

Improving recommendations with implicit trust propagation from ratings and check-ins

Sara Medjroud¹, Nassim Dennouni^{2,3}, Mourad Loukam²

¹Mechanics and Energy Laboratory, Hassiba Benbouali University, Chlef, Algeria

²Computer Science and its Applications Laboratory, Hassiba Benbouali University, Chlef, Algeria

³Higher School of Management, Tlemcen, Algeria

Article Info

Article history:

Received Oct 17, 2024

Revised Jul 14, 2025

Accepted Oct 14, 2025

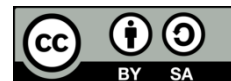
Keywords:

Collaborative filtering
Implicit trust
location-based social network
Point of interest
Propagation
Recommender system
Sparsity

ABSTRACT

This paper investigates how the propagation of implicit trust between users affects the quality of point-of-interest (POI) recommendations in location-based social networks (LBSNs). Through the analysis of user interactions via ratings and check-ins, this work proposes a recommendation model known as propagation of rating/check-in for implicit trust (PRCT). This model relies on two primary approaches: Similarity trust rating (STR), which utilizes user ratings, and similarity trust check-in (STC), which focuses on check-ins data. Both approaches employ trust propagation to enhance their similarity matrices between users. An evaluation of the PRCT model using the Yelp dataset shows that the STR approach surpasses other variants in terms of PRECISION and RECALL, while the STC approach demonstrates superior performance in terms of RMSE. Furthermore, while trust propagation in the PRCT model increases the density of its similarity matrices, it does not consistently enhance its PRECISION parameter. Only the similarity Jaccard check-in (SJC) and similarity cosine check-in (SCC) approaches show a significant improvement of this parameter.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Sara Medjroud
Mechanics and Energy Laboratory, Hassiba Benbouali University
Chlef, Algeria
Email: s.medjroud@univ-chlef.dz

1. INTRODUCTION

Location-based social networks (LBSNs) are becoming more and more important in various fields, including Smart Tourism [1], as they can shape users' decisions and behaviors by providing personalized recommendations [2]. These recommendations for hotels, restaurants, historical monuments ... are derived from interactions like user ratings and check-ins with points of interest (POIs) already available in the LBSNs. However, the implicit trust that can spread among users is a major factor in the quality and relevance of these recommendations. This trust, although not explicitly stated, can be inferred from shared behaviors and the data collected on LBSNs.

Collaborative filtering (CF) has proven effective for implementing POI recommendation systems (RSs), but its success largely depends on its ability to accurately identify similar neighbors. However, due to the sparsity of data in the user-item matrix, it is often challenging to identify these neighbors, particularly in the case of a new user or a new item during the cold start of the RS. To overcome these limitations, explicit trust-based recommender systems (TRS) have been developed. Explicit trust is gathered directly from users, forming a "trust statement network". However, this imposes an extra burden on users, as they are required to explicitly express their trust in others. Implicit trust deduced from user behavior, appears to be a promising solution to this issue since it makes it possible to build trust connections through the analysis of evaluation

patterns. Furthermore, this approach can be especially effective as it also enables the integration of trust propagation between LBSN users.

Existing literature indicates that recommender systems utilizing explicit trust information outperform CF-based systems in terms of accuracy [3]–[5]. However, for various reasons, it is more challenging to obtain direct trust relationships compared to implicit trust relationships, which are inferred from user preferences. For this reason, several implicit trust filtering techniques are becoming more prominent in the literature, as they allow for the deduction of trust scores between users based on their item rating data [6]–[8]. To enhance the effectiveness of POI RSs, researchers concentrate on factors such as the geographic location of POIs and the trust relationships among users within LBSNs [9]. These elements assist users in discovering locations that align with their preferences in real-time, based on the analysis of trajectories and the sharing of feedback from their previous visits. Recommender systems calculate trust scores between users based on their ratings of POIs [10], their friendships [11]–[13], their check-ins [14]–[16], and their reviews [17].

