International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

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

Towards a systematic point-of-interest recommendations based on trust between users deduced from their ratings and check-ins in a LBSN

Sara Medjroud[1], Nassim Dennouni[2,3] and Mourad Loukam[1,2]

[1] *Computer Science Department,Hassiba BENBOUALI University, Chlef, Algeria*
[2] *LIA Laboratory,Hassiba BENBOUALI University, Chlef, Algeria*
[3] *Higher School of Management, Tlemcen, Algeria*

Received Mon. 20, Revised Mon. 20, Accepted Mon. 20, Published Mon. 20

Abstract: Nowadays, location-based social networks (LBSNs) have advanced to facilitate users in quickly sharing their check-ins and rating points of interest (POIs), aiding in better targeting the preferences of future users. However, the variety of POIs and the increasing interactions of LBSN users with these locations diminish the accuracy of memory-based recommendation methods (including collaborative filtering, context integration, etc.), especially when dealing with new users. Furthermore, these methods also demonstrate limited effectiveness, even when employing models such as matrix factorization, deep learning, etc., primarily due to the rapid evolution of the history of LBSNs. To tackle these challenges, this article introduces a POI recommendation system (RS) that relies on implicit trust among users within an LBSN. The system aims to (1) enhance the accuracy of recommendation methods through collaborative filtering and (2) provide an alternative to explicit trust models that involve active user participation. This RS utilizes the HRCT (Hybrid Rating Check-in Trust) model to deduce implicit trust from POI ratings and user check-ins, employing three types of trust matrices: the TDMR (Trust Derivation Matrix based on Rating), the TDMC matrix (Trust Derivation Matrix based on Check-in) and the H-Trust matrix combining these two matrices. The preliminary experimental results obtained with this model reveal that its algorithms achieve better performance in terms of RMSE and Precision/Recall compared to collaborative filtering techniques using Pearson, Cosine and Jaccard similarities. Moreover, this model can effectively address the data sparsity challenge of user/user similarity matrices by enhancing the density of the model's trust matrix derived from the H-Trust algorithm.
Keywords: LBSN, recommender system, collaborative filtering, implicit trust, HRCT, POI, check-in, rating, RMSE, Precision/Recall

1. INTRODUCTION

Today, the development of internet infrastructure and the introduction of GPS-enabled smartphones has reduced the boundary between the real and virtual worlds, generating significant interest in LBSNs that merge location services with online social networks [1]. Platforms like Yelp [2], Gowalla [3], Foursquare [4] and Brightkite [5] allow users to save and share their favorite points of interest (POIs), thus contributing to better understanding their preferences and behaviors [6]. However, the challenges posed by the increasing volume of information generated by these platforms and the proliferation of POIs make conventional recommendation methods (which rely on user/user and POI/POI similarities) imprecise, as they do not consistently meet the expectations of their users [7]. On the other hand, model-based POI recommendation methods may not rapidly adjust to changes in user preferences or the frequent emergence of new POIs [8].

To address these issues, POI RSs based on trust (explicit or implicit) among users can offer a promising solution to help future users discover new, relevant and interesting places during their visits [9].

POI recommendations based on explicit trust [10] depend on users' direct evaluations and ratings of each other. In this framework, users manually assign trust scores to their peers, indicating their level of confidence in each other's recommendations [11]. While this approach enables for precise reliability estimation, it requires active user participation to provide these evaluations [12] [13].

Conversely, recommendations based on implicit trust derive trust levels from users' past behavior and interactions. The system examines user data, such as POI ratings, check-ins, etc., to deduce similarities between users and identify implicit trust relationships. This approach does not demand extra effort from users, but it may be less accurate than explicit trust [14].

E-mail address: s.medjroud@univ-chlef.dz, n.dennouni@univ-chlef.dz, m.loukam@univ-chlef.dz

Therefore, recommendations based on explicit trust can be utilized when direct feedback is available, thus providing more reliable recommendations. However, when explicit data is limited or unavailable, RSs can analyze users' behaviors (check-ins) and preferences (ratings) to recommend potentially interesting POIs [15] [16]. Thus, POI RSs can offer relevant and reliable suggestions, even without direct evaluations from users [17] [18].

In the context of LBSNs, POI ratings/check-ins are generally more available than explicit user trust scores. Indeed, LBSNs such as Foursquare, Yelp, etc., encourage users to leave reviews (comments), ratings and recommendations (check-ins) on the places they visit. These POI ratings are a crucial component of user-generated content and are widely accessible. However, explicit trust scores between users are less commonly found in LBSNs. Although some platforms allow users to friend or follow other users (explicitly assigning trust scores is not a standard feature) [19] [20]. As a result, a substantial amount of work on RSs for LBSNs is dedicated to inferring implicit trust from POI ratings, user preferences, and user behaviors, rather than relying solely on explicit trust scores. In this article, we propose a hybrid system [21] [22] that utilizes a similarity measure between users based on implicit trust inferred from POI ratings and check-ins exhibited by these users. This system facilitates the calculation of rating predictions essential for recommending POIs.

The rest of the document is organized as follows. Section 2 presents a state-of-the-art overview of current POI recommendation approaches, primarily focusing on implicit trust between users of an LBSN and exploring the motivations for this research. Then, in section 3, the mathematical formulas and algorithms for deducing the implicit trust between users from their check-ins and POI ratings are detailed. Section 4 describes the design of the model called HRCT (Hybrid Rating Check-in Trust) and explains the operation of its main components. Before the conclusion, sections 5 and 6 analyze and discuss the experimental results, comparing the HRCT model with other existing models. Finally, the last section summarizes the article's contributions and offers future perspectives.

2. LITERATURE REVIEW

POI recommendation in LBSNs is a powerful technology that assists users in discovering new and relevant places they have never visited before. However, two main trust-based approaches exist in the literature to generate this type of recommendation: recommendations based on explicit trust and those based on implicit trust [23].

Currently, the POI recommendation approach based on implicit trust is more commonly used in the context of LBSNs because users of this network type are often hesitant to explicitly provide trust evaluations towards other users. Therefore, explicit trust data is sparse and limited [24] [25].

Moreover, LBSNs generate large amounts of behavioral data, such as check-ins, POI ratings, travel histories, etc. this data can be leveraged to derive implicit trust scores between users [26].

In the same context, Zhu et al. introduced another algorithm with the aim of identifying trust clusters to utilize them in a trust prediction method. These trust values were then combined with the similarities between individuals to recommend friends to the target user. Subsequently, these authors developed a hybrid framework integrating user preferences, geographic influence and trust relationships to recommend POIs [27].

Furthermore, Wang et al. [28] discovered that if two users visit the same location within a given time frame, trust links between them can be inferred. This technique, based on the co-visiting of places by users, enables the recommendation of POIs while considering geographical and temporal influences.

On the other hand, Ekaterina et al. emphasize that older reviews (written several years ago) are generally less informative than recent ones. Consequently, trust in the authors' reviews for recommending POIs varies depending on the dates of their reviews [29].

In the literature, several works are based on collaborative filtering memory based techniques like those cited above. However, certain studies choose to utilize models such as matrix factorization instead.

Logesh and Subramaniyaswamy [30] employ a POI recommendation algorithm called Social Relevant Trust Walker (SPTW), which relies on the levels of trust between users calculated through matrix factorization. This algorithm is an extended version of the work by Jamali and Ester [31], aimed at efficiently recommending locations by integrating user similarities, trust relationships and location categories.

Unlike traditional trust-based approaches, Zhu et al. [32] combine user preferences, social trust-distrust and geographic influence to recommend POIs. This fusion allows for the computation of both trust and distrust scores using a modified normalized Jaccard coefficient, thereby facilitating the integration of distrust links and the examination of their propagations.

In the same context, Xu et al. incorporate multiple factors including preferences, social relationships and spatiotemporal factors into a matrix factorization model to propose a POI recommendation method. This method calculates similarity based on Jaccard root mean square difference (JMSD) to measure direct trust scores and estimates indirect trust values using propagation [33].

On the other hand, to address the data sparsity issue in user-POI rating matrices, An et al. propose a temporal similarity measure and utilize it with another type of matrix factorization model to infer missing user preferences. This model enables the inference of both direct and indirect trust between users, leveraging a POI category factor along with temporal, geographic and textual factors [34].

Previous work falls within the framework of implicit trust deduced from user interactions with location-based social networking platforms. These works either utilize models such as matrix factorization for collaborative filtering or simply employ collaborative filtering techniques using various similarity measures. In this article, a new similarity measure is proposed to estimate the similarity scores between different users of an LBSN during its testing phase. This score is calculated from POI ratings and user check-ins.

The Table I below compares some works in the literature with our approach. Finally, our work enables us to conduct several levels of offline evaluation of POI recommendations using indicators such as RMSE and Precision/Recall, which are calculated according to the progressive evolution of the history of our LBSN. Unlike previous works, our contribution facilitates the testing of a heuristic based on the k nearest neighbors during the construction of trust relationships inferred from the ratings of the POIs and the check-ins of the users of the new LBSN.

3. METHOD

This section explains how to deduce implicit trust between users based on their POI ratings and check-ins data. The formulas used to calculate this trust are detailed, followed by the introduction of two algorithms designed to deduce trust between users and predict ratings based on their interactions (ratings and check-ins) with POIs.

