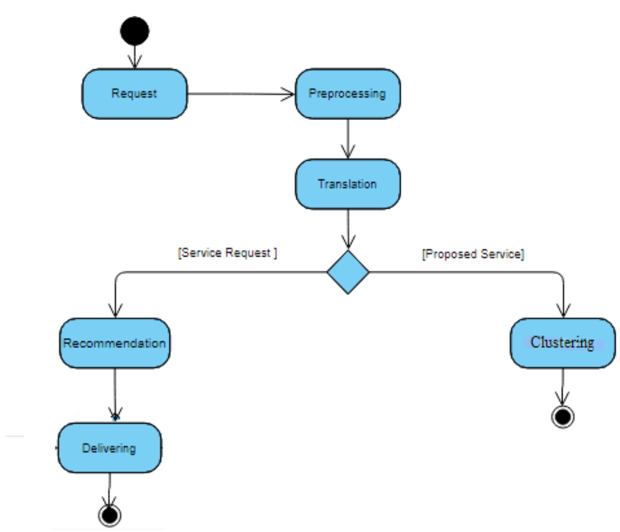


Figure 8. Structure of the delivering agent

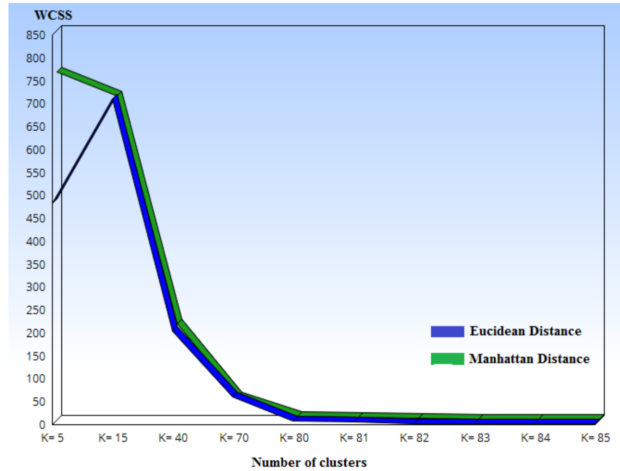


Figure 9. Comparison curves between Euclidean and Manhattan distance

interface JADE, the database creation language PostgreSQL and the ontology creation software Protégé. This implementation has been then migrated to a cloud infrastructure with a physical storage capacity (RAM) 4 times larger (16 GiB) and 8 processors.

B. Experiments: evaluation and calculation of performances (clustering, recommendation, agent and cloud integration)

1) The first experiment: a comparison between clustering methods and similarity measures

First, a comparative study has been conducted to select the appropriate distance measurement with the classical K-means. Two measures have been involved: the Euclidean distance and the Manhattan distance. The evaluation has been applied on a dataset, composed of 1683 services that belong to the industrial domain, and has been performed considering the response time and the within cluster sum of squares (WCSS).

As it's observed in the figure (Figure 9) and the table (Table II), Euclidean distance has the lowest WCSS in most cases, even though the Manhattan distance may be more or less quickly. The choice of the Euclidean distance is the most appropriate, because our priority is to find the most optimal distribution for a highly relevant recommendation, and we plan to achieve time savings by using other tools. Following the Elbow method focusing on the K-means, which decreases the WCSS gradually and linearly, the best number of clusters K is: 70 clusters.
As a second step, clustering with the classical K-means using Euclidean distance has been compared to the proposed algorithm (the previously presented improved K-means) applied with Euclidean distance and Cosine similarity. The results are shown in Table III.

