International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Feb-24)

http://dx.doi.org/10.12785/ijcds/150148

Toward a Systematic Evaluation Approach of Point Of Interest Recommendation Algorithms of a Novel Smart Tourism Tool

Mhamed HADJHENNI[1,2], Nassim DENNOUNI[1,3] and Zohra SLAMA[1]

[1] *EEDIS Laboratory,Djillali Liabes University, Sidi Bel Abbes, Sidi Bel Abbes*
[2] *Computer Science Department,Hassiba Benbouali University, Chlef, Algeria*
[3] *Higher School of Management, Tlemcen, Algeria*

Received 11 May. 2023, Revised 6 Jan. 2024, Accepted 21 Jan. 2024, Published 1 Feb. 2024

Abstract: Intelligent tourism can increase the interest of nomadic tourists in discovering new cities. However, many Points of Interest (POIs) are available, creating an information overload for tourists when choosing POIs to visit. For this reason, CARS (Context-Aware Recommendation Systems) can play an important role by exploiting the experiences of previous tourists and their contexts to recommend attractive POIs. Consequently, choosing the right POI recommendation algorithm (RA) for CARS is crucial because it involves the costly intervention of real tourists during the test phase. In order to make this phase more cost-effective, we can test several RAs simultaneously in order to assess their limitations in terms of cold start and tourist satisfaction. To compare these RAs, we propose in this article an approach called SEPRA (Systematic Evaluation for POI Recommendation Algorithms), which allows us to carry out an initial online evaluation of each tourist during their visit and a second offline evaluation of each CARS after the end of each POI path. To achieve this objective, we designed and implemented a new smart tourism tool that makes POI recommendations using two algorithms: the first is based on tourist/tourist similarity, and the second uses POI/POI similarity. These algorithms use memory-based collaborative filtering and are executed in parallel by our tool in the form of CARSs, incorporating time or weather as context variables. To evaluate these systems during their test phases, Our approach enables: (1) the calculation of prediction accuracy; (2) the examination of the relevance of the recommended POIs; and (3) the estimation of the acceptance rate of the recommendation process. Finally, the experimental results obtained with our approach show that the algorithm based on tourist similarity is more resistant to the cold start problem during the test phase and has a better satisfaction rate than the algorithm based on POI similarity.
Keywords: Smart tourism, CARS, point of interest recommendation, collaborative filtering, online evaluation, offline evaluation.

1. INTRODUCTION

Smart tourism allows people to visit a city or a country by relying on new technologies such as the cloud, the Internet of Things (IoT) and smartphone applications. As a result, today's tourists can transmit, receive, and share a wide variety of information via the Internet at any time, anywhere and with anyone [1]. Indeed, this mode of tourism breaks away from conventional methods based on advertising brochures and tourist guides because smart tourism takes full advantage of ubiquitous technologies (GPS, weather API, etc.) to collect in real-time and in a sustainable way information about points of interest (POIs) visited and on tourists' behaviour. However, processing this web-based data requires the use of intelligent techniques (geographic information systems, big data, data mining, machine learning, etc.) to store, process, analyse and filter the large amount of information produced every day by mobile tourists[2].

Among these information retrieval techniques, recommender systems (RS) are notable for their ability to extract relevant information from the vast amount of knowledge resulting from the interactions between users and items. These systems can recommend POIs such as hotels [3], restaurants [4], e-commerce products [5], tourist destinations [6], movies [7], and scientific articles [8]. In addition, these systems can also predict actions such as criminal behaviour associated with a place [9], the intellectual performance of learners [10], etc. However, in the field of smart tourism, RSs can recommend POIs without taking into account the actual context of the tourist (location, weather, company, means of travel, etc.) [11].

For instance, a traditional RS recommends the best restaurants that match a tourist's taste without taking into account the location of the tourist provided by his smartphone. To address this situation, the new generation of recommendation systems, known as Context-Aware Recommendation Systems (CARSs), can improve the POI recommendation process by bridging the gap between traditional RSs and the spatiotemporal context of mobile tourists [12]. Furthermore, tourists may change their behaviour at any

E-mail address: mhamed.hadjhenni@dl.univ-sba.dz, n.dennouni@univ-chlef.dz, zslama@yahoo.com

time to follow new trends in the travel market, such as participation in international events like the FIFA World Cup, the discovery of a new archaeological site, etc. These events can lead tourists to change their travel plans, neglect their contexts and ignore old evaluations (ratings, comments) of POIs, causing cold start problems. To resolve this problem, real tourists must participate in the testing (learning) phase of CARS to facilitate the bootstrapping of POI recommendations. This phase enables the selection of the optimal recommendation algorithm (RA) for each CARS based on multiple evaluation criteria. These criteria estimate: (1) the efficiency of each RA in predicting ratings close to those given by the tourist, (2) the capacity of the CARS to provide relevant POIs recommendations for each tourist and (3) the POI recommendations systems acceptance ratio.

Current approaches use collaborative filtering techniques for testing new CARS and consider that online evaluation is inseparable from offline evaluation, especially for the comparison of future RAs to be used [13]. However, according to our knowledge, there is no approach to evaluate systematically one or more CARSs during the testing phase to identify the most suitable RA for implementation. To address this need, we propose in this paper an approach that involve real tourists during the test phase to perform two types of evaluation: a first online evaluation of the RA behaviour during tourist visits and a second offline evaluation after the end of each cycle of the test phase. To achieve this goal, we decided to design and develop a novel smart tourism tool that allows different smartphone users to make a systematic evaluation of the CARS during tourist visits. This tool allows (1) adding POIs suggested by tour guides, (2) adding ratings and comments for POIs by tourists, (3) using tourists' contexts in the recommendation process, (4) the parallel execution of several RAs and POIs recommendation to mobile tourists using smartphones and (5) the incremental computation of performance indicators of the different CARS executed simultaneously.

Our work aims to evaluate one or more POI recommendation algorithms(RAs) used by a new smart tourism tool during the test phase. This evaluation facilitates the selection of the best RA for scaling up. For this reason, the objectives of this article can be organised as follows:

1) Establish a literature review on approaches that integrate the tourist's context to improve the satisfaction of travel experiences using smart tourism tools.
2) Propose a modest approach for comparing two RAs of POIs launched in parallel during the test phase. Both algorithms use collaborative filtering based on memory. However, the first RA uses POI/POI similarity and the second RA uses tourist/tourist similarity. In addition, these two RAs should allow the integration or not of contextual information such as time and weather in their POI recommendation process.
3) Use metrics such as CTR, RMSE and F1 for the online evaluation of the different RAs variants during the test phase by real users and for the offline evaluation after the end of the test phase.

In the literature, existing approaches use either online or offline evaluation or a combination of the two approaches. However, to the best of our knowledge, no approach allows a systematic evaluation to be carried out during the test phase of a new smart tourism tool because the data set that allows us to make the comparison with other approaches is not very large. Furthermore, we have not found any studies that propose a comparison method similar to our approach, as these studies are highly dependent on experimental constraints (the number of real users required for the tests, the number of POIs to be visited, etc.) and the contextual information taken into account (weather, location, etc.).

The novelty of our research is to evaluate in real-time the results of POI recommendations obtained from many recommendation systems with context integration and launched in parallel during the test phase of a new smart tourism tool. Our main contribution is the implementation of an evaluation approach capable of choosing the best POI recommendation algorithm thanks to an online evaluation carried out during the test phase and an offline evaluation carried out after the end of the test.

This paper is organised as follows: Section 2 presents a comprehensive review of the literature on CARS for the tourism field and their evaluation techniques in order to elucidate the motivations for our research. Section 3 explains the methods used in our work through the exploration of RAs (recommender algorithms) and CARSs. After that, Section 4 discusses the models developed for designing and implementing our new SEPRA tool to evaluate smart tourism prototypes during testing. Section 5 then concentrates on the experimentation phase of our tool and on explaining the calculations of the performance indicators for the CARS deployed. The results of this experiment will be analysed and discussed in Section 6. Finally, Section 7 summarises the contributions made by our paper and proposes perspectives for our work.

2. LITERATURE REVIEW

During a trip, traditional RSs propose places recommendations to visit or services to discover by their users based solely on the relationship between the tourist and the POI. Therefore, CARSs can play a more significant role than RSs because these systems can easily integrate tourists' contextual information retrieved from their smartphones. Indeed, depending on the CARS used, contextual information is extracted (1) implicitly[14] [15], (2) explicitly [16] [17], or inferred through the machine learning method [18]. These contextual information extraction techniques help CARS enrich the dataset used in future POI recommendations. However, this dataset may be significant (after scaling) and/or not very meaningful during the testing phase (data sparsity). To address both of these issues, we looked at CARSs that use collaborative filtering with an explicit approach based on tourist ratings and generalised contextual information. However, there are several techniques in the literature for developing this type of CARS. Kulkarni and Rodd [19] classified these techniques into two families: ”bio-inspired computing techniques” and ”statistical computing techniques”. The first one contains techniques such as artificial neural networks, genetic algorithms, ant colony optimisation, etc. and the second one includes techniques such as matrix factorisation, tensor factorisation, Bayesian theory and learning, the K-nearest neighbour algorithm, support vector machines, hidden Markov models, etc. The algorithms used can be classed into two categories: (1) memory-based algorithms that use heuristics to predict the target user's rating for an item based on partial information available about that user; and (2) algorithms based on machine learning models [19].

To implement a new CARS, the K-Nearest Neighbour heuristic appears more suitable for this situation. This approach calculates similarities between active tourist and other tourists to predict their future ratings for recommended POIs. Several works in the literature employ this collaborative filtering approach and below, nine of them are cited.

In [20], a heuristic approach uses the user-user similarity computed from the Pearson correlation coefficient to predict a tourist's preferences based on his neighbours who share the same tastes. This collaborative filtering approach integrates the user's location as the only context dimension for recommending tourist activities for next time. In [21], another heuristic uses the demographic characteristics of tourists, their geographical distance to the event (context: Location) and the tourist's time of arrival to calculate user-user similarities. This calculation allows classifying tourists by category in order to predict their preferences.