A. Calculate Implicit Trust

O'Donovan and Smith [35] define trust as relying on the reliability of a partner's profile to provide accurate recommendations in the past. For instance, a profile that has consistently made accurate recommendation predictions in the past may be considered more reliable than another profile that has frequently made poor predictions. This type of prediction can be calculated using the formula 1 provided below [36]:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{b=1}^{N} (\mathrm{r_{b,i}} - \overline{r_b})\,\mathrm{sim(a,b)}}{\sum_{b=1}^{N} \mathrm{sim(a,b)}} \tag{1}$$

Where:

- $P_{a,i}$ The predicted rating for user a on item i.
- $\bar{r}_a$ The average ratings of user a for all items.
- $\mathrm{r_{b,i}}$ The actual rating given to item i by user b.
- $\mathrm{sim(a,b)}$ The similarity between user a and user b.
- N The set of neighbors for user a.

However, to calculate the rating prediction of user a for a given item i based on user b considered as the only recommender [35], formula 2 derived from formula 1 can be used [37]:

$$P^b_{a,i} = \overline{r_a} + (r_{b,i} - \overline{r_b}) \tag{2}$$

Where:

- $P^b_{a,i}$ The predicted rating for user a on item i based on user b.
- $\overline{r_a}$ The average ratings of user a for all items.
- $\overline{r_b}$ The average ratings of user b for all items.
- $r_{b,i}$ The actual rating given to item i by user b.

According to O'Donovan and Smith, the prediction of a rating for user a on item i based on recommender b is considered *correct* only if the predicted rating $P^b_{a,i}$ is close to the actual rating given by user a, denoted as $r_{a,i}$, as shown in equation 3.

$$\mathrm{Correct}(i,b,a) \Leftrightarrow \left|P^b_{a,i} - r_{a,i}\right| < \varepsilon \tag{3}$$

Therefore, $Correct(i,b,a)$ takes the value 1 if $\left|P^b_{a,i} - r_{a,i}\right| < \varepsilon$ and 0 otherwise.

Next, O'Donovan and Smith use formula 4 below to define *RecSet(b)* as the complete set of recommendations in which recommender b was involved:

$$\mathrm{RecSet}(b) = \left\{\left(P^b_{1,1}, r_{1,1}\right), \ldots, \left(P^b_{n,m}, r_{n,m}\right)\right\} \tag{4}$$

Where:

- $P^b_{j,k}$ represents the prediction of recommender b for the rating that a user j (where j varies from 1 to n) will give to an item k (where k varies from 1 to m).
- $r_{j,k}$ represents the real rating of item k (where k varies from 1 to m) given by a user j (where j varies from 1 to n).

From *RecSet(b)*, the subset of correct recommendations, denoted as *CorrectSet(b)*, is calculated using formula 5 as shown below [35].

$$\mathrm{CorrectSet}(b) = \left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{RecSet}(b) : \mathrm{Correct}\left(k, b, P^b_{j,k}\right)\right\} \tag{5}$$

Finally, the notion of trust at the profile level, denoted $Trust^P$ for a recommender b, can be defined as the percentage of correct recommendations out of all the recommendations in which this recommender participated, using formula 6 as shown below [35].

$$\mathrm{Trust}^P(b) = \frac{\mathrm{card}\{\mathrm{CorrectSet}(b)\}}{\mathrm{card}\{\mathrm{RecSet}(b)\}} \tag{6}$$

TABLE I. Comparison Grid of Literature Works with Our Approach

Work	Method	Trust	Other Factors	Dataset	Evaluation metric	Prediction	Propagation
[30]	CF model: matrix factorization	Implicit (relationship)	location category	Foursquare, Gowalla, Jiepang, Britekite	RMSE, coverage, F-Measure, Precision, DOA	Rating prediction	No
[32]	CF Memory	Implicit (friend)	user preference, geographic influence and social trust	Foursquare, Gowalla	Precision@N, Recall@N	Check-in probability	No
[27]	CF Memory	Implicit (friend)	user preference, geographical influence, and trust relationship	Foursquare, Instagram	Precision@N, Recall@N	Check-in probability	Yes
[28]	CF Memory	Implicit (users went to the same POI)	User Preference, temporal influence, geographic influence	Foursquare, Gowalla	Precision@k, Recall@k	Check-in probability	No
[29]	CF	Implicit	trust in the review, Relevance	"Tripadvisor" and "Restoclub" services	MAP@k	POI recommendation	No
[33]	CF model: matrix factorization	Implicit	user trust relationship, user preference, check-in time, geographical location	Gowalla, Weeplaces, Yelp	Precision, Recall, F-score, MAE, NDCG	Predicted rating	Yes
[34]	CF model: matrix factorization	Implicit	POI category, temporal, geographical, and textual content factors	Gowalla, Foursquare	Precision@k, Recall@k	Predicted rating and Check-in	Yes
Our work	CF Memory	Implicit	POI ratings, User Check-in	Our LBSN	RMSE, Precision, Recall	Predicted rating	Yes

From formula 6, a more refined trust metric at the item level, denoted $Trust^I$, can be defined to measure the percentage of correct recommendations for item i obtained by a recommender b out of all its recommendations, as indicated in formula 7 [35].

$$\mathrm{Trust}^I(b,i) = \frac{\mathrm{card}\left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{CorrectSet}(b) : k = i\right\}}{\mathrm{card}\left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{RecSet}(b) : k = i\right\}} \quad (7)$$

Formula 6 can be used to represent the reputation of a user because it allows for calculating the overall trust of a given user in all users based on its common ratings of all items [37] [38]. On the other hand, formula 7 highlights the reputation of a given user among all users based on its common ratings for a specific item.

In the same context, drawing inspiration from the work of [39], the trust of a given user a in another user b (recommender) based on their common ratings for all items can be defined using formula 8 [40]:

$$\mathrm{Trust}^U(a \to b) = \frac{\mathrm{card}\left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{correctSet}(b) : j=a\right\}}{\mathrm{card}\left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{RecSet}(b) : j=a\right\}} \quad (8)$$

Where $Trust^U(a \to b)$ is the trust of user a in recommender b, calculated as the percentage of correct recommendations in which recommender b participated with user a based on their common ratings of all items.

From formula 8, the trust of user a in recommender b for a particular item i denoted as $Trust^U(a \to b, i)$ can be deduced by the percentage of correct recommendations in which recommender b participated with user a based only on this item, as indicated in formula 9 below:

$$\mathrm{Trust}^U(a \to b, i) = \frac{\mathrm{card}\left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{CorrectSet}(b) : j=a \,\&\, k=i\right\}}{\mathrm{card}\left\{\left(P^b_{j,k}, r_{j,k}\right) \in \mathrm{RecSet}(b) : j=a \,\&\, k=i\right\}} \quad (9)$$

In the following, we utilized formula 8 to infer the implicit trust between users from their POI ratings.

This type of trust will be used to compute the rating prediction using formula 10.

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N} (\mathrm{r}_{\mathrm{b},x} - \overline{r_b}) * Trust^U(\mathrm{a} \rightarrow \mathrm{b})}{\sum_{b=1}^{N} Trust^U(\mathrm{a} \rightarrow \mathrm{b})} \tag{10}$$

$Trust^U(\mathrm{a} \rightarrow \mathrm{b})$: Trust derived from ratings.

In this article, we used formula 2 above to compute the trust score of a given user based on their accurate predictions for future POI ratings. The same principle is also used to compute users' trust levels from their check-ins, as shown in formula 11 below.

$$C_{a,i}^{b} = \overline{c_a} + (c_{b,i} - \overline{c_b}) \tag{11}$$

Where:

- $C_{a,i}^{b}$: the check-in on item *i* predicted for user *a* based on user *b*.
- $c_{b,i} \in \{0, 1\}$: denotes the check-in of POI *i* by user *b*.
- $\overline{c_a}$: denotes the average check-ins of user *a*.
- $\overline{c_b}$: denotes the average check-ins of user *b*.

Formula 3 above can be applied in the case of check-ins to obtain equation 12 below:

$$Correct\_C(i, b, a) \Leftrightarrow \left|C_{a,i}^{b} - c_{a,i}\right| = 0 \tag{12}$$

Replacing the check-ins with the ratings in formulas 4 and 5 above, *RecSet_C(b)*, which represents the complete set of recommendations, is given by formula 13 below, and *CorrectSet_C(b)*, indicating the subset of correct recommendations, is given by formula 14 below:

$$\mathrm{RecSet\_C}(b) = \left\{\left(C_{1,1}^{b}, c_{1,1}\right), \ldots, \left(C_{n,m}^{b}, c_{n,m}\right)\right\} \tag{13}$$

Where:

- $C_{j,k}^{b}$ represents the prediction made by recommender *b* for the check-in that a user *j* (where j ranges from 1 to n) will give to an item *k* (where k ranges from 1 to m).
- $c_{j,k}$ represents the actual check-in of item *k* (where k varies from 1 to m) given by a user *j* (where j varies from 1 to n).