According to the comparison of cluster generation time in relation to the number of clusters in the tables (Table II

TABLE II. COMPARISON BETWEEN EUCLIDEAN AND MANHATTAN DISTANCE

Nb Cluster	Euclidean distance		Manhattan distance	
	WCSS	Runtime (ms)	WCSS	Runtime (ms)
K=5	481.53136052368336	14370	757.1361754542049	14014
K=15	709.2992908388202	15454	705.6482908388201	16249
K=40	203.04899330047363	18641	209.46050742609734	12385
K=70	58.91534146341461	17252	48.46424999999997	14752
K=80	7.302499999999997	15430	7.398333333333333	17338
K=81	4.6000000000000005	15015	4.829999999999998	14684
K=82	1.725000000000005	18262	2.875000000000006	30258
K=83	1.3800000000000041	18427	1.3900000000000041	28779
K=84	1.3800000000000041	16766	1.3800000000000041	17746

TABLE III. COMPARISON BETWEEN THE IMPROVED K-MEANS AND CLASSICAL K-MEANS USING EUCLIDEAN AND COSINE MEASURES

Nb Cluster	Improved K-means Euclidean distance		Imropved K-means Manhattan distance	
	WCSS	Runtime (ms)	WCSS	Runtime (ms)
K=5	8.0402737	16575	1.7914986	18808
K=15	8.454343	34484	1.7905511	22642
K=40	8.5375834	42674	1.7890052	45705
K=70	3.2016816	38753	0.3882726	29558
K=80	3.028817	68158	0.3856079	48138
K=81	2.798172	54188	0.34837747	57412
K=82	2.7691953	61729	0.34982944	39407
K=83	1.2903903	63214	0.3288336	41037
K=84	0.85257584	68558	0.3040395	39849

and Table III), a curve has been drawn (Figure 10) to show that the time taken by the K-means algorithm using the Euclidean distance is smaller than that of the improved K-means algorithm using cosine similarity and Euclidean distance respectively.

Based on the comparison between different WCSSs provided according to the number of clusters for each method, the obtained results are presented in the figure below (Figure 11).This has led to the conclusion that in our case study, the WCSS rate of the classical K-means method exceeds that of the improved Method. Furthermore, using the cosine similarity measure with the proposed method further improves WCSS and increases its stability in a very important way.

The proposed method confirms the number of clusters K. Even if it takes more time, but has a very significant error rate.
In relation to our case, clustering is done by the administrator out of service request time. So, the more important is that the WCSS, which presents to the user a better quality of clusters, so that it will serve to decrease the response time to his request.
According to this information and that presented in figures (Figure 10 and Figure 11), the most appropriate method is

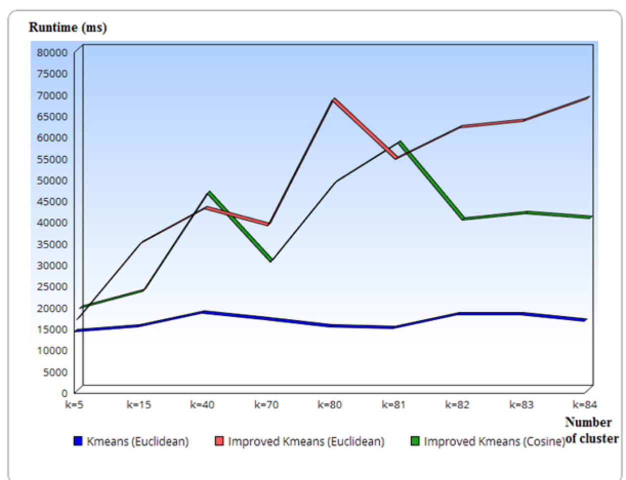


Figure 10. Comparison curves based on runtime for classical and improved k-means applied with different similarity measures

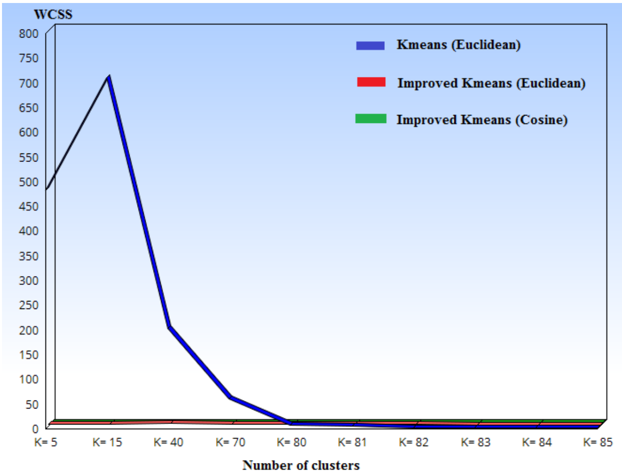


Figure 11. Comparison curves based on WCSS for classical and improved k-means applied with different similarity measures