The aim of this study is to investigate how implicit trust propagation [18] affects the quality of POI recommendations in a LBSN and to explore how this propagation can help address issues related to data sparsity. To address this question, this paper investigates methods for inferring implicit trust from users' interactions with POIs through their ratings and check-ins, and subsequently applies the principle of trust propagation [19] to enhance the density of trust matrices. These methods can address the limitations of data sparsity in recommender systems, particularly in cases where two users do not have any common rated (checked in) items. Among these methods, we have selected the propagation [10] of implicit trust, which enables the inference of indirect trust relationships between users by considering the various paths that can exist within a trust network. In this paper, we introduce a recommendation model, named propagation of rating/check-in for implicate trust (PRCT), which is founded on two primary approaches: one utilizing user ratings (STR) and the other relying on check-ins (STC). This model seeks to enhance POI predictions by these two types of similarities and employing the propagation principle to enrich the trust matrices among users.

The paper also outlines the necessary algorithms to implement these concepts and provides an evaluation of the PRCT model based on experiments conducted with the Yelp dataset. In order to compare this model with other recommendation approaches from the literature using common metrics such as RMSE, PRECISION, and RECALL, we evaluated its performance against works based on trust (O'Donovan and Smyth [20]) and those based on CF using Pearson, Jaccard, and Cosine similarities. Based on our knowledge, no prior work combines user-memory-based CF with trust inferred from past prediction consistency deduced from check-in data and incorporates trust propagation for POI recommendation.

The remainder of the paper is organized as follows: Section 2 details the methodology, including how trust is calculated and propagated. Section 3 introduces the proposed PRCT model, describes its functioning, and outlines the algorithms used. Section 4 presents experimental results and discusses the findings using standard evaluation metrics (RMSE, Precision, Recall), comparing PRCT with baseline models from the literature. Finally, section 5 concludes the study, highlighting its contributions and relevance to POI recommendation research.

2. METHOD

This section begins by explaining how to utilize user ratings and POI check-ins to deduce implicit trust between users. It then discusses applying the trust propagation principle to increase the density of the user trust matrix.

2.1. Calculating implicit trust

O'Donovan and Smith define trust as the extent to which a partner's profile has proven reliable in offering accurate recommendations in the past. For instance, a profile that has consistently provided accurate recommendations in the past will be regarded as more trustworthy than a profile that has mostly produced inaccurate predictions. This type of evaluation can be calculated using a (1) provided [20], [21]. Note that in the following, RC refers to either a user's check-in or a POI's rating.

$$P_{a,i} = \overline{RC}_a + \frac{\sum_{b=1}^N (RC_{b,i} - \overline{RC}_b) sim(a,b)}{\sum_{b=1}^N sim(a,b)} \quad (1)$$

- $P_{a,i}$: The predicted rating (or check-in) that the active user "a" assigns to POI "i".
- $\overline{RC}_a$: The average of the ratings (or check-ins) of user "a" for all POIs.
- $RC_{b,i}$: The actual rating (check-in) made to POI "i" by user "b".
- $sim(a, b)$: The similarity between user "a" and user "b".

- N : The set of user a 's neighbors.

However, to calculate the predicted rating (or check-in) of user "a" for a specific POI "i" based solely on user "b", who is regarded as the recommender [20], (2), derived from (1) can be utilized [22],

$$P_{a,i}^b = \overline{RC}_a + (RC_{b,i} - \overline{RC}_b) \quad (2)$$

- $P_{a,i}^b$: The predicted rating (or check-in) for user "a" on POI "i" based on user "b".
- $\overline{RC}_a$: The average of the ratings (or check-ins) of user "a" for all POIs.
- $\overline{RC}_b$: The average of the ratings (or check-ins) of user "b" for all POIs.
- $RC_{b,i}$: The actual rating (or check-in) made to POI "i" by user "b".

According to O'Donovan and Smith, the prediction of a rating (or check-in) for user "a" on POI "i" based on recommender "b" is considered "correct" if the predicted rating (or check-in) $P_{a,i}^b$ is close to the actual rating (or check-in) given by user "a", denoted as $RC_{a,i}$, as indicated in (3).

$$Correct(i, b, a) \Leftrightarrow |P_{a,i}^b - RC_{a,i}| < \varepsilon \quad (3)$$

Therefore, $Correct(i, b, a)$ takes the value « 1 » if $|P_{a,i}^b - RC_{a,i}| < \varepsilon$ and « 0 » otherwise.