From *RecSet_C(b)*, the subset of correct recommendations denoted *CorrectSet_C(b)* is calculated using formula 14.

$$CorrectSet\_C(b) = \{(C_{j,k}^{b}, c_{j,k}) \in \mathrm{RecSet\_}C(b) : \mathrm{Correct\_}C(k, b, C_{j,k}^{b})\} \tag{14}$$

Then, leveraging check-ins, the derivation of user *a*'s trust towards user *b* can be inferred from formula 8 above and is applied by replacing the ratings with the check-ins to obtain formula 15 below:

$$Trust\_C^U(a \rightarrow b) = \frac{\mathrm{card}\left\{\left(C_{j,k}^{b}, c_{j,k}\right) \in CorrectSet\_C(b) : j = a\right\}}{\mathrm{card}\left\{\left(C_{j,k}^{b}, c_{j,k}\right) \in RecSet\_C(b) : j = a\right\}} \tag{15}$$

Finally, note that formula 16 below, deduced from formula 10 above, can be utilized to calculate the rating predictions of POIs from their check-ins.

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N} (\mathrm{r}_{\mathrm{b},x} - \overline{r_b}) * Trust\_C^U(\mathrm{a} \rightarrow \mathrm{b})}{\sum_{b=1}^{N} Trust\_C^U(\mathrm{a} \rightarrow \mathrm{b})} \tag{16}$$

$Trust\_C^U(\mathrm{a} \rightarrow \mathrm{b})$: Trust derived from check-ins.

B. *Proposed Algorithms*

After explaining how to calculate the trust between users, noted as TDM (Trust Derivation Matrix), based on their check-ins and POI ratings, we propose in this subsection two algorithms to implement these calculations. Algorithm 1 below takes as input the rating matrix, denoted UPRM (User-POI Rating Matrix), of dimension n × m (where n is the number of users and m is the number of POIs) for the calculation of the TDMR (Trust Derivation Matrix based on Rating) of dimension n × n (where n is the number of users). Algorithm 2 below takes as input the check-in matrix, denoted UPCM (User-POI Check-in Matrix), of dimension n × m (where n is the number of users and m is the number of POIs) for the calculation of the TDMC (Trust Derivation Matrix based on Check-ins) of dimension n × n (where n is the number of users). These two algorithms use formulas 10 and 16 to calculate their prediction matrices of dimension n × m (where n is the number of users and m is the number of POIs), denoted respectively TPMR (Trust Prediction Matrix based on Rating) and TPMC (Trust Prediction Matrix based on Check-ins).

4. PROPOSED MODEL

This section presents in detail the POI recommendation approach proposed by the model called Hybrid Rating Check-in Trust (HRCT). This model is based on both the ratings of POIs and user check-ins, as well as on the two algorithms explained in section 3. The first algorithm 1 uses the rating matrix denoted UPRM to firstly calculate the user/user trust matrix noted TDMR, and secondly, the matrix TPMR which contains the predictions of the POI ratings by users.

The second algorithm (ALGO2) uses the check-in matrix denoted UPCM to calculate the user/user trust matrix noted TDMC. The latter will be used to calculate the TPMC matrix which contains predictions of POI ratings by users.

Then, the two trust matrices (TDMR and TDMC) obtained from algorithms 1 and 2 above can be combined using algorithm 3 below to generate the H-Trust matrix of dimension n × n (where n represents the number of users). This matrix can be used to calculate the rating predictions of the POIs in the TPMH matrix of dimension n × m (where

Algorithm 1 User-user trust based on rating

Input:
UPRM: User-POI Rating Matrix
Output:
TDMR: Trust Derivation Matrix based on Rating
TPMR: Trust Prediction Matrix based on Rating
Var M2, M3, M4: User-User-POI Matrix of dimension $n \times n \times m$
Begin // trust between users
for each user b **do**
  for each user $a \neq b$ **do**
    for each POI i **do**
      $M2(a,b,i) \leftarrow$ meanRate(a) + Rate(b,i) − meanRate(b)
// Compute predict rating $M2(a,b,i)$ using formula 2
      $M3(a,b,i) \leftarrow |$Rate$(a,i) - M2(a,b,i)|$
// Compute distance error $M4(a,b,i)$ using equation 3
      if $M3(a,b,i) < \varepsilon$ **then**
        $M4(a,b,i) \leftarrow 1$
      else
        $M4(a,b,i) \leftarrow 0$
      end if
    end for
    $RecSet(b) \leftarrow$ sum$(M4(a,b,i))$
// the set of user b's recommendations using formula 4
    $CorrectSet(b) \leftarrow$ sum$(M4(a,b,i) \mid M4(a,b,i) = 1)$
// the set of user b's correct recommendations using formula 5
  end for
  $TDMR(a,b) \leftarrow \frac{\text{CorrectSet}(b)}{\text{RecSet}(b)}$
// Compute user-user trust TDMR (a,b) using formula 8
end for
for each user a **do**
// Compute Rating Prediction (TPMR) based on rating trust using formula 10
  for each POI x **do**
    if $UPRM$(user a, POI x) == empty **then**
// $b \in N$ set of user a's neighborhood
      for user b **do**
        $v \leftarrow \sum$((Rate(user b, POI x) − meanRate(user b)) $\times$ TDMR(user a, user b))
        $w \leftarrow \sum$ TDMR(user a, user b)
      end for
      TPMR(user a, POI x) $\leftarrow$ meanRate(user a) + $\frac{v}{w}$
    end if
  end for
end for
return TDMR, TPMR
End

Algorithm 2 User-user trust based on check-in

Input:
UPCM: User-POI Check-in Matrix
UPRM: User-POI Rating Matrix
Output:
TDMC: Trust Derivation Matrix based on Check-in
TPMC: Trust Prediction Matrix based on Check-in
Var Ma, Mb, Mc: User-User-POI Matrix of dimension $n \times n \times m$
Begin // trust between users
for each user b **do**
  for each user $a \neq b$ **do**
    for each POI i **do**
      $Ma(a,b,i) \leftarrow$ meanCheck(a) + Check(b,i) − meanCheck(b)
// Compute predict check-in Ma(a, b, i) using formula 11
      $Mb(a,b,i) \leftarrow |$Check$(a,i) - Mb(a,b,i)|$
// Compute distance error Mb(a,b,i) using equation 12
      if $(Mb(a,b,i) == 0)$ **then**
        $Mc(a,b,i) \leftarrow 1$
      else
        $Mc(a,b,i) \leftarrow 0$
      end if
    end for
    $RecSet(b) \leftarrow$ sum(Mc(a,b,i))
// the set of user b's recommendations using formula 13
    $CorrectSet(b) \leftarrow \sum$(Mc$(a,b,i)$ | Mc(a,b,i) == 1)
// the set of user b's correct recommendations using formula 14
  end for
  $TDMC(a,b) \leftarrow \frac{CorrectSet(b)}{RecSet(b)}$
// Compute user-user trust TDMC(a,b) using formula 15
end for
for each user a **do**
// Compute Rating Prediction (TPMC) based on check-in trust using formula 16
  for each POI x **do**
    if $UPRM$(user a, POI x) == empty **then**
      for user b **do**
// $b \in N$ set of user a's neighborhood
        $v \leftarrow \sum$((Rate(user b, POI x) − meanRate(user b)) $\times$ TDMC(user a, user b))
        $w \leftarrow \sum$ TDMC(user a, user b)
      end for
      TPMC(user a, POI x) $\leftarrow$ meanRate(user a) + $\frac{v}{w}$
    end if
  end for
end for
return TDMC, TPMC
End

n represents the number of users and m represents the number of POIs), utilizing formula 17 below as described in algorithm 3:

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N} (\mathrm{r}_{\mathrm{b,x}} - \overline{r_b}) * \textit{H-Trust}^{U}(\mathrm{a} \rightarrow \mathrm{b})}{\sum_{b=1}^{N} \textit{H-Trust}^{U}(\mathrm{a} \rightarrow \mathrm{b})} \quad (17)$$

Where:

- $P_{a,x}$: The predicted rating for user a on item x.
- $\overline{r_a}$: The average ratings of user a for all items.
- $\mathrm{r}_{\mathrm{b,x}}$: The actual rating given to item x by user b.
- *H-Trust*^{U}(a $\rightarrow$ b) : Trust derived from both ratings and check-ins.