Bagci et al. implemented a heuristic approach based on user-user similarity using a graph that combines the current context of the user and the social relationships deduced from a location-based social network (LBSN). This approach predicts the ratings associated with different tourist activities and recommends the best ones[22]. On the other hand, Mingxin Gan and Ling Gao have integrated the psychological effects and preferences of a user into the POI recommendation heuristic. This process uses check-ins to calculate the similarity of POI preferences between users of an LBSN in order to predict their future places to visit [23]. Tenemaza et al. also proposed a tourist trip planning heuristic that analyses in real-time the constraints of the user and POI. This heuristic uses a function that includes several contextual factors and generates a personalised recommendation of POIs and itineraries optimised for tourists[24]. Arif et al. used a heuristic based on destination ratings to visualise halal POIs for potential tourists. This approach measures the cosine similarity between users based on a tourism dataset specific to the Batu City (Indonesia)[25].Lin Wan et al. proposed iTourSPOT, a contextual framework that exploits user check-in sequences and contextual information to provide personalised POI recommendations. The system consists of modules for extracting contextual session characteristics (weather, location) and user preference learning models. Predictions of recommendation scores are generated, taking into account temporal weight and collaborative filtering, to recommend the best POIs for each user[26]. Liu et al. proposed a Context Awareness Attention Network model, which is based on an attention network that integrates context into POI recommendation. This model comprises a contextual interaction layer, a geo-temporal attention network and a co-attention network to infer dynamic tourist preferences. This approach calculates the tourist's probability of visiting POIs and then ranks the POIs to generate an ordered list of the best recommendations by incorporating context into the recommendation process[27]. Finally, Laskoski et al. have proposed a mobile application featuring a recommendation system designed for personalised visits to cultural heritage sites. Their system combines a context model with a collaborative filtering method, employing Pearson's correlation coefficient to determine the similarity between users. Contextual modeling and post-filtering techniques are employed to enrich the recommender system, considering factors such as the tourist's current position and the time spent during the visit[28].

The advantages of these works are the use of collaborative filtering and the integration of context (see Table I). Previous work suffers from cold start problems, which occur when a new tourist or a new POI are added, and the problem of sparsity, as well as the lack of performance evaluation for some work (see Table I). In the studies above, no approach conducts an online evaluation of RSs in real time. This method of evaluation far surpasses offline evaluations and user surveys, as it is the only way of assessing the degree of satisfaction with the use of their RAs with a real group of testers. In the literature, we found that the work mentioned above frequently uses context dimensions such as visiting time and weather. On the other hand, in order to mitigate the cold start problem, we are convinced that hybrid approaches that combine several RAs and integrate several contexts can give good results in terms of user satisfaction. For these reasons, in our approach, we have merged the results of several RAs (Hybridisation) and adopted pre-filtering as a solution for integrating context using the relaxation principle, which allows only one context variable to be considered at a time[29]. Our approach also uses a new type of evaluation based on a combination of online evaluation during the test phase and offline evaluation after the test. These two types of evaluation make it possible to monitor the evolution of our systems over time to identify periods of performance, stagnation or regression of our RSs.

Table I compares some works in the literature to our approach. These works agree on the need to use a heuristic based on collaborative filtering for new tourism CARS,

TABLE I. COMPARISON GRID OF LITERATURE WORKS WITH OUR APPROACH

work	RS Algorithm	Context integration (context dimensions)	Evaluation method (metrics used)
[20]	User-user similarity	Not indicated (user's location)	No evaluation (No metrics used)
[21]	User-user similarity	Context modelling (geographical distance, time of arrival)	Offline evaluation user studies
[22]	User-user similarity	Not indicated (social relations,current location)	Offline evaluation (precision, recall, F1)
[23]	POI-POI similarity	Not indicated (Time)	Offline evaluation (precision, recall, F1)
[24]	Genetic algorithm and k-means	Pre filtering (Time,budget)	Offline evaluation (precision, recall, F1) and user studies
[25]	User-user similarity	Pre filtering (facilities,services available)	Offline evaluation (accuracy, precision, recall, F1) and user studies
[26]	Session-based recommendation	Pre filtering (Time,weather, location)	Offline evaluation (precision, recall, F1)
[27]	Model based on probability	Context modeling (Time,location, social)	Offline evaluation (accuracy,precision, recall)
[28]	User-user similarity	Context modeling pre filtring (Time,location)	Offline evaluation (MAE,RMSE) and user studies
Our approach	Tourist/tourist similarity POI-POI similarity	Pre filtering (Time of arrival and weather)	Offline evaluation (RMSE, Precision, Recall, F1) , Online evaluation (CTR) and user studies in progress

especially during the test phase, which is necessary to initiate POI recommendations. Moreover, to our knowledge, there is no formal method in the literature to make instantaneous parallel evaluation of several RS or CARS for smart tourism or any other domain. Indeed, each new tourism tool integrating an RS or CARS, we can use several parameters for evaluating the quality of recommendations to choose the best algorithm that suits tourists' preferences. These parameters should evaluate, on the one hand, the quality of the recommendations provided by the RS/CARS and, on the other hand, the different objectives addressed by tourism promoters, such as the number of clicks per POI(CTR), the number of sites visited, the number of comments, the time spent per POI, tourists' ratings, etc. To evaluate these parameters, the following approaches can be used: (1) offline evaluation of website or mobile application history; or (2) online evaluation of implicit tourist interactions and explicit POI rating. The first one is generally independent of the user interface and therefore simpler to implement, reproducible, fast and able to compare an unlimited number of models. In contrast, the second approach is closely related to the user interface, domain-dependent, and produce non-reproducible results[30].

Evaluating recommendation systems is a crucial stage in validating their usage. Consequently, several studies have investigated the impact of time on online evaluations[31] [32] [33] because this temporal factor aids in interpreting the acceptance of the obtained recommendation results[34]. Online evaluation is the best method to evaluate RSs and CARS. This method measures the acceptance of the recommendation in the real world; the most parameter used in this case is the click-through ratio (CTR). However, this type of evaluation is very expensive and time-consuming compared to any other evaluation method. For these two reasons, offline evaluation remains the simplest and most practical evaluation method which does not require any interaction with real users[30] .To address both of these issues, researchers consider user studies to be an optimal evaluation method[35]. In this type of study, users are asked to rate their overall satisfaction with the RS/CARS through a survey. Nevertheless, the results of this study often depend on the questions asked and the people interviewed[36]. For these reasons, we first used the online evaluation by calculating mainly indicators like CTR during tourists' visit and in a second step; the parameters like MAE/RMSE, precision/recall and F1 are incrementally calculated according to the evolution of our application history. Then, a satisfaction study was realised when our system reached maturity by registering a considerable number of users. Contrary to the previous works, our contribution exploits the flexibility offered by a heuristic based on the K-nearest neighbours and makes the test phase more profitable by experimenting with several CARSs simultaneously.

3. METHODS

This section delineates the precise methodologies employed in this research for recommendation. It covers the use of recommendation algorithms employing memory-based collaborative filtering, the selection of context variables, and their integration methods into the pre-filtering recommendation process. As shown in Figure 1, the system below allows (1) the introduction of contextual information such as the tourist's arrival time and the day's weather. Then, (2) this system uses the ubiquitous context of the tourist to pre-filter the data set. Then, (3) runs two RAs in parallel: the first RA is called RPBP (Recommendation POI Based on POI similarity) and the second RA is called RPBT (Recommendation POI Based on Tourist similarity). Finally, (4) these two RAs provide lists of K-dimensional POIs ranked according to their relevance. These two RAs (RPBP and RPBT) can (1) ignore the tourist's context to work as simple RSs and can (2) integrate contextual information such as the tourist's arrival time and the day's weather to operate as CARSs.

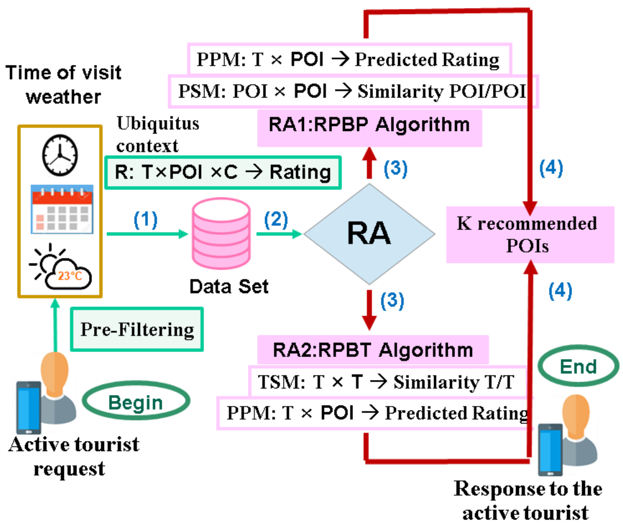

Figure 1. The functioning of our system

A. *RSs functioning*

In our research, memory-based collaborative filtering algorithms are employed, which depend on the idea that similar users have comparable habits in rating behaviour and that comparable items are rated similarly. The following steps are done:

1) Calculate the similarity sim(x, y), which reflects the correlation or distance between two tourists or two POIs, x and y.
2) Calculate the predicted score that can be attributed to the current tourist for a given POI.

In the following, The RPBT algorithm and the RPBP algorithm are presented below. These algorithms allow us to achieve two RSs that use the same input data set.

1) *RPBT algorithm*

In the literature, several similarity measures[37] allow us to calculate the degree of similarity between two given vectors. However, in the field of RSs, similarity based on Pearson correlation seems to be the most commonly used metric for calculating similarities between users[38]. Our two algorithms use a variant of this type of similarity called "Pearson's empirical correlation coefficient"[39] described by Eq.(1):

$$\mathrm{sim}(t,v)=\frac{\sum_{i\in I_{tv}}\left(\mathrm{rat}_{t,i}-\overline{\mathrm{rat}_t}\right)\left(\mathrm{rat}_{v,i}-\overline{\mathrm{rat}_v}\right)}{\sqrt{\sum_{i\in I_{tv}}\left(\mathrm{rat}_{t,i}-\overline{\mathrm{rat}_t}\right)^2}\sqrt{\sum_{i\in I_{tv}}\left(\mathrm{rat}_{v,i}-\overline{\mathrm{rat}_v}\right)^2}} \quad (1)$$

With:

- $\mathrm{Sim}(t,v)$: The similarity between the active tourist t and its neighbour v
- $I_{(t,v)}=I_t\cap I_v$: The set of POIs co-rated by the active tourist t and the tourist v.
- $\mathrm{rat}_{(t,i)}$: Score assigned by the tourist t to the POI i.
- $\overline{rat_t}$: The average rating of tourist t for all his POIs evaluation history.