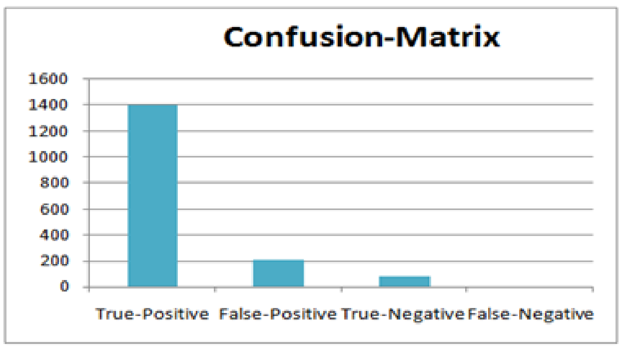


Figure 12. Recommendation filtering confusion matrix

the proposed K-means method, which is used with cosine similarity.

2) The second experiment: a comparison between the classical recommendation based on content filtering and the recommendation based on clustering

This experimentation has been approached to choose the best method applicable on our dataset in terms of quality of recommendations and response time, besides the evaluation of the applied clustering method (Table IV). The study has been conducted on a dataset of 1683 records, with the basic query (Query-b): an expert with a master's degree in the field of automation and an experience of 8 years with a remuneration not exceeding 30000DA.

The most similar case must exceed a similarity rate equal to 0.80. The choice of this threshold is explained by the false positives of the confusion matrix (Figure 12) having a similarity between [0.5-0.8[. The evaluation metrics have been implemented, and the results have automatically been generated, after launching the query used for the experiments. According to these results, the confusion matrix has been generated.

The services classified as false-positive and true-negative (contradiction between prediction and observation), have been retrieved for future contributions in experiments based on user satisfaction.

The recommendation of the same list of services in (1) and (2) confirms the quality of the partitioning (clustering) with a slightly reduced response time, because the most similar cluster contains an important number of services. This leads to the choice of a recommendation conducted on clusters. The evaluation of the recommendations' quality has been performed by considering different measures presented in table (Table V). The latter provides a comparison of our method with those of some related works.

Where: MAE: Mean Absolute Error. RMSE: Root Mean Squared Error. NRMSE: Normalized Root Mean squared Error.

The comparative study reported above considers the top 10 item lists recommended for datasets of different sizes.
Compared to the other works, the interpretation of this table (Table V) shows that the results of our approach are very significant. It presents an approximate full coverage (recall 98.8%), significant accuracy, a practically negligible margin of error and a reduced time.
These results indicate that the recommendation process applying the improved K-means clustering, in this paper, is an accurate approach, which proves the efficiency of the recommendations, from one side, and clustering, from the other one, since the latter directly affects the relevance of the recommendations.

3) The third experiment: agent integration

The importance of integrating agents in reducing time is shown in (Table VI), which presents less run time of the executed methods in an agent-based architecture compared to the results above. The importance of agents also resides in the parallelism, which enables the execution of several agents (methods) at the same time so that the information overload is supported.

According to the above table, we notice that the time required by both agents (C-Ag) and (R-Ag) is reduced compared to the execution of these same algorithms independently of the agent-based architecture.

4) The fourth experiment: cloud integration

The last experiment consists of comparing between the global system in and out of the cloud, in order to gain flexibility, storage capacity and execution speed. This comparison confirms our expectations (Results in Table VII).

The results of the experiments performed in/out of a cloud environment, presented in the table below, show an encouraging improvement in terms of execution time. In addition to that, an improvement in the use of CPU (reduced to 9%) and memory usage (estimated at 21%).