In Figure 1 below, the HRCT model comprises 5 main steps (a to e in Figure 1), with each step described based on the data it manipulates and the algorithms it employs. These steps can be summarized as follows:

a) The user can activate their access to the LBSN by logging into their own session. This will load the rating/check-in data related to their smartphone user profile and GPS location context.

b) After loading this rating/check-in data (the UPRM and UPCM matrices), the HRCT model can employ

Algorithm 3 Fusion rating and check-in user-user trust

Input:
TDMR: Trust Derivation Matrix based on Rating
TDMC: Trust Derivation Matrix based on Check-in
UPRM: User-POI Rating Matrix
Output:
H-Trust: User-user hybrid trust matrix
TPMH: Trust Prediction Matrix based on Hybrid Trust

Begin // combine user-user trust
for each user x **do**
  for each user $y \neq x$ **do**
    if (TDMR(x, y) exist and TDMC(x, y) exist) **then**
      H-Trust(x, y) $\leftarrow \frac{2 \cdot TDMR(x,y) \cdot TDMC(x,y)}{TDMR(x,y)+TDMC(x,y)}$
    else if (TDMR (x, y) exist and TDMC (x, y) ! exist) **then**
      H-Trust(x, y)← TDMR(x, y)
    else if (TDMR(x, y) ! exist and TDMC(x, y) exist) **then**
      H-Trust(x, y)← TDMC(x, y)
    else
      H-Trust(x, y)← 0
    end if
  end for
end for
for each user a **do**
// Compute Rating Prediction (TPMH) based on Hybrid trust using formula 17
  for each POI x **do**
    if UPRM(user a, POI x) == empty **then**
      // $b \in N$ set of user a's neighborhood
      for user b **do**
        $v \leftarrow \sum$((Rate(user b, POI x) − meanRate(user b)) × H-Trust(a, b))
        $w \leftarrow \sum$ H-Trust(a, b)
      end for
    end if
    TPMH(user a, POI x) ← meanRate(user a) + $\frac{v}{w}$
  end for
end for
return H-Trust, TPMH
End

Algorithm 1 and Algorithm 2 (see arrows I.1 and II.1 in Figure 1) to calculate the user/user trust matrices TDMR and TDMC. Then, these two matrices enable the initiation of Algorithm 3 (see arrow III.1 in Figure 1), which can deduce the H-Trust matrix. Finally, these three trust matrices: TDMR, TDMC and H-Trust enable the computation of the three rating prediction matrices: TPMR, TPMC and TPMH, respectively (see arrows I.2, II.2 and III.2 in Figure 1).

c) These three prediction matrices allow for the generation of three lists of POIs, each containing K POIs ranked from the most to the least interesting (relevant). These lists can be merged and displayed on a map interface.

d) After browsing the available POIs on the map, the user can select the POI that suits their preferences and proceed to rate it or check in at its location.

e) This user's rating and check-in will be included in the UPRM and UPCM matrices, enhancing the dataset for future system recommendations.

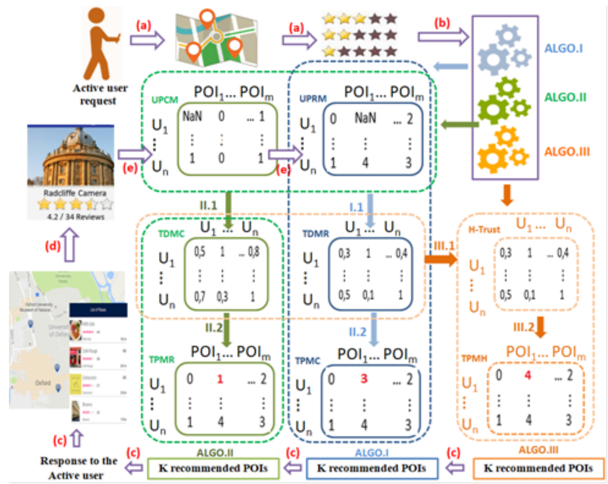


Figure 1. Functional Description of the HRCT Model

5. EXPERIMENTATION AND RESULTS

In this section, we assess the effectiveness of our HRCT model by evaluating key metrics such as RMSE, Precision and Recall. This evaluation is conducted using a dataset obtained during the test phase of an LBSN. We employ these metrics to compare the performance of the three variants of the HRCT model corresponding to the three algorithms presented previously. Next, these three model variants will be compared with similarity measures from the literature, such as Pearson Correlation Coefficient (PCC), Cosine Similarity and Jaccard Similarity. Then, various combinations of the HRCT model variants and algorithms utilizing the aforementioned similarities are weighted to enhance the performance of our approach. Finally, a study of the data sparsity is conducted to demonstrate the contribution of our model in addressing this issue. It is important to note that in the following, the results of ALGO1, ALGO2, and ALGO3 are respectively denoted as R-Trust, C-Trust and H-Trust.

A. Experimental Setup

To compute the parameters (RMSE, Precision and Recall) for comparing the HRCT model (R-Trust, C-Trust and H-Trust) with other approaches (PCC, Cosine and Jaccard) during the LBSN test phase, we utilize a dataset (currently being collected) as shown in Table II, and a set of hyperparameters defined in Table III. This dataset comprises user interactions with POIs via ratings and check-ins, while the hyperparameters specify the settings to be employed for all comparisons conducted in Section V.

TABLE II. Description of data set columns

Column name	Values	Explanation
User_ID	integer	The identifier assigned to a given user
POI_ID	integer	The identifier assigned to a given POI
Rating_User_POI	{1,2,3,4,5}	The rating given by a user to a POI
Check-in_User_POI	0/1	The check-in made by a user on a POI

TABLE III. List of the HRCT Hyperparameters

Parameter Settings	Values	Explanation
ε	] 0, 1[	The threshold; a precision parameter
Training set	70%..90%	The train set (trust)
Test set	10%..30%	The test set (prediction, evaluation)
N	{1, 2, 3, ..,20}	The set of user's neighborhood

B. Evaluation metrics

To evaluate the HRCT model's performance, we utilize the dataset and hyperparameters described earlier, along with the RMSE, Precision and Recall metrics.

a) The RMSE metric:

The RMSE parameter enables the evaluation of the disparity between the actual ratings (denoted as r_i) provided by users and those predicted (denoted as P_i) by the RS using the HRCT model [41]. This parameter is computed using the following formula 18:

$$RMSE_{user} = \sqrt{\frac{\sum_{i=1}^{n}(r_i - P_i)^2}{n}} \quad (18)$$

Where:

- r_i : is the i rating provided by this user.
- P_i : represents the predicted rating i derived from a particular model.
- n : represents the total number of ratings made by the user.

This metric enables the evaluation of the accuracy of ratings predicted by various versions of the HRCT model and facilitates their comparison with other existing POI recommendation approaches in the literature.

b) Precision and Recall metrics:
Precision and recall are commonly used to assess the quality of POI lists provided by an RS.

The precision of an RS for a user i measures the proportion of pertinent (truly relevant) POI recommendations in the list of suggested POIs [42], as indicated in formula 19 below:

$$Precision_{RS}(i) = \frac{\text{Card}\left\{\text{POIs}_{\text{ rec and pert}}\right\}}{\text{Card}\left\{\text{POIs}_{\text{ rec}}\right\}} \quad (19)$$

Where:

- $\text{POIs}_{\text{ rec and pert}}$: is the set of recommended POIs wich are pertinent for the user i.
- $\text{POIs}_{\text{ rec}}$: is the set of recommended POIs for user i.

Recall measures the ratio of pertinent POI recommendations among all relevant POIs for a user i as indicated in formula 20 below [43]:

$$Recall_{RS}(i) = \frac{\text{Card}\left\{\text{POIs}_{\text{ rec and pert}}\right\}}{\text{Card}\left\{\text{POIs}_{\text{ pert}}\right\}} \quad (20)$$

Where:

- $\text{POIs}_{\text{ pert}}$:is the set of pertinent POIs for user i.

In our study, these two metrics serve to assess the quality of recommendations offered by the three variants of our HRCT model on one hand, and to compare these recommendations with those obtained from other models in the state-of-the-art.

C. Comparison of the HRCT Model Variants

To compare the three variants of the HRCT system (Algorithm 1: R-Trust, Algorithm 2: C-Trust and Algorithm 3: H-Trust), the dataset (detailed in Table II), which contains user ratings and check-ins of POIs, is divided into two parts based on the hyperparameters in Table III: 80% for training and 20% for testing (see arrows I.1 and II.1 in Figure 2). The system then takes the first part (80% of the dataset) as input for Algorithms 1, 2 and 3, as explained in Sections 3 and 4. These algorithms use this training portion to construct the implicit trust matrices of the HRCT model, which will be used for calculating predictions, as indicated by arrow I.2 in Figure 2. Finally, the trust matrices derived from the training portion of the dataset will be utilized to predict the ratings corresponding to the test portion of the same dataset (see arrows I.3 and II.2 in Figure 2). The predictions derived from this process will be compared to the actual values present in the test portion of the dataset using the RMSE (Root Mean Square Error) evaluation metric, while Precision and Recall metrics will be utilized to assess the precision and quality of the POI recommendations (see arrow III in Figure 2).

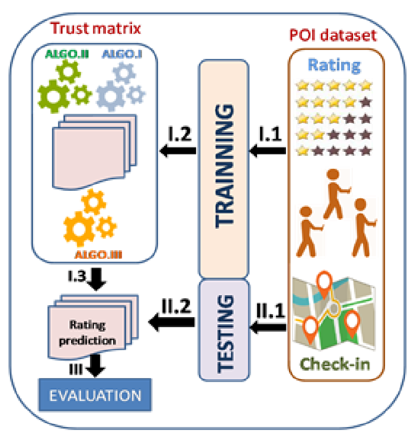


Figure 2. Evaluation Framework for the Three Variants of the HRCT Model

In the following, Figure 3, 4 and 5 illustrate a comparison of the three variants of the HRCT model regarding RMSE, Precision and Recall.