After computing the similarities between tourists using Eq.(1), the prediction of an active tourist rating for a POI i is calculated using Eq.(2) [40].

$$pred(t,i)=\overline{rat_t}+\frac{\sum_{v\in V_t}(rat_{v,i}-\overline{rat_v})\cdot sim(t,v)}{\sum_{v\in V_t}|sim(t,v)|} \quad (2)$$

With:

- V_t: Set of k nearest neighbours of the active tourist t.
- $\mathrm{rat}_{(v,i)}$: Score assigned by the tourist v to the POI i.

Algorithm 1 represents the pseudocode of our first algorithm, called RPBT (Rating Prediction Based on Tourist/Tourist Similarity), which is based on similarity between tourists (tourist/tourist similarity).

2) *RPBP algorithm*

The RPBP algorithm relies on computing the similarity between two POIs, denoted as i and j, which is determined by Eq.(3):

$$\mathrm{sim}(i,j)=\frac{\sum_{t\in T_{ij}}(\mathrm{rat}_{t,i}-\overline{\mathrm{rat}_i})(\mathrm{rat}_{t,j}-\overline{\mathrm{rat}_j})}{\sqrt{\sum_{t\in T_{ij}}(\mathrm{rat}_{t,i}-\overline{\mathrm{rat}_i})^2}\sqrt{\sum_{t\in T_{ij}}(\mathrm{rat}_{t,j}-\overline{\mathrm{rat}_j})^2}} \quad (3)$$

With:

- T_{ij}: Set of tourists who co-rated POI i and POI j.
- $\mathrm{rat}_{(t,j)}$: Score assigned by the tourist t to the POI j.
- $\overline{rat_i}$ (resp $\overline{rat_j}$): Average ratings of POI i (resp POI j) .

Algorithm 1: RPBT

Input: R: Tourists-POIs Rating Matrix;
T: Vector of tourists;
POI: Vector of POIs;
$AvgRat$: Vector of tourists' Average rating;
k: Total number of POIs to be recommended;
TId: Id of the current tourist.
Output: TSM: Tourists similarity matrix;
PPM: POIs prediction matrix;
$LRecPOI$: List of POIs' id to be recommended.

// Similarity between tourists using Eq.(1);
for *each tourist* $T(i)$ **do**
  for *each tourist* $T(j) \neq T(i)$ **do**
    $N = 0$; $Di = 0$; $Dj = 0$;
    for *each POI* p **do**
      if $(R(i,p) \neq 0$ *and* $R(j,p) \neq 0)$ **then**
        $N = N + (R(i,p) - AvgRat(i)) \cdot (R(j,p) - AvgRat(j))$;
        $Di = Di + (R(i,p) - AvgRat(i))^2$;
        $Dj = Dj + (R(j,p) - AvgRat(j))^2$;
    $TSM(i,j) = \frac{N}{\sqrt{Di} \cdot \sqrt{Dj}}$;
// Prediction computation using Eq.(2);
for *each tourist* t **do**
  for *each POI i not rated by the tourist t* **do**
    $N = 0$; $D = 0$;
    for *each tourist* $v \neq t$ *who rated the POI i* **do**
      $N = N + TSM(t,v) \cdot (R(v,i))$;
      $D = D + |TSM(t,v)|$;
    $PPM(t,i) = AvgRat(t) + \frac{N}{D}$;
// Select k predictions for the current tourist;
$LRecPOI = Descendingsorting(PPM, TId, k)$;
return $LRecPOI$;

Algorithm 2: RPBP

Input: R: Tourists-POIs Rating Matrix;
T: Vector of tourists;
POI: Vector of POIs;
$AvgRatP$: Vector of tourists' Average rating;
k: Total number of POIs to be recommended;
TId: Id of the current tourist.
Output: PSM: POIs similarity matrix;
PPM: POIs prediction matrix;
$LRecPOI$: List of POIs' id to be recommended.

// Similarity between POIs using Eq.(3);
for *each POI i* **do**
  for *each POI j* **do**
    $N = 0$; $Di = 0$; $Dj = 0$;
    for *each Tourist t of T* **do**
      if $(R(t,i) \neq 0$ *and* $R(t,j) \neq 0)$ **then**
        $N = N + (R(t,i) - AvgRatP(i)) \cdot (R(t,j) - AvgRatP(j))$;
        $Di = Di + (R(t,i) - AvgRatP(i))^2$;
        $Dj = Dj + (R(t,j) - AvgRatP(j))^2$;
    $PSM(i,j) = \frac{N}{\sqrt{Di} \cdot \sqrt{Dj}}$;
// Prediction computation using Eq.(4);
for *each tourist* t **do**
  for *each POI i not rated by the tourist t* **do**
    $N = 0$; $D = 0$;
    for *each POI j rated by tourist t* **do**
      $N = N + PSM(i,j) \cdot R(t,j)$;
      $D = D + |PSM(i,j)|$;
    $PPM(t,i) = \frac{N}{D}$;
// Select k predictions for the current tourist
$LRecPOI = Descendingsorting(PPM, TId, k)$;
return $LRecPOI$;

After computing the similarities between POIs, the active tourist t rating prediction is calculated for a POI i using Eq.(4) [41].

$$pred(t,i) = \frac{\sum_{j \in I_t} (rat_{t,i} \cdot sim(i,j))}{\sum_{j \in I_t} |sim(i,j)|} \tag{4}$$

Where I_t is the set of POIs rated by tourist t.

In the following, Algorithm 2 represents the pseudocode of our proposed algorithm called RPBP(Rating Prediction Based on POI/POI Similarity), which is based on POIs (POI/POI similarity).

3) Discussions

RPBT and RPBP algorithms allow us to achieve two RSs. The first system uses the RPBT algorithm with a two-dimensional data set (tourist, POI, rating) as input and the second uses the RPBP algorithm with the same data set. These two systems are executed in parallel for the computation of POI predictions.

As shown in Figure 1, each of the RAs (RPBT and RPBP) calculates the POI recommendations for an active tourist by following three steps:

1) The system introduces the data set in a matrix form which contains the previous tourists' ratings at the different POIs already displayed in our interactive tourism map.
2) From this matrix, our system uses the RPBP and the RPBT algorithms in parallel to predict the POIs ratings to the active tourists.
3) Using the results of these two algorithms, our system selects the k POIs with best-predicted scores to recommend them for the active tourist.

B. CARSs functioning

In the context-aware recommendation framework, in addition to the user and the item, the context dimension is added in the following form:

R: User × Item × Context – > Rating

The contextual recommendation process can assume one of three forms, depending on the use manner of contextual information: pre-filtering, post-filtering, or contextual modelling[42].

Pre-filtering techniques use context to eliminate irrelevant rating information. In contrast, post-filtering applies context as a filter to the recommendation system after the recommendations have been generated. Lastly, in contextual modelling, the system employs predictive models that are based on the context.

Our CARS employs a contextual pre-filtering approach, which integrates contextual information to select the most relevant 2D data (users x items) based on the tourist's current context, facilitating the generation of POIs recommendations. Table II provides a description of the context variables used in the pre-filtering process.

In Figure 1, the pre-filtering operation based on "arrival time" or "weather of the day" must be carried out before launching our two algorithms (RPBP and RPBT) presented previously. In order to ensure compliance with the relaxation principle, which allows only one context variable to be included at a time[29].

Our proposition of the pre-filtering algorithm based on Weather is described in Algorithm 3.

Algorithm 3: Pre-filteringWeather

```
Input: R: Tourists-POIs Rating Matrix;
T: Vector of tourists;
POI: Vector of POIs;
Act_Weather: Context weather during a tourist visit.
Output: datasetC1: filtered Tourists-POIs Rating Matrix;
for each tourist T(i) do
    for each POI p do
        if (R(i, p) ≠ 0) then
            Rat_Weather = R(i, p).weather;
            if ( Rat_Weather = Act_Weather) then
                dataSetC1(i, p) = R(i, p);
            else
                dataSetC1(i, p) = 0 // ignored rating;
return datasetC1
```

TABLE II. VALUES OF CONTEXT VARIABLES USED

Context variables	Values of context
Time	Early Morning, Morning, afternoon, evening
Weather	Sunny, rain, snow, clear, cloudy

In the following, Algorithm 4 which uses Time-based pre-filtering is similar to Algorithm 3. However, Algorithm 3 uses the "weather" context variable while Algorithm 4 uses the "arrival time" context variable.

Algorithm 4: Pre-filteringTime

```
Input: R: Tourists-POIs Rating Matrix;
T: Vector of tourists;
POI: Vector of POIs;
Output: datasetC2: filtered Rating Matrix;
switch Act_Time do
    case Early morning: do
        InfV=01:00, SupV=05:59; break;
    case Morning: do
        InfV=06:00, SupV=11:59; break;
    case Afternoon: do
        InfV=12:00, SupV=17:59; break;
    case Evening: do
        InfV=18:00, SupV=23:59; break;
for each tourist T(i) do
    for each POI p do
        if (R(i, p) ≠ 0) then
            Rat_time = R(i, p).time;
            if (InfV ≤ Rat_time ≤ SupV) then
                dataSetC2(i, p) = R(i, p);
            else
                dataSetC2(i, p) = 0 // ignored rating;
return datasetC2;
```

4. PROPOSED MODEL

In this section, our main contributions are summarized as follows: First, a model of tourist visit scenarios and the corresponding data models are designed. Following that, the technical details of the implementation of our approach called SEPRA (Systematic Evaluation for POI Recommendation Algorithms) are provided in detail. Finally, this approach will be used to compare the RPBP algorithm and the RPBT algorithm through the RSs and CARSs (described in Section 3) used during the evaluation phase of our smart tourism tool.

A. System Model

Our smart tourist tool is used by four types of users (see Figure 2): (a) the tourist, (b) the touristic guide, (c) the tourism operator, and (d) the system manager of the smart tourism application. Each tourist can create an account; the system administrator can validate it or not. Then the tourist receives contextual information such as their location on the map, the weather associated with the moment of their visit, etc. and chooses whether to visit or not the recommended POIs by the RS/CARS of our application. While visiting a place, the tourist can rate and/or comment on POIs. The tourist guide can assume the role of the tourist thanks to the inheritance relationship. This actor can also make suggestions to the system administrator, such as adding new photos for existing POIs or proposing new POIs to be included in the database of places. In addition, the tourism operator (travel agency, tourist board, etc.) can provide pricing for tourist routes that include recommended POIs.