Then, O'Donovan and Smith use the (4) to define $RecSet(b)$ as the complete set of recommendations in which recommender "b" has been involved,

$$RecSet(b) = \{(P_{1,1}^b, RC_{1,1}), \dots, (P_{m,n}^b, RC_{m,n})\} \quad (4)$$

- $P_{j,k}^b$: represents the prediction of recommender b for the rating (or check-in) that user j (where j varies from 1 to m) will give (make) to POI k (where k varies from 1 to n).
- $RC_{j,k}$: represents the actual rating (or check-in) f POI k (where k varies from 1 to n) given (made) by user j (where j varies from 1 to m).

From $RecSet(b)$, the subset of correct recommendations, denoted as $CorrectSet(b)$, is calculated using the (5) [20].

$$CorrectSet(b) = \{(P_{j,k}^b, RC_{j,k}) \in RecSet(b) : Correct(k, b, P_{j,k}^b)\} \quad (5)$$

Finally, the concept of trust at the profile-level, denoted as $Trust^P$ for recommender "b", can be defined by the percentage of correct recommendations out of all the recommendations in which this recommender has participated, using the (6) [20].

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

Based on (6), a more refined trust metric at the item-level, denoted as $Trust^I$, can be defined to measure only the percentage of correct recommendations for POI "i" obtained by recommender "b" out of all their recommendations, as indicated in the (7) [20].

$$Trust^I(b, i) = \frac{card\{(P_{j,k}^b, RC_{j,k}) \in CorrectSet(b) : k=i\}}{card\{(P_{j,k}^b, RC_{j,k}) \in RecSet(b) : k=i\}} \quad (7)$$

In (6) can be used to represent the reputation of a user, as it allows for the calculation of the overall trust of a given user based on their common ratings (or check-ins) for all POIs [22], [23] On the other hand, (7) emphasizes the reputation of a specific user among all users based on their common ratings (or check-ins) for a particular POI.

In this same context, but drawing inspiration from the work of [8], the trust of a given user "a" in another user "b" (the recommender) based on their common ratings (or check-ins) for all POIs can be defined using (8) [24],

$$Trust^U(a \rightarrow b) = \frac{card\{(P_{j,k}^b, RC_{j,k}) \in CorrectSet(b) : j=a\}}{card\{(P_{j,k}^b, RC_{j,k}) \in RecSet(b) : j=a\}} \quad (8)$$

where $Trust^U(a \rightarrow b)$ represents 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 (or check-ins) for all POIs.

Based on (8), the trust of user "a" in recommender "b" for a particular POI "i", denoted as $Trust^U(a \rightarrow b, i)$, can be derived from the percentage of correct recommendations in which recommender "b" participated with user "a", based solely on that POI, as indicated in (9),

$$Trust^U(a \rightarrow b, i) = \frac{card\{(p_{j,k}^b, RC_{j,k}) \in CorrectSet(b): j=a \& k=i\}}{card\{(p_{j,k}^b, RC_{j,k}) \in RecSet(b): j=a \& k=i\}} \quad (9)$$

in the following, we used (8) to deduce implicit trust between users based on their ratings (or check-ins) of the POIs. This form of trust serves to calculate the predicted rating, as defined by (10).

$$P_{a,x} = \bar{r}_a + \frac{\sum_{b=1}^N (r_{b,x} - \bar{r}_b) * Trust^U(a \rightarrow b)}{\sum_{b=1}^N Trust^U(a \rightarrow b)} \quad (10)$$

- $P_{a,x}$: The predicted rating for user "a" on POI "x".
- $\bar{r}_a, \bar{r}_b$: The average of the ratings of users "a" and "b" for all POIs.
- $r_{b,x}$: The actual rating given to POI "x" by user "b".
- $Trust^U(a \rightarrow b)$: The trust based on the ratings (or check-ins) of user "a" towards user "b".