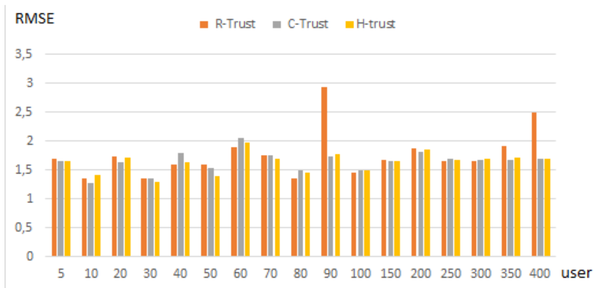


Figure 3. Comparison of R-Trust, C-Trust and H-Trust Using the

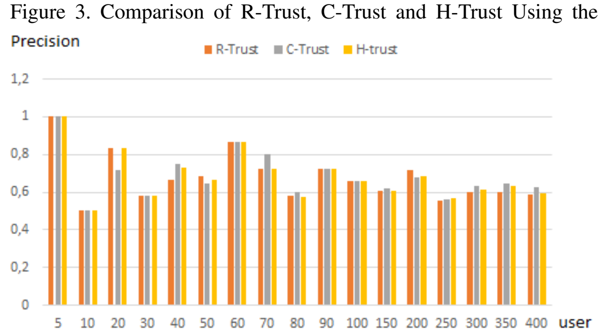


Figure 4. Comparison of R-Trust, C-Trust and H-Trust Using the

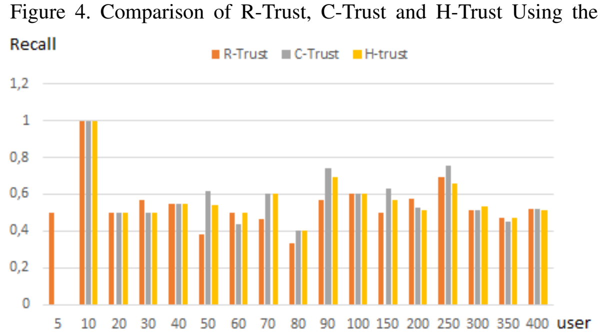


Figure 5. Comparison of R-Trust, C-Trust and H-Trust Using the Recall Parameter

In Table IV below, the C-Trust algorithm demonstrates superior performance in terms of Precision and Recall compared to the R-Trust and H-Trust algorithms. Conversely, the H-Trust algorithm exhibits better accuracy in recommendation (RMSE) compared to the R-Trust and C-Trust algorithms.

TABLE IV. Comparison of the Three Variants of the HRCT Model Using the Average of Parameters: RMSE, Precision and Recall

	Average			Deviation		
	R-Trust	C-Trust	H-Trust	R-Trust	C-Trust	H-Trust
Precision	0.675	**0.682**	0.679	0.1277	0.120	0.125
Recall	0.543	**0.549**	0.537	0.142	0.202	0.190
RMSE	1.760	1.641	**1.631**	0.408	0.181	0.174

D. Compararison of HRCT Model with Other Models

In this section, we compare the HRCT system with other models for POI recommendation. These models utilize various similarity measures, including user ratings such as Pearson Correlation Coefficient (PCC) [44] similarity, Cosine similarity [45] and Jaccard similarity [46], as well as user check-ins, also employing Pearson Correlation Coefficient (PCC) [47] similarity, Cosine similarity [48] [49] [50] and Jaccard similarity [51]. To achieve this objective, we divide the dataset described in Table II into two segments: 80% for training and 20% for testing (as depicted by arrow I and arrow II in Figure 6), employing the same hyperparameters outlined in Table III. The initial portion (80% of the dataset) serves as input for computing the user/user trust matrices (R-Trust, C-Trust and H-Trust), utilized for predicting POI ratings, as indicated by arrow a.1 in Figure 6 below. This same training portion (80% of the dataset) is also utilized as input for computing the user/user similarity matrices (PCC, Cosine and Jaccard), which are then used for rating prediction, as indicated by arrow b.1 in Figure 6 below.

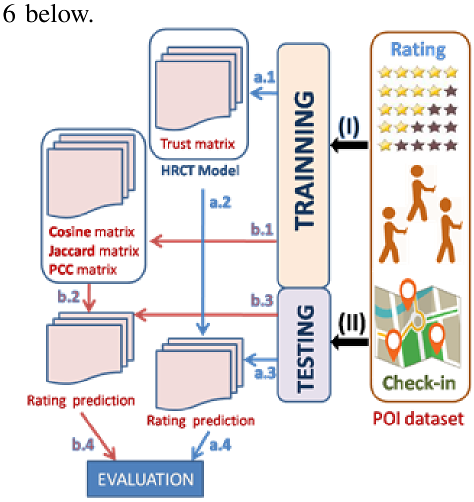


Figure 6. Comparison Methodology: HRCT Model versus Other Models

Finally, the trust matrices (R-Trust, C-Trust and H-Trust) derived from the training portion will be employed to compute the rating predictions corresponding to the testing set (as indicated by arrow a.2 and arrow a.3 in Figure 6). In a similar manner, the similarity matrices derived from the same training portion of this dataset will also be employed to compute the rating predictions corresponding to the testing set of this dataset (as depicted by arrow b.2 and arrow b.3 in Figure 6).

These predictions derived from these two processes (trust and similarity) can be compared using the RMSE (Root Mean Square Error) and Precision/Recall metrics. The RMSE metric allows for the comparison of the actual values in the test dataset with those predicted by the HRCT model and the models based on PCC, Cosine and Jaccard similarities. On the other hand, Precision/Recall metrics are utilized to assess the quality of POI recommendations from these similarity models compared to the HRCT model (refer to arrows a.4 and b.4 in Figure 6).

Figure 7, 8 and 9 below illustrate the comparative analysis between two HRCT model variants (C-Trust and H-Trust) and the PCC, Cosine and Jaccard similarity methods across RMSE, Precision and Recall metrics.

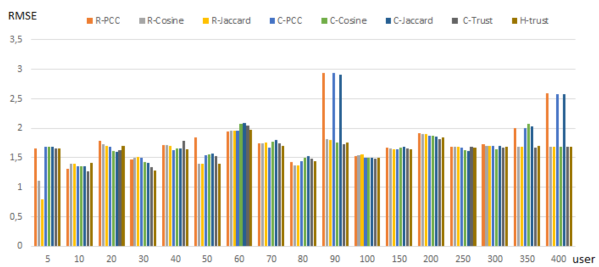


Figure 7. Comparison of C-Trust and H-Trust Variants of the HRCT Model with R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine and C-Jaccard Similarity Approaches Using RMSE Metric

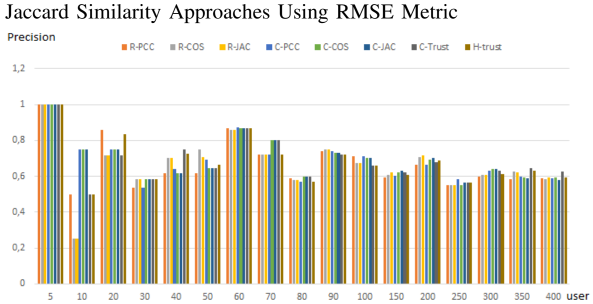


Figure 8. Comparison of C-Trust and H-Trust Variants of the HRCT Model with R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine and C-

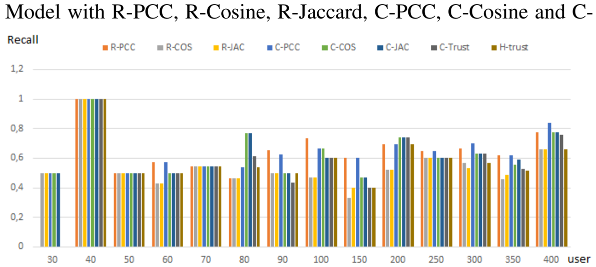


Figure 9. Comparison of C-Trust and H-Trust Variants of the HRCT Model with R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine and C-Jaccard Similarity Approaches Using Recall Metric

In the following, Table V illustrates the results obtained by calculating the RMSE and Precision/Recall metrics relating to LBSN user groups ranging from 5 to 400 users. These results allow a comparison between the H-Trust and C-Trust variants of the HRCT model and the R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine and C-Jaccard similarity models. Note that R-PCC, R-Cosine and R-Jaccard are user similarities based on their ratings, while C-PCC, C-Cosine and C-Jaccard are user similarities based on their check-ins.

In Table V, AVG and Dev represent the average and the standard deviation of the values obtained by the different techniques mentioned earlier.

In this table, the H-Trust algorithm outperforms the C-PCC, C-Cosine and C-Jaccard algorithms when using check-ins as input dataset. However, this algorithm performs less well in terms of RMSE compared to the R-Cosine and R-Jaccard algorithms.

Furthermore, Algorithm 2 (C-Trust) performs better than the R-PCC, R-Cosine and R-Jaccard algorithms when using ratings as input. However, this algorithm is less effective in terms of Precision compared to the C-PCC, C-Cosine and C-Jaccard algorithms when using the check-in dataset.

Finally, the C-Trust algorithm outperforms the R-Cosine and R-Jaccard algorithms when using ratings as input dataset. However, this algorithm performs less well in terms of Recall compared to the R-PCC, C-PCC, C-Cosine and C-Jaccard algorithms.