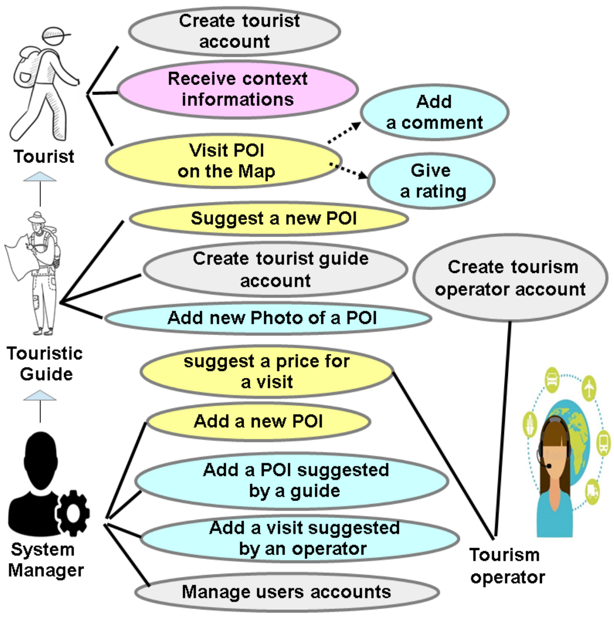

Figure 2. Use cases diagram of our smart tourism tool.

The data related to our tool is stored in the relational tables described by the UML class diagram (see Figure 3). Firstly, this diagram structures the data concerning POIs, users and POI/user interactions using the classes "POI", "User" and "transition". Next, the "POI_suggestion" class keeps track of the POI recommendations provided by our CARS and RSs to the various users of our system. Finally, the "feedback" class collects online user evaluations for the different POI recommendations.

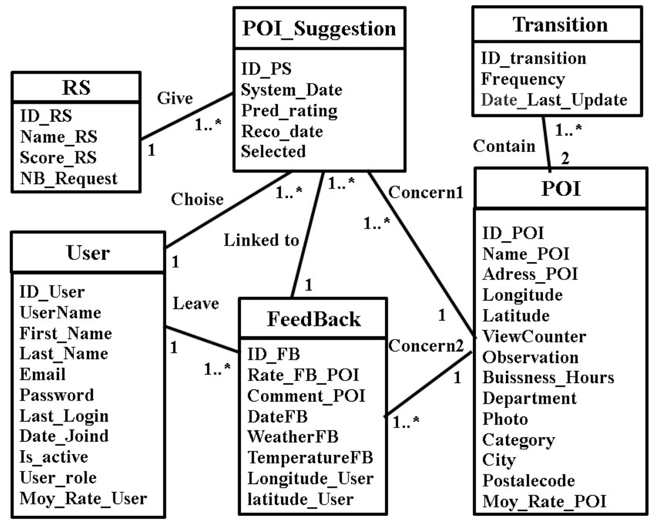

Figure 3. Class diagram of our smart tourism tool.

B. Architecture and Working

To experiment with these two RAs, we have set up our prototype, which uses OpenStreetMap [1] as a cartographic support, which is a free Geographic Information System based on the Web (Web GIS) that facilitates the processing and display of geographic information on maps via the Internet.

Our prototype is a web application that uses the Apache web server to receive requests and send responses to tourists. The data in our application is managed by the MySQL database management system, which is free and open source. This prototype allows tourists to visualise in real time their location as well as the positions of POIs on an interactive map and provides personalised POI recommendations that take into account the weather conditions of each tourist's position and each POI, thanks to the Weather API[2]. This environment makes possible the parallel execution of memory-based RSs and CARS that can integrate the "time" and "weather" dimensions. However, in order to be able to compare the two RAs used, we have proposed a modest approach called SEPRA (Systematic Evaluation for POIs Recommendation Algorithms). This approach is based on online evaluation during the test phase of our new tool by real users and uses offline evaluation after the end of the test phase.

According to Figure 4, the main steps in the operation of our system are described as follows:

[1] https://www.openstreetmap.org/
[2] https://www.weatherapi.com/

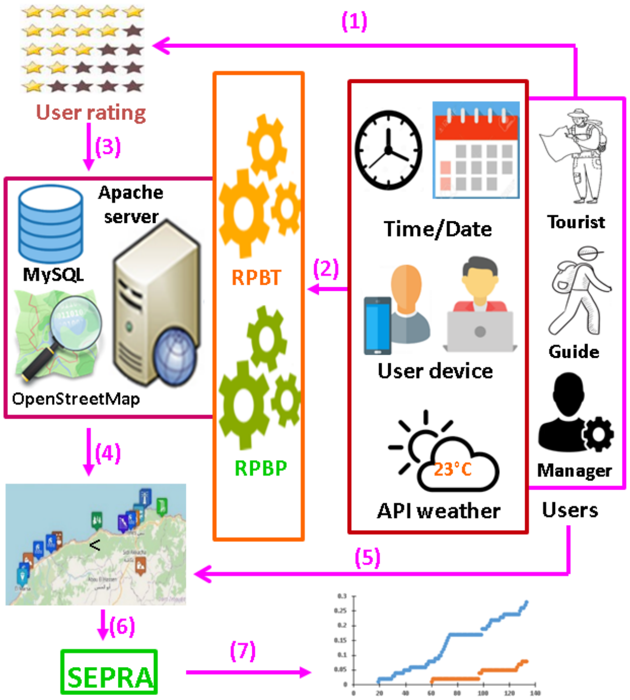

Figure 4. The SEPRA smart tourism tool architecture.

1) The user evaluates a POI and asks the system for recommendations for the next POI to visit.
2) The system automatically introduces the profile of each user using contextual variables.
3) The system calculates predictions for the POIs to visit based on the user's evaluation and the context variables deduced automatically.
4) The system displays on a map the list of POI recommendations resulting from the output of each RS or CARS using the RPBT and RPBP algorithms, together with the contextual rating entered by the user.
5) The user can choose (or not) a POI from the POI recommendations made by the RSs or CARS and can give (or not) a new rating for the chosen POI.
6) Based on the user's deliberate choices, our SEPRA approach calculates the parameters MAE, RMSE, accuracy, recall, F1 and CTR and saves these parameters.
7) The SEPRA approach retrieves these parameters in order to compare the two algorithms, RPBT and RPBP, during the test phase.

In the following, Algorithm 5 explains the steps required to compute the parameters RMSE, F1 and CTR for each user and each system.

In the following, we explain how our SEPRA approach enables us to carry out a systematic evaluation per tourist in order to compare different variants of the RPBT and RPBP algorithms. This approach simultaneously evaluates six systems:

Algorithm 5: SEPRA

Input: R: Tourists-POIs Rating Matrix;
T: Vector of tourists . POI: Vector of POIs;
$k1$, $k2$, $k3$, $k4$, $k5$, $k6$: Total number of POIs to be recommended by RA and CARS;
IdU: Id of the actual user
Output: $MetricsByTourist$: Evaluations metrics by tourist;
$MetricsByRA$: Evaluations metrics by RA

$C1\_Weather$= Retrieve Context weather;
$C2\_Time$= Retrieve Context time;
$dataSetC1$= Pre-filteringWeather(R, $C1\_Weather$);
$dataSetC2$= Pre-filteringTime(R, $C2\_Time$);
$Rec1$= RPBT(R, T, POI, $AvgRat$, $k1$, IdU);
$Rec2$=RPBP(R, T, POI, $AvgRat$, $k2$, IdU);
// RA CARS Based on Weather;
$Rec3$= RPBT($dataSetC1$, T, POI, $AvgRat$, $k3$, IdU);
$Rec4$= RPBP($dataSetC1$, POI, $AvgRat$, $k4$, IdU);
// RA CARS Based on Time;
$Rec5$=RPBT($dataSetC2$, T, POI, $AvgRat$, $k5$, IdU);
$Rec6$= RPBP($dataSetC2$, T, POI, $AvgRat$, $k6$, IdU);
Rec_POI= Concatenate($Rec1$, $RS1$,$Rec2$, $RS2$,$Rec3$, $RS3$,$Rec4$, $RS4$,$Rec5$, $RS5$,$Rec6$, $RS6$);
Display Rec_POI;
Display Rec_POI on MAP;
On EVENT OnToursit_feedBack()
    Date1= Retrieve time of the visit;
    $MetricsByTourist$=Calculate_Eval_Metric_ByTourist; //According to (6), (9), (12), (14), (16), and (19);
    Save_Eval_Metric_Tourist($date1$)
On EVENT OnEndTrip()
    $Date2$ = Retrieve the current time;
    $MetricsByRA$= Calculate_Eval_Metric_ByRS; //According to (7), (10), (13), (15), (17), and (20);
    Save_Eval_Metric_RS($date2$);
return $MetricsByTourist$, $MetricsByRA$;

1) RS based on the RPBT algorithm
2) RS based on the RPBP algorithm
3) CARS integrating the context variable 'tourist arrival time' using the RPBT algorithm
4) CARS integrating the context variable 'tourist arrival time' using the RPBP algorithm
5) CARS integrating the context variable 'weather of the day' using the RPBT algorithm.
6) CARS integrating the context variable 'weather of the day' using the RPBP algorithm

5. EXPERIMENTATION, RESULTS AND ANALYSIS

In this section, the focus is on validating the use of our new smart tourism tool. For this purpose, nomadic tourists were invited to test our application's functionalities and contribute to the enrichment of the database with new POIs. The experimentation of our tool will be presented through a calcul explanation of the performance indicators of our RAs.

A. Experimental Setup

During the test phase of our smart tourism prototype, the website was put online for 12 weeks to calculate the experimental evaluation parameters. This website features an interactive map of tourist sites in the Chlef region(Algeria), recorded the browsing history of 30 tourists. These users used two types of terminals: PCs and smartphones. Tourists using PCs use RSs because PCs do not provide context, whereas smartphone users can integrate their contexts using CARSs. These nomadic tourists freely tested the functionalities offered by our application and added new POIs to our database.