TABLE IV. COMPARISON BETWEEN THE DIFFERENT APPLIED RECOMMENDATION METHODS

	Basic Recommendation (1)	Recommendation with clustering (2)
Used Query	Query-b	Query-b
Top 10 recommended services (Identifier ID)	190, 806, 29, 483, 893, 1221, 1676, 1282, 453, 13	190, 806, 29, 483, 893, 1221, 1676, 1282, 453, 13
Runtime (ms)	4114	2569

TABLE V. COMPARATIVE TABLE RELATED TO THE QUALITY OF RECOMMENDATIONS

Works	Precision	Recall	F1	MAE	RMSE	NRMSE	User satisfaction	Response time (ms)
[11]	-	-	-	0.89	0.89	-	-	-
[4]	61%	58%	-	0.278	-	-	-	20000
[7]	94.5%	91%	93%	-	0.142	-	-	-
[44]	50%	48%	-	-	-	-	-	-
[46]	92.5%	81.8%	86.8%	-	-	-	+	-
Our work	90.1%	98.8%	94.2%	0.001	0.011	0.022	+	2569

TABLE VI. REDUCED RESPONSE TIME BY INTEGRATING AGENTS

Process	Runtime (ms)		Specifications
	Without agents	With agents	
Partitioning	29839	21106	with improved k-means + cosine similarity (**C-Ag**)
Recommendation	2569	1942	with centroids (**R-Ag**)

TABLE VII. COMPARISON TABLE BETWEEN THE AGENT-BASED ARCHITECTURE DEPLOYED IN AND OUT OF CLOUD

	Runtime (ms)	
	Without Cloud	With Cloud
Clustering (Improved k-means + cosine)	21106	4961
Recommendation (clustering)	1942	1097

The gained response time and CPU resources has been recorded for a data set of 1683. We are considering further expansion of the dataset, to compare the improvements in the cloud, in order to find the best ratio between the performances of our system with the amount of explored data.

5. CONCLUSION AND FUTURE WORK

Multi-entity collaboration requires a consistent work environment that allows accomplishing accurately the desired tasks in a specific time and a consistent information sharing. The accomplished work involves enterprises and their potential exchanged services.

In this paper, the design of a coherent and flexible environment has been investigated in cloud computing as an agent-based architecture, which includes several axes and aspects such as: clustering and recommendation. Initially, the standardization of the discussed language has been applied using a domain ontology, which is followed by an unsupervised classification of the directory into clusters using a new approach. Then, the other axis, focused on filtering and targeted recommendation of a relevant service, has been initiated.

After various experiments, the standardization of the language has greatly facilitated the introduction of semantics and consistency, which was achieved by surrounding the language used among the different involved entities. The proposed clustering has also contributed to the relevance of the recommendations and the improvement in terms of response time. In addition to that, the integration of agents and cloud services has improved the system performance.

In the future, an improvement to the decision making and negotiation method is envisaged by comparing and suggesting new techniques and integrating the fuzzy method of cognitive agents, as well as an expansion of the number of the involved services and companies.

ACKNOWLEDGMENT

Sincere thanks are expressed to all involved members. Especially, authors would like to thank the Directorate General for Scientific Research and Technological Development (DGRSDT), an institution of the Algerian Ministry of Higher Education and Scientific Research, for their support on this work.

References

[1] A. Hervé, G. Vincent, E. Aida, and B. Derek, "Canonical correlation analysis," *Encyclopedia of Social Network Analysis and Mining*, 2018.

[2] W. ZHE, "Research on personalized recommendation algorithm based on collaborative filtering and partition clustering," *2nd International Conference on Test, Measurement and computational method (TMCM), DEStech Transactions on Engineering and Technology Research*, 2017.

[3] Z. Hafed, A.-S. Ziad, A.-A. Mahmoud, and J. Yaser, "A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques," *ICICS 9th International Conference on Information and Communication Systems*, pp. 102–106, 2018.