E. Combining the HRCT Model with Other Models

In this subsection, two studies on combining the HCRT model with other similarity models are presented. These two studies use the same dataset with the same proportions (80% for training and 20% for testing) to explore combinations between the HRCT model and the PCC, Cosine and Jaccard similarity models (see Figure 10).

These combinations will allow for two types of predictions. The first type of predictions concerns the combination of Algorithm 1, denoted as R-Trust (trust based on POI's Rating), with the PCC, Cosine and Jaccard similarities.

The second type of predictions concerns the combination of Algorithm 2, denoted as C-Trust (trust based on POI's check-ins), with the PCC, Cosine and Jaccard similarities.

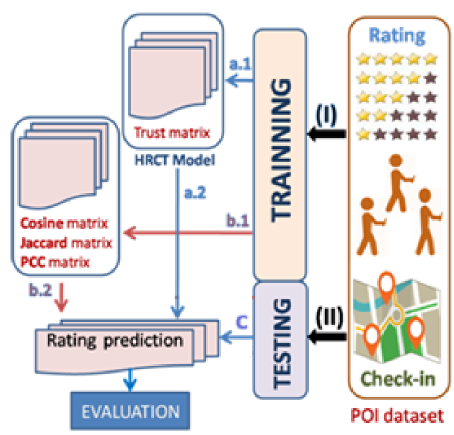


Figure 10. Combining the HRCT Model with PCC, Cosine and Jaccard Similarities

TABLE V. HRCT Model Performance Evaluation Using RMSE, Precision and Recall

	Techniques							
Metrics	R-PCC	R-Cosine	R-Jaccard	C-PCC	C-Cosine	C-Jaccard	C-Trust	H-Trust
AVG_Precision	0,667	0,6624	0,6615	0,6865	0,6907	**0, 6916**	0,6828	0,6794
AVG_Recall	0,6057	0,5313	0,5348	**0, 6493**	0,6056	0,61	0,5495	0,5375
AVG_RMSE	1,822	1,624	**1, 605**	1,789	1,677	1,8	1,641	1,6314
Dev_Precision	0,6852	0,5966	0,5847	0,6475	0,5955	0,6647	0,1206	0,1259
Dev_Recall	0,6852	0,5966	0,5847	0,6475	0,5955	0,6647	0,2020	0,1905
Dev_RMSE	0,4031	0,2116	0,2655	0,4076	0,1972	0,4087	0,1818	0,1740

In the following sections, Figure 11, 12 and 13 compare two variants of the HRCT model (R-Trust and H-Trust) with their combinations with similarities derived from ratings: R-Trust-PCC, R-Trust-Cosine, and R-Trust-Jaccard, using RMSE, Precision and Recall metrics.

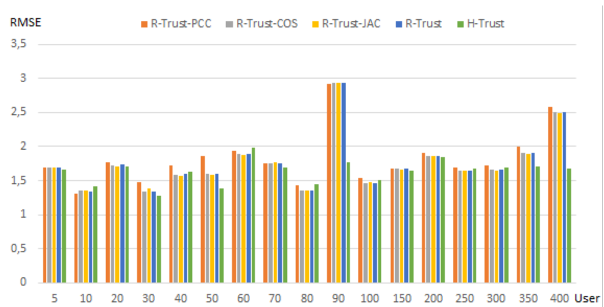


Figure 11. Comparison of R-Trust and H-trust variants of the HRCT model with combinations of R-Trust with PCC, Cosine and Jaccard

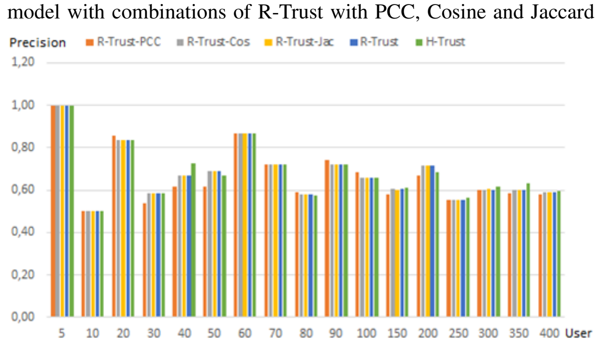


Figure 12. Comparison of R-Trust and H-trust variants of the HRCT model with combinations of R-Trust with PCC, Cosine and Jaccard

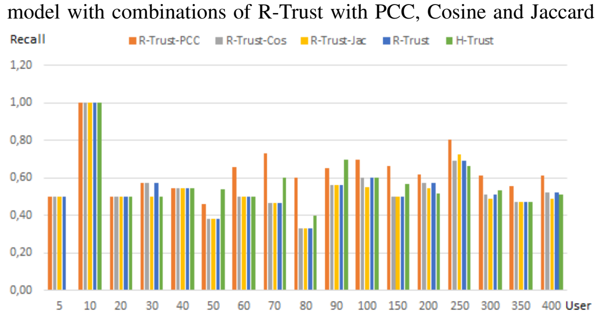


Figure 13. Comparison of R-Trust and H-trust variants of the HRCT model with combinations of R-Trust with PCC, Cosine and Jaccard using the Recall Metric

Table VI below compares the prediction performances of three combinations (R-Trust + PCC similarity, R-Trust + Cosine similarity and R-Trust + Jaccard similarity) using two algorithms: Algorithm 1 (R-Trust) and Algorithm 3 (H-Trust).

In Table VI, the R-Trust-JAC recommendation algorithm, which combines trust based on POI ratings with Jaccard similarity between these POIs, demonstrates better RMSE in comparison to R-Trust-PCC, R-Trust-COS and R-Trust algorithms. However, the R-Trust-JAC algorithm performs less effectively than algorithm 3 (H-Trust) of the HRCT model.

In the following, Figure 14, 15 and 16 provide a comparison of two variants of the HRCT model (C-Trust and H-Trust) along with their combinations with check-in similarities denoted as C-Trust-PCC, C-Trust-Cosine and C-Trust-Jaccard, evaluated using RMSE, Precision and Recall parameters.

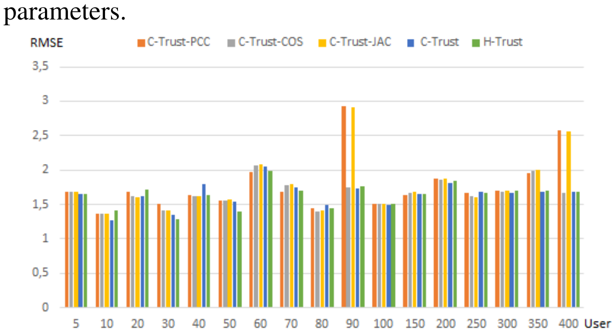


Figure 14. Comparative Analysis of C-Trust and H-Trust Variants of the HRCT Model with Combinations of C-Trust, PCC, Cosine and Jaccard Using the RMSE Metric

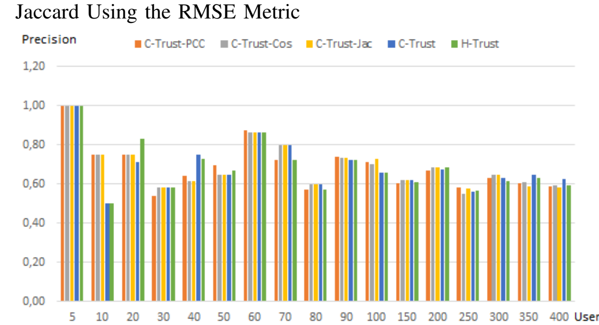


Figure 15. Comparative Analysis of C-Trust and H-Trust Variants of the HRCT Model with Combinations of C-Trust, PCC, Cosine and Jaccard Using the Precision Metric

TABLE VI. Analyzing R-Trust Combinations with PCC, Cosine and Jaccard: RMSE, Precision and Recall Evaluation

	Average					Deviation				
	R-Trust-PCC	R-Trust-COS	R-Trust-JAC	R-Trust	H-Trust	R-Trust-PCC	R-Trust-COS	R-Trust-JAC	R-Trust	H-Trust
Precision	0.6644	0.6753	0.6755	0.6753	**0.6794**	0.1349	0.1277	0.1276	0.1277	0.1259
Recall	**0.6350**	0.5433	0.5332	0.5433	0.5375	0.1288	0.1427	0.1450	0.1427	0.1905
RMSE	1.8234	1.7604	1,7591	1.7605	**1.6314**	0.3997	0.4086	0.4055	0.4087	0.1740

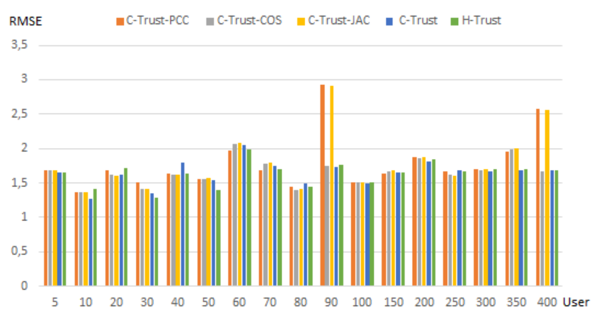


Figure 16. Comparative Analysis of C-Trust and H-Trust Variants of the HRCT Model with Combinations of C-Trust, PCC, Cosine and Jaccard Using the Recall Metric

Table VII below also includes a comparison of the prediction performances among the three combinations: C-Trust with PCC similarity, C-Trust with Cosine similarity, and C-Trust with Jaccard similarity, using the two algorithms, Algorithm 2 (C-Trust) and Algorithm 3 (H-Trust).