B. Performance Metrics

In the literature, RS/CARS evaluation approaches (online evaluation, offline evaluation and user studies) use several parameters[43]. However, this section explains the parameters used to evaluate our developed tool SEPRA:

1) Top-K

This parameter defines the number of items that a RS or CARS will suggest to a user within a single user session. In this context, all eligible items receive scores, but only the K items with the highest scores are displayed as recommendations to the currently active user. For example, the Top-5 recommend at most five POIs to each identified tourist in a session by his username and password[44].

2) CTR (Click Through Rate)

The click-through rate (CTR) measures the frequency with which users accept recommendations [13]. This parameter represents the ratio of clicked recommendations to the displayed recommendations. Therefore, the CTR is calculated by Eq.(5):

$$CTR = \frac{number of accepted POI recommendations}{number of displayed POI recommendations} \quad (5)$$

For example, if during one or more login sessions of a given tourist, an RS displays 200 possible POI recommendations (depending on the Top K used) and this tourist clicked only on 11 POIs, then the CTR is 5.5%.

This ratio measures the effectiveness of an RS because when a tourist clicks on a recommended POI, the system infers that the user indicated interest in the recommendation, even if the evaluation is negative[13]. In this article, we used two types of CTRs: the CTR per tourist named $CTR_{RS(i)}$ and the CTR per RS named CTR_{RS}. The $CTR_{RS(i)}$ evaluates the acceptance rate of POI suggestions provided by an RS for a given tourist i according to equation (6):

$$CTR_{RS}(i) = \frac{card\{selectedPOIs_{bytourist_i}\}}{card\{RecommendedPOIs_{byRS}\}} \quad (6)$$

Where *card* refers to the number of elements in a given set.

The CTR_{RS} calculate the acceptance rate of each RS using the CTR per tourist, as shown in equation (7):

$$CTR_{RS} = \frac{\sum_{i=1}^{m} CTR_{RS}(i)}{card\{m\}} \quad (7)$$

Where: m: Set of all real ratings given by tourists on a set of POIs.

3) MAE (Mean Absolute Error)

MAE is the absolute difference between predicted ratings by the RS/CARS and actual ratings given by the user (tourist)[45]. MAE is calculated by Eq.(8):

$$MAE = \frac{\sum_{(i,j)\in m} \left|\mathrm{pred}_{(T(i),\mathrm{POI}(j))} - \mathrm{rat}_{(T(i),\mathrm{POI}(j))}\right|}{\mathrm{card}(m)} \quad (8)$$

With:

- (i,j) represent the tourist T(i) who rate the POI(j)
- $pred_{T(i),POI(j)}$: The predicted rating by tourist T(i) for POI(j).
- $rat_{T(i),POI(j)}$: Score assigned by tourist T(i) for POI(j).

MAE measure determines the relevance of the estimated scores. This parameter is calculated for a test set with real ratings obtained via a user study or online experiment. We hide these ratings during offline experiments [13].

Two types of MAEs are used: The $MAE_{RS(i)}$ calculate MAE for a given tourist i according to equation (9):

$$MAE_{RS}(i) = \frac{\sum_{(i,j)\in mt(i)} |pred_{T(i),POI(j)} - rat_{T(i),POI(j)}|}{card\{mt(i)\}} \quad (9)$$

Where: $mt(i)$: The set of scores assigned by tourist i on a set of POIs.

The MAE_{RS} which calculate the MAE of each RS using MAE per tourist, is computed using equation (10).

$$MAE_{RS} = \frac{\sum_{i=1}^{m} MAE_{RS}(i)}{card\{m\}} \quad (10)$$

4) RMSE (Root Mean Squared Error)

The RMSE is almost similar to MAE, but this parameter squares prediction errors, which increases the influence of large errors [46]. This parameter is calculated by Eq.(11):

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in m}(pred_{T(i),POI(j)} - rat_{T(i),POI(j)})^2}{card\{m\}}} \quad (11)$$

We used Two types of RMSEs: the RMSE per tourist $RMSE_{RS(i)}$ and the RMSE per RS $RMSE_{RS}$. The $RMSE_{RS(i)}$ calculate RMSE for a given tourist i according to equation (12):

$$RMSE_{RS}(i) = \sqrt{\frac{\sum_{(i,j)\in mt(i)}(pred_{T(i),POI(j)} - rat_{T(i),POI(j)})^2}{card\{mt(i)\}}} \tag{12}$$

The $RMSE_{RS}$ calculate the RMSE of each RS using the RMSE per tourist, as shown in equation (13):

$$RMSE_{RS} = \sqrt{\frac{\sum_{i=1}^{m}(RMSE_{RS}(i))^2}{card\{m\}}} \tag{13}$$

5) Precision and recall

Precision as well as recall are commonly used to evaluate RSs based on a Top-N list of POIs[47] [48]. In fact, the higher these parameters are, the better the performance of RA [44]. The precision of an RS for a tourist i measures the proportion of truly relevant POIs recommendations in the Top-N list of POIs. We compute precision for each tourist i using equation (14):

$$Precision_{RS}(i) = \frac{Card\{POIs_{recandpert}\}}{Card\{POIs_{rec}\}} \tag{14}$$

Where: $POIs_{recandpert}$ is the set of recommended POIs which are pertinent for tourist *i*. $POIs_{rec}$ is the set of recommended POIs for tourist *i*.

The $Precision_{RS}$ calculate the aggregation of the Precision for all tourists, as shown in equation (15):

$$Precision_{RS} = \frac{\sum_{1}^{m} Precision_{RS}(i)}{card\{m\}} \tag{15}$$

Recall measures the ratio of truly relevant POIs recommendations among all relevant POIs for a tourist. This ratio is computed for each tourist i using equation (16).

$$Recall_{RS}(i) = \frac{Card\{POIs_{recandpert}\}}{Card\{POIs_{pert}\}} \tag{16}$$

Where: $POIs_{pert}$ is the set of pertinent POIs for tourist i. All tourists Recall are aggregated to obtain the $Recall_{Rs}$ using equation (17):

$$Recall_{RS} = \frac{\sum_{1}^{m} Recall_{RS}(i)}{m} \tag{17}$$

6) F1 score

After computing precision and recall, the F1 measure combining these two parameters allows us to make another type of evaluation of RSs [48][49] [50] and it is calculated by Eq.(18):

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{18}$$

This parameter is computed for each tourist i using equation (19) and for each RS using equation (20):

$$F1_{RS}(i) = \frac{2 \cdot Precision_{RS}(i) \cdot Recall_{RS}(i)}{Precision_{RS}(i) + Recall_{RS}(i)} \tag{19}$$

$$F1_{RS} = \frac{2 \cdot Precision_{RS} \cdot Recall_{RS}}{Precision_{RS} + Recall_{RS}} \tag{20}$$

C. Result

During this test phase, indicators per tourist such as CTR_t, MAE_t, $RMSE_t$, $precision_t$, $recall_t$ and $F1_t$ are calculated during tourists' visits. Then, parameters per RS such as CTR_{RS}, MAE_{RS}, $RMSE_{RS}$, $precision_{RS}$, $recall_{RS}$ and $F1_{RS}$ are calculated according to the evolution of the number of user sessions. In the following, our approach for evaluating our algorithms is explained through a simple example (see Table III) that contains 3 tourists (TX, TY and TZ) and 5 POIs corresponding to Top-K = 5. (K is the number of items recommended for each user at different times: T1<T2<T3<T4). The parameters "ID_Tourist", "ID_RS" and "ID_POI" represent, respectively, the identifier of the active tourist, the identifier of the recommendation system in use and the identifier of the recommended POI. "Pred_POI_rating" is the predicted rating using the ID_RS system for the tourist "ID_Tourist" and the POI "ID_POI". "AVG_Pred_rating" is the average of the K predicted ratings by an RS for a given tourist and "AVG_POI_Rating" is the average of the ratings given by all tourists at a given POI. "Selected_POI" is a Boolean field that takes the value "true" when the POI with ID_POI is selected (visited) by the tourist identified by ID_tourist and the value "false" otherwise. "Tourist_POI_Rating" is the real rating value given by the tourist ID_tourist to ID_POI. The same tourist can give different ratings to the same POI several times. For example, if tourist "TY" gave a rating of "1" to POI 02 at time T2 and gave a rating of "3" to the same POI at time T4. In this example, aggregating these different ratings for the same POI is necessary to facilitate the pre-filtering operations of this data set. Several solutions can be used to solve this type of problem, such as:

1) Maximum rating (max rating = 3),
2) Minimum rating (min rating =1),
3) Average rating (average rating =2),
4) Fresh rating (fresh rating = 3),
5) Oldest rating (oldest rating = 1),
6) weighted rating (weighted rating if w1 = 0,4 and w2 = 0,6 the rating will be 1*0.4+3*0.6 = 2.2).

In our work, the most recent rating (fresh rating) was chosen, because, in general the most recent evaluations of POIs can better help future tourists in their visits. In the following, the calculation of the parameters CTR, MAE, RMSE, precision and recall are explained in Table III.