[4] A. MUSTAQEEM, S. M. ANWAR, and M. MAJID, "A modular cluster based collaborative recommender system for cardiac patients," *Artificial intelligence in medicine*, vol. 102, p. 101761, 2020.

[5] H. KOOHI and K. KIANI, "A new method to find neighbor users that improves the performance of collaborative filtering," *Expert Systems with Applications*, vol. 83, pp. 30–39, 2017.

[6] L. MIRANDA, J. VITERBO, and F. BERNARDINI, "Towards the use of clustering algorithms in recommender systems," *AMCIS Proceedings: SIGODIS*, 2020.

[7] F. MAAZOUZI, H. ZARZOUR, and Y. JARARWEH, "An effective recommender system based on clustering technique for ted talks," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 15, pp. 35–51, 2020.

[8] A. K. JAIN, N. MURTY M., and P. J. FLYNN, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, pp. 264–323, 1999.

[9] Z. Sobia, G. Mustansar Ali, K. Asra, A. Muhammad Awais, N. Usman, and P.-B. Adam, "Novel centroid selection approaches for kmeans-clustering based recommender systems," *Information sciences*, vol. 320, pp. 156–189, 2015.

[10] S. SALVADOR and P. CHAN, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," *16th IEEE international conference on tools with artificial intelligence*, pp. 576–584, 2004.

[11] Z. Hafed, M. Faiz, S. Mohamed, and C. Chaouki, "An improved collaborative filtering recommendation algorithm for big data," *IFIP International Conference on Computational Intelligence and Its Applications*, pp. 660–668, 2018.

[12] G. ADOMAVICIUS and A. TUZHILIN, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, pp. 734–749, 2005.

[13] M. BALABANOVIĆ and Y. SHOHAM, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, pp. 66–72, 1997.

[14] M. Yi, X. Nianhao, T. Ruichun, L. Luo, and Y. Xiaohan, "An efficient similarity measure for collaborative filtering," *Procedia computer science*, vol. 147, pp. 416–421, 2019.

[15] C. Ya-nan, Z. Peng, G. Jing, and G. Li, "Mining large-scale event knowledge from web text," *Procedia computer science*, vol. 29, pp. 478–487, 2014.

[16] B. Fatima Zohra, T. Noria, K.-A. Fatima Zahra, and H. Ilyes-Ahmed, "An adapted approach for user profiling in a recommendation system: Application to industrial diagnosis," *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, pp. 178–130, 2018.

[17] A. STREHL, J. GHOSH, and R. MOONEY, "Impact of similarity measures on web-page clustering," *Workshop on artificial intelligence for web search (AAAI 2000)*, p. 64, 2000.

[18] A. AKINWALE and A. NIEWIADOMSKI, "Efficient similarity measures for texts matching," *Journal of Applied Computer Science*, vol. 23, pp. 7–28, 2015.

[19] K.-S. LIN, "A case-based reasoning system for interior design using a new cosine similarity retrieval algorithm," *Journal of Information and Telecommunication*, vol. 4, pp. 91–104, 2020.

[20] B.-Y. Ricardo and R.-N. Berthier, "Modern information retrieval," *USA: ACM press*, 1999.

[21] V. THADA and V. JAGLAN, "Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm," *International Journal of Innovations in Engineering and Technology*, vol. 2, pp. 202–205, 2013.

[22] M. Maake Benard, O. Sunday O., and Z. Tranos, "A comparative analysis of text similarity measures and algorithms in research paper recommender systems," *2018 conference on information communications technology and society (ICTAS)*, pp. 1–5, 2018.

[23] J. Jeevamol and V. G. Renumol, "Comparison of generic similarity measures in e-learning content recommender system in clod-start condition," *2020 IEEE Bombay Section Signature Conference (IBSSC)*, pp. 175–179, 2020.

[24] R. Roxana, M. Patrick, R. Diederik M., and N. Ann, "Multi-objective multi-agent decision making: a utility-based analysis and survey," *Autonomous Agents and Multi-Agent Systems*, vol. 34, p. 10, 2020.