As depicted in Table VII, the C-Trust-COS recommendation algorithm, which combines trust based on POI check-ins with Cosine similarity, outperforms the other two combination algorithms (C-Trust-JAC and C-Trust-PCC). However, the C-Trust-COS algorithm remains less good, in terms of RMSE, than the HRCT model algorithms (H-Trust and C-Trust).

F. HRCT Model and Sparsity

In the context of matrices containing POI ratings and user check-ins, *sparsity* refers to the density, which is the proportion of non-zero values to the total number of values in the matrix, as illustrated in formula 21 below [52]:

$$Sparsity \; = 1 - \frac{total\ trust\ values}{no.\ of\ users\ *\ no.\ of\ users} \tag{21}$$

where:

- *total trust values* : trust values between users.
- *no. of users* : the number of users in the user/user trust matrix.

The formula above quantifies the sparsity of data in a given matrix. A higher value indicates that the matrix contains more non-zero or missing values, which can adversely impact the accuracy of prediction calculations in recommendation algorithms.

The trust matrices TDMR and TDMC (refer to Figure 1) are merged using algorithm 3 to create the "H-Trust" which is denser than both of these matrices.

Similarly, the PCC, Cosine and Jaccard similarity matrices derived from the ratings and check-ins can be combined to produce the H-PCC, H-Cosine and H-Jaccard similarity matrices, which are denser than their original similarity matrices (before combination).

To further improve our approach based on the HRCT model, a comparison of the sparsity between the combined trust matrix denoted H-Trust and the other combinations of similarity matrices (H-PCC, H-Cosine and H-Jaccard) is presented in Table VIII below.

TABLE VIII. Comparison of H-Trust Matrix Sparsity with H-PCC, H-Cosine and H-Jaccard Matrices

User	H-Trust	H-PCC	H-Cos	H-Jac
100	42.4360	58.9633	45.0204	47.2838
150	49.2716	65.7986	53.5590	56.0860
200	50.7079	67.3251	54.2820	56.6311
250	53.1374	68.9892	56.6526	59.7178
300	54.8798	69.4506	58.3775	61.1782
350	59.2518	70.9386	61.0586	63.4621
400	61.4440	71.7286	63.5435	65.7825
AVG	**41.8306**	57.0043	44.7805	48.6831

Table VIII above indicates that the H-Trust matrix is denser than the H-PCC, H-Cosine and H-Jaccard matrices. Additionally, this trust matrix can utilize the principle of trust propagation to further reduce its sparsity percentage.

6. RESULTS AND DISCUSSIONS

At the beginning of the experimentation phase, the three variants of the HRCT model are compared using a dataset that is split into 80% for training and 20% for testing. Next, the R-Trust, C-Trust and H-Trust algorithms use the training portion to build the HRCT model's trust matrices, which are then used to predict the POI ratings in the testing data. The performance of this model is evaluated using RMSE and Precision/Recall metrics to assess the quality of POI recommendations. The results indicate that the C-Trust algorithm achieves better performance in terms of Precision and Recall, whereas the H-Trust algorithm performs better in terms of RMSE.

The HRCT model is then compared to other POI recommendation models using different similarity measures, including Pearson Correlation Coefficient, Cosine and Jaccard. Performance is evaluated in the same way, using the

TABLE VII. Comparative Analysis of C-Trust Combinations with PCC, Cosine and Jaccard Using RMSE, Precision and Recall

	Average					Deviation				
	C-Trust-PCC	C-Trust-COS	C-Trust-JAC	C-Trust	H-Trust	C-Trust-PCC	C-Trust-COS	C-Trust-JAC	C-Trust	H-Trust
Precision	0.6869	0.6912	**0.6923**	0.6828	0.6794	0.1176	0.1165	0.1170	0.1206	0.1259
Recall	**0.6493**	0.6130	0.6140	0.5495	0.5375	0.1234	0.1456	0.1443	0.2020	0.1905
RMSE	1.7855	1.6608	1.7867	1.6410	**1.6314**	0.4050	0.1944	0.4126	0.1818	0.1740

same dataset and the same hyperparameters. The results indicate that the H-Trust algorithm outperforms some rating-based similarity models but is not as good as others.

Lastly, combinations of the HRCT model with other similarity models are investigated. performance is compared using identical evaluation metrics. The results show that while some combinations outperform others, the HRCT model remains competitive across most scenarios. Moreover, it's noted that the trust matrix of the HRCT model tends to be denser than other similarity matrices, which may improve the quality of recommendations.

7. CONCLUSIONS AND FUTURE WORK

The paper focuses on inferring trust among users of a Location-Based Social Network (LBSN) from two primary data sources: points of interest (POI) rating data and user check-in data. Initially, two trust matrices are separately calculated from these two sources. Subsequently, these resulting trust matrices are merged to generate a combined trust matrix, which incorporates users' preferences as expressed by their ratings and their visiting habits as indicated by their check-ins. Lastely, a POI recommendation system named HRCT (Hybrid Rating Check-in Trust) is formulated, utilizing three trust matrices: (1) TDMR, derived from the POI ratings, (2) TDMC, derived from user check-ins and (3) H-Trust, a fusion of these two matrices (TDMR and TDMC). Experimental test results demonstrate that the HRCT model outperforms state-of-the-art algorithms, such as collaborative filtering based on Pearson's user/user similarity, Cosine's user/user similarity and Jaccard's user/user similarity, in terms of Root Mean Square Error (RMSE) and Precision/Recall. Moreover, this model successfully tackles the issue of data sparsity in user/user similarity matrices from LBSNs. It presents a robust solution by utilizing denser user/user trust matrices, achieved through the utilization and fusion of data from both POI ratings and user check-ins.

In terms of future perspectives, the goal is to incorporate the propagation of implicit trust in LBSNs utilizing the matrix resulting from merging the two trust matrices derived from POI ratings and user check-ins. This integration aims to enhance the accuracy of the HRCT POI recommendation model by mitigating sparsity in the user/user trust matrices. Lastly, as the POI recommendation system employing the HRCT model is currently in the testing phase and our new LBSN is in the data collection phase, incorporating user feedback regarding implicit trust suggestions appears to be an intriguing prospect.

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] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "Mobile recommender systems in tourism," *Journal of network and computer applications*, vol. 39, pp. 319–333, 2014.

[2] Y. Inc., "Yelp dataset," 2021, accessed: 2021-12-01. [Online]. Available: https://www.yelp.com/dataset

[3] Gowalla, "Gowalla dataset," 2010, accessed: 2021-12-01. [Online]. Available: https://snap.stanford.edu/data/loc-gowalla.html

[4] Foursquare, "Foursquare dataset," 2013, accessed: 2021-12-01. [Online]. Available: https://sites.google.com/site/yangdingqi/home/foursquare-dataset

[5] Brightkite, "Brightkite dataset," 2010, accessed: 2021-12-01. [Online]. Available: https://snap.stanford.edu/data/loc-brightkite.html

[6] M. A. Islam, M. M. Mohammad, S. S. S. Das, and M. E. Ali, "A survey on deep learning based point-of-interest (poi) recommendations," *Neurocomputing*, vol. 472, pp. 306–325, 2022.

[7] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–45, 2014.

[8] R. Konow, W. Tan, L. Loyola, J. Pereira, and N. Baloian, "A visited item frequency based recommender system: Experimental evaluation and scenario description." *J. Univers. Comput. Sci.*, vol. 17, no. 14, pp. 2009–2028, 2011.

[9] J. Shokeen and C. Rana, "Social recommender systems: techniques, domains, metrics, datasets and future scope," *Journal of Intelligent Information Systems*, vol. 54, no. 3, pp. 633–667, 2020.

[10] R. Gao, J. Li, X. Li, C. Song, and Y. Zhou, "A personalized point-of-interest recommendation model via fusion of geo-social information," *Neurocomputing*, vol. 273, pp. 159–170, 2018.

[11] X. Ma, H. Lu, Z. Gan, and J. Zeng, "An explicit trust and distrust clustering based collaborative filtering recommendation approach," *Electronic Commerce Research and Applications*, vol. 25, pp. 29–39, 2017.

[12] H. Torkamaan and J. Ziegler, "Recommendations as challenges: estimating required effort and user ability for health behavior change recommendations," in *27th International Conference on Intelligent User Interfaces*, 2022, pp. 106–119.

[13] M. Kunaver and T. Požrl, "Diversity in recommender systems–a survey," *Knowledge-based systems*, vol. 123, pp. 154–162, 2017.

[14] J.-H. Cho, K. Chan, and S. Adali, "A survey on trust modeling," *ACM Computing Surveys (CSUR)*, vol. 48, no. 2, pp. 1–40, 2015.