TABLE III. CALCULATION EXAMPLE OF ONLINE/OFFLINE PARAMETERS PER TOURIST AND RS

Time	ID_Tourist	ID_RS	ID_POI	Pred_POI_Rating	AVG_Pred_rating	Selected_POI	Tourist _POI_Rating	AVG_POI_Rating	CTRt	CTRRS	MAEt	MAERS	RMSEt	RMSERS	Recommended POI	Relevant POI	Precisiont	Recallt	F1t	PrecisionRS	RecallRS	F1RS
T1	Tourist_TX	RS1	POI21	1,7		No	-	2							1							
			POI37	0,8		No	-	0,7								1						
			POI 03	**2,2**	1,66	Yes	**3**	2,2	1/5	**0,2**	0,8	**0,8**	0,8	**0,8**	1	1	1/3	1/2	**0,4**	**1/3**	**1/2**	**0,4**
			POI 09	1		No	-	2,3														
			POI 05	2,6		No	-	4							1							
		RS2	POI 04	1		No	-	2														
			POI 02	0,5		No	-	2,6														
			POI 03	**2,1**	2,1	Yes	**3**	2,2	1/5	**0,2**	0,9	**0,9**	0,9	**0,9**	1		2/3	2/2	**0,8**	**2/3**	**2/2**	**0,8**
			POI 44	2,9		No	-	2,3							1	1						
			POI 55	4		No	-	4							1	1						
T2	Tourist_TY	RS1	POI 11	2,4		No	-	2							1	1						
			POI 02	**0,9**		**Yes**	**1**	2,6	1/5		0,1		0,1									
			POI 03	**2,3**	1,88	No	-	**2,6**		**0,2**		**0,45**		**0,57**	1		1/3	1/1	**0,5**	**2/6**	**3/4**	**0,46**
			POI 04	1,1		No	-	2,3														
			POI 05	2,7		No	-	4							1							
		RS2	POI 21	3,9		No	-	2							1	1						
			POI 02	**2,7**		**Yes**	**1**	2,6	1/5		1,7		1,7			1						
			POI 03	**2,5**	2,82	No	-	**2,6**		**0,2**		**1,3**		**1,36**			2/2	2/3	**0,8**	**5/6**	**5/6**	**1**
			POI 24	2,9		No	-	2,3							1	1						
			POI 15	2,1		No	-	4														
T3	Tourist_TZ	RS1	POI 11	2,4		No	-	2			-		-		1	1						
			POI 02	2,7		No	-	**1,8**			-		-		1	1						
			POI 03	2,6	2,08	No	-	**2,6**		**0,2**	-	**0,5**	-	**0,58**	1	1	3/3	3/3	**1**	**5/9**	**5/6**	**0,67**
			POI 14	1,4		Yes	**2**	2,3	1/5		0,6		0,6									
			POI 05	1,3		No	-	4			-		-									
		RS2	POI 21	2,2		No	-	2			-		-		1	1						
			POI 02	2,4		No	-	**1,8**			-		-		1	1						
			POI 03	1,7	2,18	No	-	**2,6**	-	**0,13**	-	**1,3**	-	**1,36**			3/3	3/3	**1**	**8/9**	**8/9**	**1**
			POI 22	2,7		No	-	2,3			-		-		1	1						
			POI 05	1,9		No	-	4			-		-									
T4	Tourist_TY	RS1	POI 21	2,6		No	-	2							1	1						
			POI 22	2,6		No	-	1,6							1	1						
			POI 03	2,5	2,08	No	-	**2,6**		**0,2**		**0,8**		**0,99**	1		2/3	2/2	**0,8**	**7/12**	**7/8**	0,69
			POI 04	1,4		No	-	2,3														
			POI 11	1,3		Yes	**3**	**1,8**	0,2		1,7		1,7									
		RS2	POI 01	2,2		No	-	2							1	1						
			POI 23	2,4		No	-	**1,8**							1	1						
			POI 03	1,7	2,18	No	-	**2,6**		**0,15**		**1,23**		**1,31**			3/3	3/3	**1**	**11/12**	**11/12**	**1**
			POI 04	2,7		No	-	2,3							1	1						
			POI 11	1,9		Yes	**3**	4	0,2		1,1		1,1									

1) CTR parameter Computation

The CTR parameter evaluates the ratio acceptance of the POI suggestions provided by each RS (RS1 and RS2) for a given tourist. In the following, this parameter is calculated from Table III using Eq.(6):

$$CTR_{TXRS1T1} = \frac{card\{poi_{03}\}}{card\{poi_{21}, poi_{37}, poi_{03}, poi_{09}, poi_{05}\}} = \frac{1}{5}$$

In order to compare the acceptance ratios of RS1 and RS2, the parameter CTR_{RS} is computed for RS1 using Eq.(7) as follows:

$$CTR_{RS1} = \frac{SumCTR_{RS1}}{4} = 0,2$$

Where: $SumCTR_{RS1} = CTR_{TX-RS1-T1} + CTR_{TY-RS1-T2} + CTR_{TZ-RS1-T3} + CTR_{TY-RS1-T4}$

In this example, RS1 has achieved a higher acceptance ratio than RS2 (see Table III) because $\text{CTR}_{RS1} > CTR_{RS2}$.

2) MAE parameter computation

The MAE parameter evaluates the difference between the tourist's rating and the rating predicted by the RS. In the following, this parameter is calculated from Table III using Eq.(9): $MAE_{TX-RS1-T1} = |3 - 2,2| = 0,8$ Then MAE_{RS1} is given by Eq.(10) as follows:

$$MAE_{RS1} = \frac{SumMAE_{RS1}}{4} = 0,8$$

Where $SumMAE_{RS1} = (MAE_{TX-RS1-T1} + MAE_{TY-RS1-T2} + MAE_{TZ-RS1-T3} + MAE_{TY-RS1-T4})$.

In this example, RS_1 records a smaller MAE than RS_2, so the ratings predicted by RS1 are closer to the ratings provided by tourists.

3) RMSE parameter computation

The RMSE parameter also allows us to evaluate the difference between the tourist's rating and the rating predicted by the RS.

In the following, this parameter is computed from Table III using Eq.(12) : $RMSE_{TX-RS1-T1} = \sqrt{(3-2,2)^2} = 0,8$

RMSEs per RS is deduced from Eq.(13) as follows: $RMSE_{RS1} = \sqrt{\frac{SumRMSE_{RS1}}{4}} = 0,99$

Where $SumRMSE_{RS1} = (RMSE^2_{(TX-RS1-T1)} + RMSE^2_{(TY-RS1-T2)} + RMSE^2_{(TZ-RS1-T3)} + RMSE^2_{(TY-RS1-T4)})$

In this example, RS_1 has a smaller RMSE than RS_2, so the ratings predicted by RS1 are closer to the ratings provided by tourists, which confirms the result obtained by the MAE calculation for RS_1 and RS_2.

4) Precision and recall calculation

In our previous work [51] [52], we assumed that the recommended POIs are determined by the condition "Pred_POI_rating >= 1" and the relevant POIs are defined through the condition "Pred_POI_rating >= 2.5". These two conditions and the Top-K, which does not exceed 10 at most, can reduce the possible choices for a tourist, especially on a tourist map where the visibility of the POIs strongly depends on the reduced size of the smartphone screen. To solve this problem, we defined the POIs to be recommended using the condition "Pred_POI_rating >= AVG_Pred_rating" and the relevant POIs using the condition "Pred_POI_rating >= AVG_POI_Rating". In the following, precision, recall per tourist, and RS are computed from Table III using Eq.(14), and Eq.(16) respectively.

$Precision_{(TX-RS1-T1)} = \frac{Card\{POI_{03}\}}{Card\{POI_{21},POI_{03},POI_{05}\}} = \frac{1}{3} = 0,33$

$Recall_{(TX-RS1-T1)} = \frac{Card\{POI_{03}\}}{Card\{POI_{37},POI_{03}\}} = \frac{1}{2} = 0,5$

Having calculated the precision and recall parameters per tourist, the precision and recall parameters per RS are deduced using Eq.(15) and Eq.(17) as shown below:

$$Precision_{(RS1)} = \frac{SumPrecision_RS1}{4} = \frac{7}{12} = 0,58$$

Where $SumPrecision_{(RS1)} = (Precision_{(TX-RS1-T1)} + Precision_{(TY-RS1-T2)} + Precision_{(TZ-RS1-T3)} + Precision_{(TY-RS1-T4)})$.

$$Recall_{(RS1)} = \frac{SumRecall_RS1}{4} = \frac{7}{8} = 0,875$$

Where $SumRecall_{(RS1)} = (Recall_{(TX-RS1-T1)} + Recall_{(TY-RS1-T2)} + Recall_{(TZ-RS1-T3)} + Recall_{(TY-RS1-T4)})$.

5) F1 parameter calculation

The F1 factor is a consolidation of recall and precision into a single value, thus balancing the values associated with these two parameters [53]. In the following, this parameter is computed from Table III using Eq.(19):

$$F1_{(TX-RS1-T1)} = \frac{2 \cdot (Precision_{(TX-RS1-T1)} \cdot Recall_{(TX-RS1-T1)})}{(Precision_{(TX-RS1-T1)} + Recall_{(TX-RS1-T1)})} = 0,4$$

After calculating the F1 parameter for each tourist above, the F1 parameter for each RS is derived using Eq.(20), as shown below:

$$F1_{RS1} = \frac{2 \cdot (Precision_{RS1} \cdot Recall_{RS1})}{(Precision_{RS1} + Recall_{RS1})} = 0,69$$

6. RESULT ANALYSIS AND DISCUSSION

In Section 5, we have explained the experimental process and the performance calculation metrics using a concrete example. This theoretical framework is used to implement our SEPRA approach, which can compare RSs and CARSs deduced from the two RAs and contextual information such as "arrival time" and "weather of the day". This comparison is carried out online to estimate (1) the degree of tourist satisfaction using the CTR parameter and (2) the quality of the recommendations using precision, recall, and F1 parameters. In addition, after the end of the test phase, our approach also allows us to compare the accuracy of rating predictions made by the RBBP and RPBT algorithms via the MAE and RMSE metrics. In this section, the experimental phase of our new tool for smart tourism by tourists is presented using a real case study. For this reason, we focused on studying the reaction of our prototype to classical RS/CARS problems such as cold start, data sparsity and scalability. Then, we studied (1) The efficiency of algorithms used by our tool and (2) the quality of the POI recommendations from tourists' point of view. Finally, the results of the online and offline evaluation use of our prototype are presented in the following.

A. Case study

Our SEPRA approach is used to compare two RAs (RBBP and RPBT) through the case study of a website for discovering tourist sites in the department of Chlef/Algeria. This case study concerns nomadic tourists using this new smart tourism tool during the test phase and afterwards. Details of this study are given below in Table IV.

TABLE IV. THE PARAMETERS USED FOR OUR CASE STUDY

Number of tourists	Number of POIs	Number of requests	Duration of experimentation phase
30	51	133	3 months

The class diagram shown in Figure 3 describes our dataset's structure and can be implemented using SQL. Our data set comprises three tables: POI, USER and Feedback. These tables are described as follows: Table V illustrates the POI table, which provides POIs information.

TABLE V. POI TABLE STRUCTURE

Attribute	Description
Id	The POI identifier
Designation	POI designation
Type	POI type
Description	POI description
Latitude	POI location by latitude
Longitude	POI location by longitude
Views	POI views count
MoyRatePoi	POI Average rating

Table VI illustrates the USER table, which provides tourist information.

TABLE VI. USER TABLE STRUCTURE

Attribute	Description
Id	The user id
Lname	The last name of the user
Fname	The first name of the user
Password	User password
Email	User email
Phone	User phone
Last_seen	User last seen date

Table VII illustrates the Feedback table, which provides tourist ratings for POIs according to the time or weather contextual factors.