[25] D. Ali, K. Salil S., and J. Raja, "Multi-agent systems: a survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2016.

[26] L. Álvaro, D. P. Juan F., G. Gabriel Villarrubia, D. L. I. Daniel H., and B. Javier, "Multi-agent system for demand prediction and trip visualization in bike sharing systems," *Applied sciences*, vol. 8, p. 67, 2018.

[27] C. YANG and Z. GU, ZHUOXING an YdAO, "Adaptive urban

design research based on multi-agent system-taking the urban renewal design of shanghai hongkou port area as an example," *INTELLIGENT INFORMED, Proceedings of CAADRIA*, vol. 15, p. 225, 2019.

[28] M. PIPATTANASOMPORN, H. FEROZE, and S. RAHMAN, "Multi-agent systems in a distributed smart grid: Design and implementation," *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–8, 2009.

[29] C. Janae, F. Jason, F. Omar, C.-A. Alba Y., S. A. Eduardo, and B. Eric A., "A novel loss-based energy management approach for smart grids using multi-agent systems and intelligent storage," *Sustainable cities and society*, vol. 39, pp. 344–357, 2018.

[30] V. Sergey, E. Konstantin, N. Anaid, and Y. Arkady, "Multi-agent robotic systems in collaborative robotics," *International Conference on Interactive Collaborative Robotics*, pp. 270–279, 2018.

[31] A. FORESTIERO, "Multi-agent recommendation system in internet of things," *2017 17th IEEE/ACM International Symposium on Cluster*, pp. 772–775, 2017.

[32] E. Dario, S. Davide, C. Domenico, and K. Yehuda E., "Decision support systems based on multi-agent simulation for spatial design and management of a built environment: the case study of hospitals," *Computational Science and Its Applications–ICCSA 2020: 20th International Conference*, pp. 340–351, 2020.

[33] J. Bajo Pérez, d. l. Prieta, Pintado Fernando, J. M. Corchado Rodríguez, and S. Rodríguez González, "A low-level resource allocation in an agent-based cloud computing platform," *Appl. Soft Comput*, vol. 48, p. 716–728, 2016.

[34] J. FIOSINA and M. FIOSINS, "Density-based clustering in cloud-oriented collaborative multi-agent systems," *Hybrid Artificial Intelligent Systems: 8th International Conference*, pp. 639–648, 2013.

[35] A. Michael, F. Armando, G. Rean, J. Anthony D., K. Randy, K. Andy, L. Gunho, P. David, R. Ariel, S. Ion, and Z. Matei, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50–58, 2010.

[36] D. l. P. Fernando, R.-G. Sara, C. Pablo, C. Juan Manuel, and B. Javier, "Survey of agent-based cloud computing applications," *Future generation computer systems*, vol. 100, pp. 223–236, 2019.

[37] D. TALIA, "Cloud computing and software agents: Towards cloud intelligent services," *WOA*, vol. 11, pp. 2–6, 2011.

[38] K. AL-BARZNJI and A. ATANASSOV, "A framework for cloud based hybrid recommender system for big data mining," *J Science, Engineering and Education*, vol. 2, pp. 58–65, 2017.

[39] R. Célia Ghedini, H. M. Aldo, L. Luiz A., A. Aletéia P.F., and M. Alba C.M.A., "Multiagent system for dynamic resource provisioning in cloud computing platforms," *Future Generation Computer Systems*, vol. 94, pp. 80–96, 2019.

[40] M. VALERA, K. RATHOD, and U. CHAUHAN, "A neoteric web recommender system based on approach of mining frequent sequential pattern from customized web log preprocessing," *International Journal of Computer Applications*, vol. 69, 2013.

[41] B. SHOJAIEMEHR, A. M. RAHMANI, and N. N. QADER, "Cloud computing service negotiation: A systematic review," *Computer Standards Interfaces*, vol. 55, pp. 196–206, 2018.