[15] M. G. Manzato, E. B. S. Junior, and R. Goularte, "Leveraging hybrid recommenders with multifaceted implicit feedback." *J. Univers. Comput. Sci.*, vol. 21, no. 2, pp. 223–247, 2015.

[16] N. Ugarte, M. Larrañaga, and A. Arruarte, "The use of recommender systems in formal learning. a systematic literature mapping." *JUCS: Journal of Universal Computer Science*, vol. 28, no. 4, 2022.

[17] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–38, 2021.

[18] F. Gasparetti, G. Sansonetti, and A. Micarelli, "Community detection in social recommender systems: a survey," *Applied Intelligence*, vol. 51, no. 6, pp. 3975–3995, 2021.

[19] M. U. Demirci and P. Karagoz, "Trust modeling in recommendation: Explicit and implicit trust model compatibility and explicit trust prediction," in *Proceedings of the 13th International Conference on Management of Digital EcoSystems*, 2021, pp. 8–14.

[20] X. Ma, H. Lu, and Z. Gan, "Implicit trust and distrust prediction for recommender systems," in *Web Information Systems Engineering–WISE 2015: 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I 16*. Springer, 2015, pp. 185–199.

[21] W. Hussein, R. M. Ismail, T. F. Gharib, and M. G. Mostafa, "A personalized recommender system based on a hybrid model." *J. Univers. Comput. Sci.*, vol. 19, no. 15, pp. 2224–2240, 2013.

[22] Q. Y. Shambour, M. M. Abualhaj, and A. A. Abu-Shareha, "Restaurant recommendations based on multi-criteria recommendation algorithm," *Journal of Universal Computer Science*, vol. 29, no. 2, p. 179, 2023.

[23] S. Medjroud, N. Dennouni, and M. Loukam, "Recommendation system: a review of trust techniques," *NCAIA'2023*, p. 84, 2023.

[24] N. Idrissi and A. Zellou, "A systematic literature review of sparsity issues in recommender systems," *Social Network Analysis and Mining*, vol. 10, no. 1, p. 15, 2020.

[25] P. Victor, M. De Cock, and C. Cornelis, "Trust and recommendations," *Recommender systems handbook*, pp. 645–675, 2011.

[26] D. Cantürk, "Trust-aware location recommendation in location-based social networks," 2021.

[27] J. Zhu, C. Wang, X. Guo, Q. Ming, J. Li, and Y. Liu, "Friend and poi recommendation based on social trust cluster in location-based social networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, pp. 1–12, 2019.

[28] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. Gong, "Trust-enhanced collaborative filtering for personalized point of interests recommendation," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6124–6132, 2019.

[29] G. Ekaterina, D. Ivan, and S. Oksana, "A trust and relevance-based point-of-interest recommendations method with inaccessible user location," *Procedia Computer Science*, vol. 178, pp. 153–161, 2020.

[30] R. Logesh and V. Subramaniyaswamy, "A reliable point of interest recommendation based on trust relevancy between users," *Wireless Personal Communications*, vol. 97, pp. 2751–2780, 2017.

[31] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 397–406.

[32] J. Zhu, Q. Ming, and Y. Liu, "Trust-distrust-aware point-of-interest recommendation in location-based social network," in *Wireless Algorithms, Systems, and Applications: 13th International Conference, WASA 2018, Tianjin, China, June 20-22, 2018, Proceedings 13*. Springer, 2018, pp. 709–719.

[33] C. Xu, A. S. Ding, and K. Zhao, "A novel poi recommendation method based on trust relationship and spatial–temporal factors," *Electronic Commerce Research and Applications*, vol. 48, p. 101060, 2021.

[34] J. An, W. Jiang, and G. Li, "Bidirectional trust-enhanced collaborative filtering for point-of-interest recommendation," *Sensors*, vol. 23, no. 8, p. 4140, 2023.

[35] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th international conference on Intelligent user interfaces*, 2005, pp. 167–174.

[36] 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.

[37] C.-S. Hwang and Y.-P. Chen, "Using trust in collaborative filtering recommendation," in *New Trends in Applied Artificial Intelligence: 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2007, Kyoto, Japan, June 26-29, 2007. Proceedings 20*. Springer, 2007, pp. 1052–1060.

[38] Q. Shambour and J. Lu, "A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services," *International Journal of Intelligent Systems*, vol. 26, no. 9, pp. 814–843, 2011.

[39] A. Zahir, Y. Yuan, and K. Moniz, "Agreereltrust—a simple implicit trust inference model for memory-based collaborative filtering recommendation systems," *Electronics*, vol. 8, no. 4, p. 427, 2019.

[40] 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.

[41] M. HADJHENNI, N. DENNOUNI, and Z. SLAMA, "Toward a systematic evaluation approach of point-of-interest recommendation algorithms of a novel smart tourism tool," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 653–670, 2024.

[42] H. J. C. Pampín, H. Jerbi, and M. P. O'Mahony, "Evaluating the relative performance of collaborative filtering recommender systems," *Journal of Universal Computer Science*, vol. 21, no. 13, pp. 1849–1868, 2015.

[43] T. Wiedmann, J. L. V. Barbosa, S. J. Rigo, and D. N. F. Barbosa, “Recsim: A model for learning objects recommendation using similarity of sessions.” *J. Univers. Comput. Sci.*, vol. 22, no. 8, pp. 1175–1200, 2016.

[44] L. Sheugh and S. H. Alizadeh, “A note on pearson correlation coefficient as a metric of similarity in recommender system,” in *2015 AI & Robotics (IRANOPEN)*. IEEE, 2015, pp. 1–6.

[45] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, “Movie recommendation system using cosine similarity and knn,” *International Journal of Engineering and Advanced Technology*, vol. 9, no. 5, pp. 556–559, 2020.

[46] S. Bag, S. K. Kumar, and M. K. Tiwari, “An efficient recommendation generation using relevant jaccard similarity,” *Information Sciences*, vol. 483, pp. 53–64, 2019.

[47] D. Zhou, B. Wang, S. M. Rahimi, and X. Wang, “A study of recommending locations on location-based social network by collaborative filtering,” in *Advances in Artificial Intelligence: 25th Canadian Conference on Artificial Intelligence, Canadian AI 2012, Toronto, ON, Canada, May 28-30, 2012. Proceedings 25*. Springer, 2012, pp. 255–266.

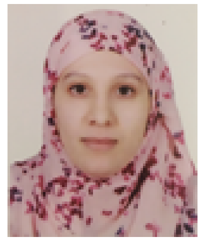
[48] J. Zeng, X. He, F. Li, and Y. Wu, “A recommendation algorithm for point of interest using time-based collaborative filtering,” *International Journal of Information Technology and Management*, vol. 19, no. 4, pp. 347–357, 2020.

[49] L. Guo, H. Jiang, X. Wang, and F. Liu, “Learning to recommend point-of-interest with the weighted bayesian personalized ranking method in lbsns,” *Information*, vol. 8, no. 1, p. 20, 2017.

[50] C. Song, J. Wen, and S. Li, “Personalized poi recommendation based on check-in data and geographical-regional influence,” in *Proceedings of the 3rd International conference on machine learning and soft computing*, 2019, pp. 128–133.

[51] B. Eravci, N. Bulut, C. Etemoglu, and H. Ferhatosmanoğlu, “Location recommendations for new businesses using check-in data,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2016, pp. 1110–1117.

[52] M. Singh, “Scalability and sparsity issues in recommender datasets: a survey,” *Knowledge and Information Systems*, vol. 62, no. 1, pp. 1–43, 2020.



Sara Medjroud is presently enrolled as a PhD student in the Computer Science Department at Hassiba Benbouali University of Chlef, located in western Algeria. Additionally, she is an active member of the university’s laboratory. In 2019, she attained her Master’s degree in Computer Science from Ibn Khaldoun University of Tiaret, Algeria. Prior to this, in 2017, she completed her Bachelor’s degree in Computer Science from the Department of Computer Science at Hassiba Benbouali University of Chlef. She is presently dedicated to her doctoral research on artificial intelligence, POI recommendations, and algorithms that leverage trust among users in Location-Based Social Networking (LBSN).

Nassim Dennouni is currently holds the position of associate professor at the Higher School of Management of Tlemcen since October 2022. Concurrently, he leads the ICAR team at the LIA laboratory of Hassiba BENBOUALI University in Chlef (Algeria) and he is a member of the ISIBA team at the EEDIS laboratory of the Department of Computer Science at Djillali Liabes University in Sidi Bel Abbes (Algeria). He obtained his university accreditation from Djillali Liabes University in Sidi Bel Abbes in 2018, following his PhD in computer science from the same university in 2016. His primary research areas encompass artificial intelligence, smart tourism, education sciences, ubiquitous computing, mobile learning, activity orchestration, and POI recommendation.

Mourad Loukam is currently a Lecturer-Associate Professor at the Computer Science Department of Hassiba Benbouali University of Chlef (western Algeria). He received his PHD in Computer Science, natural language processing-speciality, from the Higher School of Computer Science of Algiers (Algeria) in 2016. He has several published papers and Communications in various Journals and Conferences. His current research interests are: NLP, Speech Recognition, Text Mining and Knowledge Representation.