TABLE VII. FEEDBACK TABLE STRUCTURE

Attribute	Description
Id	Feedback id
Rate	Rate given by tourist for a POI
Comment	Comment on POI
Date_feedback	Feedback date
Weather	POI Weather during feedback
Temperature	Temperature during feedback
Longitude_feedback	User longitude during feedback
Latitude_feedback	User latitude during feedback
User_id	User Id
Poi_id	POI Id

1) Comparing the RPBT and RPBP algorithms during the test phase

In this sub-section, the RMSE parameter is used to compare the ratings predicted by the RBPT algorithm with those predicted by the RBBP algorithm (see Figure 5) to find out which algorithm could best estimate future tourist ratings. Next, the acceptance ratios of the POI recommendations (CTR parameter) of the RBPT algorithm and those of the RBBP algorithm (see Figure 6) are calculated. From this, the algorithm that most predicted POIs followed by tourists can be deduced. Finally, we looked at the quality of the POI recommendations (F1 parameter) provided by the RPBT and RPBP algorithms (see Figure 7) in order to determine which algorithm is closer to tourists' preferences.

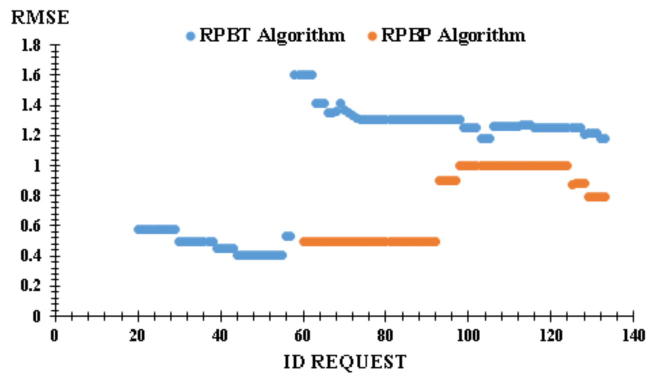

Figure 5. Rating prediction errors (RMSE) comparison for RPBT and RPBP algorithms.

These three comparisons are executed during the course of tourists' visits. Figure 5 shows that from query 19 the RPBP algorithm provides predicted ratings closer to tourists' ratings than the ratings provided by the RPBT algorithm.

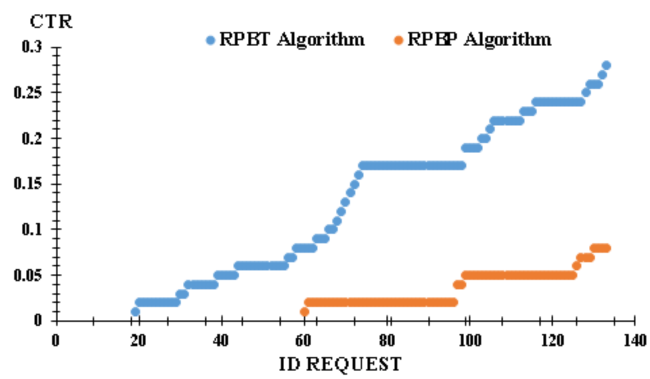

Figure 6. Acceptance rates (CTR) comparison for RPBT and RPBP algorithms.

On the other hand, Figure 6 shows that from query 19, the RPBT algorithm is better than the RPBP algorithm and has a higher acceptance ratio than the RPBP algorithm. Figure 7 also states that from query 19 the RPBT algorithm is better than the RPBP algorithm and achieves recommendations with the highest F1.

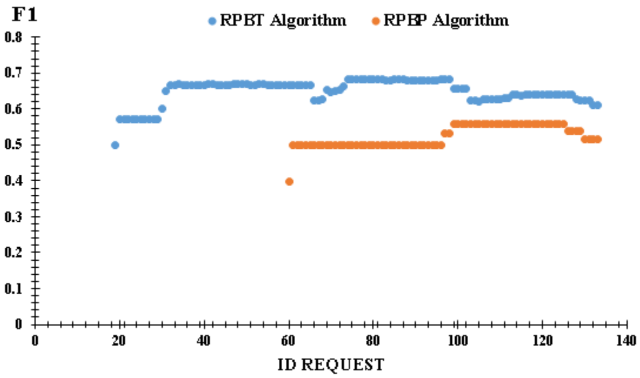

Figure 7. Recommendation qualities (F1) comparison for RPBT and RPBP algorithms.

2) Comparison of the RPBT and RPBP algorithms after the test phase

In this sub-section, we present the synthesis of the offline calculations of the parameters CTR, MAE/RMSE, Precision/Recall and F1 related to the recommendation of POIs by the algorithms RPBT and RPBP. This synthesis also concerns the behaviour of these two algorithms, taking into account the context dimensions of time and weather for tourists.

Table VIII summarizes the offline evaluation results of our two RAs (RPBT and RPBP) implemented in our prototype after their use by 30 tourists, which generated 133 queries. These queries were used to calculate seven parameters: (1) cold start time, (2) MAE, (3) RMSE, (4) CTR, (5) precision, (6) recall and (7) F1.

During the beginning of our test phase (from query 1 to query 59), we encountered the cold start problem, which is closely related to data sparsity. However, we found that the RPBT algorithm started to produce recommendations (cold start) as early as the twentieth query, while the RPBP algorithm remained blocked until the fiftieth query, as shown in Table VIII. Finally, we estimate that the RPBT algorithm had a better acceptance ratio than the RPBP algorithm for the 133 queries made by the 30 tourists.

B. Discussion

The advantages of our approach are summarized as follows:

1) Our primary contribution lies in introducing a novel approach to RS evaluation termed 'systemic evaluation.' This method seamlessly combines both online and offline evaluation, providing simultaneous and instantaneous insights. Implemented as the SEPRA tool, this innovation enables real-time monitoring of RSs performance.
2) Few datasets offer contextual information for tourism, particularly for online evaluation, and none can be used explicitly for Algerian tourism domain. Thus, we gathered contextualized data tailored to Algerian tourism sector. This initiative addressed the challenge of cold start-up, enabling real-time monitoring of Recommender System (RS) performance within this context.
3) In our approach, we integrate the context pre-filtering, which makes the calculations lighter and faster.
4) Our integration of context employs the relaxation principle, enabling us to address the sparsity problem by considering one dimension of the context at a time.
5) Our approach facilitates the concurrent operation of multiple RSs, merging their recommendations to mitigate the cold start issue linked to data sparsity.

While our approach offers several advantages, it also has some limitations:

1) Even with the merging of results from multiple RSs, the cold start issue arises upon the addition of a new POI. To tackle this, we are exploring the incorporation of a content-based RS specifically aimed at resolving the cold start problem following the addition of a new POI.
2) At present, our tool relies on memory-based collaborative filtering for recommendations. Among the challenges encountered is the issue of sparsity, which could be addressed by integrating model-based Recommender Systems (RSs).
3) While our online evaluation proposal facilitates the monitoring of RS performance, it strains the server in terms of computation and storage. To address this, we propose implementing a more suitable architecture.

7. CONCLUSIONS AND PERSPECTIVES

In this paper, we have designed and implemented a smart tourism tool enabling parallel execution of multiple RAs (e.g., RPBT and RPBP algorithms) for each tourist. We also implemented an approach called SEPRA to compare these two RAs by using parameters such as MAE/RMSE, precision/recall/F1 and CTR during the test phase (online evaluation) and after the end of this phase (offline evaluation). These evaluation parameters were used (1) to assess the effectiveness of RAs in recommending places to visit, (2) to estimate the quality of the recommendations provided to tourists during the test phase of our prototype and (3) to get an idea of the tourist's satisfaction degree. Our approach SEPRA enables RS designers to monitor the evolution of their algorithms over time, rather than just having a global performance report for a given date. In fact, the results obtained by our SEPRA approach can consolidate our choice of RA to adopt during the marketing phase of the smart tourism tool because SEPRA enable us to

TABLE VIII. OFFLINE EVALUATION OF THE 133 REQUESTS RELATED TO THE 30 TOURISTS IN OUR PROTOTYPE

RS Algorithms \ parameter		Cold Start end	MAE	RMSE	CTR	Precision	Recall	F1
RPBT (Tourist Based)	Without context	19	0.71	1.18	0,28	0.70	0.55	0.62
	With time context	19	0.43	0.74	0,22	0.58	0.54	0.56
	With weather context	19	0.44	0.78	0,24	0.67	0.55	0.60
RPBP (POI based)	Without context	60	0.43	0.79	0,08	0.50	0.53	0.51
	With time context	62	0.12	0.30	0,06	0.40	0.50	0.44
	With weather context	60	0.70	1.03	0,07	0.70	0.58	0.63

detect (1) phases of profitability, (2) phases of declining performance, (3) phases of stagnation and (4) phases of decline of the RAs. Our approach is in its early stage and needs improvement in terms of calculation time and support for other RA evaluation metrics. On the other hand, we are confident that with a few adaptations, our smart tourism prototype will be more relevant by integrating the friendship relationships between tourists using social networks. Finally, after making several evaluations of the RAs deployed by our smart tourism tool using our SEPRA approach, we have put online a survey aimed at gathering feedback from tourists after their use of our smart tourism tool. According to the initial results of our survey, our tool (which is still in the testing phase) seems to be very useful for the majority of interviewed tourists, who believe that more advanced use of each feature can better support tourism activities in the long term.

ACKNOWLEDGMENT

This research was conducted within the framework of the PRFU project (Grant: COOLO7UN020120220001) of the Department of Computer Science, Hassiba Ben Bouali University, Chlef (Algeria), funded by the Ministry of Higher Education and Scientific Research.

References

[1] Y. Mukasa and K. Yamamoto, "A sightseeing spot recommendation system for urban smart tourism based on users' priority conditions," *J. Civ. Eng. Archit*, vol. 13, pp. 622–640, 2019.

[2] M. HadjHENNI, N. DENNOUNI, and Z. SLAMA, "Sig et la recommandation contextuelle des poi," *GIS DAY*, vol. 1, pp. 39–47, 2019.

[3] M. H. Henni, N. Dennouni, Z. Slama, S. Medjroud, and D. Bettache, "Towards an approach for online evaluation of new variants of content-based poi recommender systems by mobile tourists," in *2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA)*. IEEE, 2022, pp. 89–94.