[42] K. M. SIM, "Agent-based cloud computing," *IEEE Transactions on services computing*, vol. 5, pp. 564–577, 2011.

[43] M. VALERA and U. CHAUHAN, "An efficient web recommender system based on approach of mining frequent sequential pattern from customized web log preprocessing."

[44] G. Yu, C. Yue, X. Yuanyan, and B. Xiaojuan, "An effective student grouping and course recommendation strategy based on big data in education," *Information*, vol. 13, p. 197, 2022.

[45] P. VALDIVIEZO-DIAZ and J. CHICAIZA, "Enhanced books recommendation using clustering techniques and knowledge graphs," *Applied Technologies: 4th International Conference, ICAT 2022*, pp. 89–102, 2023.

[46] s. najmeh, A. Somayeh, and J. Mohammadreza, "Developing a group urban tourism recommendation system based on the modified k-means algorithm and fuzzy best-worst method," 2023.

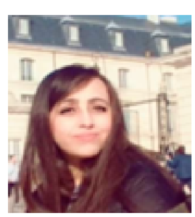
[47] H.-N. Elena, H. Guillermo, G.-G. Ana-Belén, R.-G. Sara, and C. Juan M., "Fog computing architecture for personalized recommendation of banking products," *Expert Systems with Applications*, vol. 140, p. 112900, 2020.

[48] H. SINGH and P. KAUR, "An effective clustering-based web page recommendation framework for e-commerce websites," *SN Computer Science*, vol. 2, pp. 1–20, 2021.

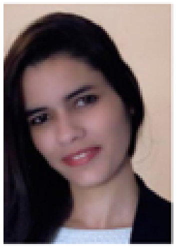
[49] J. MAWANE, A. NAJI, and M. RAMDANI, "Clustering collaborative filtering approach for diftari e-learning platform' recommendation system," *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, pp. 1–6, 2018.

[50] X. PU and B. ZHANG, "Clustering collaborative filtering recommendation algorithm of users based on time factor," *2020 Chinese Control And Decision Conference (CCDC)*, pp. 364–368, 2020.

[51] B. Djihene, T. Noria, N. SAD-HOUARI, and A. Lylia, "A service-based recommendation system to assist decision making in a small and medium company," *3rd International Conference on Intelligent Sustainable Systems (ICISS)*, vol. 6, pp. 63–70, 2020.



Djihene BOURENANE D. Bourenane is a Phd student in computer science department, University of Oran 1 Ahmed Ben Bella and had received her Master degree at this same university in 2018. Her research interests include: collaborative decision support, recommendation systems, artificial intelligence and Cloud computing.



Nawal SAD HOUARI N. Sad Houari is an assistant professor at the Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, USTO-MB. She holds her doctorate thesis in IRAD at the university of Oran1 Ahmed Ben Bella in Algeria in 2017. She received her Master degree in Information System Technologies from the same university in 2013. She is a member of the EWG-DSS (Euro Working Group on Decision Support Systems) since 2016 and she is also a member of LIO (Laboratoire d'Informatique d'Oran) laboratory. Her research interests include Business rules modeling, Artificial Intelligence, Security, Multi-agents system, Knowledge management, Recommender system, Decision support system and Bioinformatics.

Noria TAGHEZOUT N. Taghezout is a professor at University of Oran1 Ahmed BenBella, Algeria. She holds her doctorate thesis in MITT at PAUL SABATIER UNIVERSITY in France in 2011. She also received another doctorate thesis in Distributed Artificial Intelligence from University of Oran1 Ahmed BenBella in 2008. She holds a Master degree in Simulation and Computer aided-design. She conducts her research at the LIO laboratory as a chief of the research group in Modeling of enterprise process by using agents and WEB technologies. Since she studied in UPS Toulouse, she became a member of the EWG-DSS (Euro Working Group on Decision Support Systems). She is currently lecturing Collaborative decision making, Enterprise management and Interface human machine design. Her seminars, publications and regular involvement in Conferences, journals and industry projects highlight her main research interests in Artificial Intelligence.