[4] N. Koubai and F. Bouyakoub, "Myrestaurant: A smart restaurant with a recommendation system," *International Journal of Computing and Digital Systems*, vol. 8, no. 02, pp. 143–156, 2019.

[5] H.-C. Hsieh, Y.-C. Chen, and H.-C. Lin, "More precise: Stores recommendation under o2o commerce," *International Journal of Computing and Digital Systems*, vol. 3, no. 02, 2014.

[6] Z. A. Baizal, D. Tarwidi, B. Wijaya *et al.*, "Tourism destination recommendation using ontology-based conversational recommender system," *International Journal of Computing and Digital Systems*, vol. 10, 2021.

[7] M. Sharma and A. Dhamija, "A novel approach towards movie recommender system using deep learning techniques," *International Journal of Computing and Digital System*, 2021.

[8] L. Ahmedi, E. Rexhepi, and E. Bytyçi, "Using association rule mining to enrich user profiles with research paper recommendation," *International Journal Of Computing and Digital System*, 2021.

[9] D. PM, R. PB, N. Cletus, and P. Joy, "Fuzzy hypergraph modeling, analysis and prediction of crimes," *International Journal of Computing and Digital Systems*, vol. 11, no. 1, pp. 649–661, 2022.

[10] I. Burman, S. Som, and S. A. Hossain, "Model for analyzing psychological parameters recommending student learning behavior," *International Journal of Computing and Digital Systems*, vol. 10, pp. 2–17, 2020.

[11] S. Medjroud, N. Dennouni, M. H. Henni, and D. Bettache, "Towards a new poi recommendation approach based on implicit trust between users," in *2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA)*. IEEE, 2022, pp. 19–24.

[12] N. Dennouni, Y. Peter, L. Lancieri, and Z. Slama, "Vers une approche bio-inspirée pour l'orchestration pédagogique des activités mobiles d'apprentissage," in *7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2015)*, 2015, pp. 408–410.

[13] J. Beel, M. Genzmehr, S. Langer, A. Nürnberger, and B. Gipp, "A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation," in *Proceedings of the international workshop on reproducibility and replication in recommender systems evaluation*, 2013, pp. 7–14.

[14] E. R. Núñez-Valdéz, J. M. C. Lovelle, O. S. Martínez, V. García-Díaz, P. O. De Pablos, and C. E. M. Marín, "Implicit feedback techniques on recommender systems applied to electronic books," *Computers in Human Behavior*, vol. 28, no. 4, pp. 1186–1193, 2012.

[15] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, vol. 463, no. 1999.

[16] W. Yao, J. He, G. Huang, J. Cao, and Y. Zhang, "A graph-based model for context-aware recommendation using implicit feedback data," *World wide web*, vol. 18, pp. 1351–1371, 2015.

[17] W. Yuan, D. Guan, Y.-K. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 232–238, 2010.

[18] L. M. López-López, J. J. Castro-Schez, D. Vallejo-Fernandez, and J. Albusac, "A recommender system based on a machine learning algorithm for b2c portals," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1. IEEE, 2009, pp. 524–531.

[19] S. Kulkarni and S. F. Rodd, "Context aware recommendation systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, p. 100255, 2020.

[20] A. Chen, "Context-aware collaborative filtering system: Predicting the user's preference in the ubiquitous computing environment," in *Location-and Context-Awareness: First International Workshop, LoCA 2005, Oberpfaffenhofen, Germany, May 12-13, 2005. Proceedings 1*. Springer, 2005, pp. 244–253.

[21] S. Valencia Rodríguez and H. L. Viktor, "A personalized location aware multi-criteria recommender system based on context-aware user preference models," in *Artificial Intelligence Applications and Innovations: 9th IFIP WG 12.5 International Conference, AIAI 2013, Paphos, Cyprus, September 30–October 2, 2013, Proceedings 9*. Springer, 2013, pp. 30–39.

[22] H. Bagci and P. Karagoz, "Random walk based context-aware activity recommendation for location based social networks," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, pp. 1–9.

[23] M. Gan and L. Gao, "Discovering memory-based preferences for poi recommendation in location-based social networks," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 279, 2019.

[24] M. Tenemaza, S. Lujan-Mora, A. De Antonio, and J. Ramirez, "Improving itinerary recommendations for tourists through metaheuristic algorithms: an optimization proposal," *IEEE Access*, vol. 8, pp. 79 003–79 023, 2020.

[25] Y. M. Arif, H. Nurhayati, S. M. S. Nugroho, and M. Hariadi, "Destinations ratings based multi-criteria recommender system for indonesian halal tourism game," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 1, pp. 282–294, 2022.

[26] L. Wan, H. Wang, Y. Hong, R. Li, W. Chen, and Z. Huang, "itourspot: a context-aware framework for next poi recommendation in location-based social networks," *International Journal of Digital Earth*, vol. 15, no. 1, pp. 1614–1636, 2022.

[27] X. Liu, J. Guo, and P. Qiao, "A context awareness hierarchical attention network for next poi recommendation in iot environment," *Electronics*, vol. 11, no. 23, p. 3977, 2022.

[28] F. F. Laskoski and A. Goldman, "Cientec guide: Application and online evaluation of a context-based recommender system in cultural heritage," in *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação*. SBC, 2022, pp. 310–317.

[29] Y. Zheng, R. Burke, and B. Mobasher, "Differential context relaxation for context-aware travel recommendation," in *E-Commerce and Web Technologies: 13th International Conference, EC-Web 2012, Vienna, Austria, September 4-5, 2012. Proceedings 13*. Springer, 2012, pp. 88–99.

[30] L. Peska and P. Vojtas, "Off-line vs. on-line evaluation of recommender systems in small e-commerce," in *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, 2020, pp. 291–300.

[31] J. Nayek and R. Das, "Evaluation of famous recommender systems: a comparative analysis," *Lib Philos Pract (e-journal)*, vol. 5338, 2021.

[32] E. Zangerle and C. Bauer, "Evaluating recommender systems: survey and framework," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022.

[33] P. Castells and A. Moffat, "Offline recommender system evaluation: Challenges and new directions," *AI Magazine*, vol. 43, no. 2, pp. 225–238, 2022.

[34] A. Sun, "Take a fresh look at recommender systems from an evaluation standpoint," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 2629–2638.

[35] P. Cremonesi, F. Garzotto, and R. Turrin, "Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, no. 2, pp. 1–41, 2012.

[36] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, pp. 305–338, 2016.

[37] M. K. M. McGill and N. T., "An evaluation of factors affecting document ranking by information retrieval systems." p. 144, 1979. [Online]. Available: https://eric.ed.gov/?id=ED188587

[38] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web: methods and strategies of web personalization*. Springer, 2007, pp. 291–324.

[39] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Noise reduction in speech processing*. Springer Science & Business Media, 2009, vol. 2.

[40] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.

[41] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[42] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*. Springer, 2010, pp. 217–253.

[43] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. C. Pierewan, H. Chiroma, and T. Herawan, "Context-aware recommender system: A review of recent developmental process and future research direction," *Applied Sciences*, vol. 7, no. 12, p. 1211, 2017.

[44] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[45] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *arXiv preprint arXiv:1301.7363*, 2013.

[46] J. Bennett, S. Lanning *et al.*, “The netflix prize,” in *Proceedings of KDD cup and workshop*, vol. 2007. New York, 2007, p. 35.

[47] D. Billsus, M. J. Pazzani *et al.*, “Learning collaborative information filters.” in *Icml*, vol. 98, 1998, pp. 46–54.

[48] B. Sarwar, G. Karypis, J. Konstan, B. Riedl, Joh Sarwar, G. Karypis, J. Konstan, and J. T. Riedl, “Application of dimensionality reduction in recommender system-a case study,” in *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.

[49] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Analysis of recommendation algorithms for e-commerce,” in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000, pp. 158–167.

[50] T. Di Noia, I. Cantador, and V. C. Ostuni, “Linked open data-enabled recommender systems: Eswc 2014 challenge on book recommendation,” in *Semantic Web Evaluation Challenge*. Springer, 2014, pp. 129–143.

[51] N. DENNOUNI, Y. PETER, L. LANCIERI, and Z. SLAMA, “Towards a geographical orchestration of mobile learning activities.” *International Journal of Interactive Mobile Technologies*, vol. 8, no. 2, 2014.

[52] N. Dennouni, “Orchestration des activités d’apprentissage mobile,” Theses, Université de Lille 1 - Sciences et Technologies, Jan. 2016. [Online]. Available: https://hal.science/tel-01742212

[53] T. Kanstrén, “A look at precision, recall, and f1-score: exploring the relations between machine learning metrics,” 2020, [Online; accessed April 08, 2023]. [Online]. Available: https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec

Mhamed Hadj Henni is a PhD student in the computer science department at Sidi Bel-Abbes University (Algeria). He earned a Master’s (Magister) degree in computer science in 2009 from Ecole nationale Supérieure d’Informatique (ESI) Oued Semar Algeria. His primary research domains are artificial intelligence, ubiquitous computing, orchestration of activities, and POI recommendation. He is a member of the ISIBA team of EEDIS laboratory at the computer science Department of Djillali Liabes University of Sidi Bel Abbes and he is also a member of the ICAR team of the LIA laboratory at the Hassiba BENBOUALI University of Chlef.

Nassim DENNOUNI is an associate professor at the Higher School of Management at Tlemcen since October 2022. He is the ICAR team leader of the LIA Laboratory of the Hassiba BENBOUALI University of Chlef (Algeria) and he is also a member of the ISIBA team of EEDIS laboratory of Computer Science Department of Djillali Liabes University of Sidi Bel Abbes (Algeria). In 2018, he obtained his university accreditation from Djillali Liabes University of Sidi Bel Abbes. In 2016, he received a PhD in computer science from Djillali Liabes University of Sidi Bel Abbes. His primary research domains are artificial intelligence, ubiquitous computing, mobile learning, orchestration of activities, and POI recommendation.

Zohra SLAMA is an associate professor in computer science and she is the ISIBA team leader of the EEDIS laboratory at the Djilali Liabes University of Sidi Bel Abbes (Algeria). She received her PhD from Djillali Liabes University of Sidi Bel Abbes. Her major research area is paradigms of artificial intelligence, mobile learning, orchestration of activities, POI recommendation, mobile computing, and advanced databases.