2.2. Implicite trust propagation

Sometimes, the rating and check-in matrices may contain several instances where two users have no common POIs: that is, no co-rated POIs in the rating matrix and no shared check-ins in the check-in matrix. Since there are no direct trust links between users, we propose calculating trust propagation scores to address this issue. Thus, from the direct trust network, it becomes possible to propagate trust and establish new relationships between users who do not share a direct trust link [22]. For example, if user $a \in U$ (source user) trusts user $b \in U$ (intermediate user) and user b trusts user $c \in U$ (target user), we can deduce using the principle of propagation in the trust matrix that user a can assign a trust score to user c using the (11),

$$Ptrust_{a \rightarrow c} = \frac{\sum_{b \in adj(a)} (|I_{a,b}| \times Trust^U(a \rightarrow b) + |I_{b,c}| \times Trust^U(b \rightarrow c))}{\sum_{b \in adj(a)} (|I_{a,b}| + |I_{b,c}|)} \quad (11)$$

- $Ptrust_{a \rightarrow c}$: The propagated trust value of user "a" towards user "c" inferred from their ratings (or check-ins).
- $a, c \in U$: The set of users.
- $b \in adj(a)$: The set of neighboring users of user "a" who trust user "c".
- $Trust^U(a \rightarrow b) \in [0, 1]$: The implicit trust value of user "a" towards user "b" inferred from their ratings (or check-ins).
- $Trust^U(b \rightarrow c) \in [0, 1]$: The implicit trust value of user "b" towards user "c" inferred from their ratings (or check-ins).
- $|I_{a,b}|$: The number of items that have been rated or visited by both user "a" and user "b".
- $|I_{b,c}|$: The number of items that have been rated or visited by both user "b" and user "c".

3. PROPOSED MODEL

This section provides a detailed overview of the POI recommendation approach proposed by the model named PRCT, which is based on the propagation of implicit trust inferred from POI ratings and user check-ins. This model is based on two approaches: the first approach, referred to as similarity trust rating (STR), utilizes similarity based on the trust derived from the ratings in the user-POI rating matrix (UPRM) and the second approach, denoted similarity trust check-in (STC), utilizes similarity based on the trust derived from the check-in matrix, referred to as UPCM. Then, these two approaches allow for the calculation of the user/user trust matrix, referred to as trust derivation matrix (TDM). This matrix can employ the principle of trust propagation to calculate the matrix denoted as trust prediction matrix (TPM), which contains the predicted ratings of the POIs.

3.1. The proposed algorithms

After explaining how to calculate the trust between users, referred to as TDM, based on their check-ins and ratings of POIs, as well as how to use propagation to enhance this matrix to mitigate data sparsity issues, we propose the Algorithm 1 to implement these calculations in this subsection.

Algorithm 1. User-user trust computation

Rating Based INPUT: UPRM: User-POI rating matrix OUTPUT: TDMR: Trust Derivation Matrix based on Rating RPMR: Rating Prediction Matrix based on Rating Var: PR, distance, Correct $\leftarrow$ empty lists <pre> 1 BEGIN 2 For each user b in UPRM Do 3 For each user a in UPRM AND $a \neq b$ Do 4 For each POI i in UPRM Do 5 //Compute predict rating $PR(a,b,i)$ using (2) 6 $PR(a,b,i) \leftarrow \text{meanRate}(a) + \text{Rate}(b,i) - \text{meanRate}(b)$ 7 //Compute $\text{Correct}(a,b,i)$ function using (3) 8 $\text{distance}(a,b,i) \leftarrow \text{Rate}(a,i) - PR(a,b,i)$ 9 IF ($\text{distance}(a,b,i) < \epsilon$) THEN 10 $\text{Correct}(a,b,i) \leftarrow 1$ 11 ELSE 12 $\text{Correct}(a,b,i) \leftarrow 0$ 13 END IF 14 //the set of user b's recommendations using (4) 15 $\text{RecSet}(b) \leftarrow \sum(\text{Correct}(a,b,i))$ 16 //the set of user b's correct recommendations using (5) 17 $\text{CorrectSet}(b) \leftarrow \sum(\text{Correct}(a,b,i)) \text{Correct}(a,b,i) = 1$ 18 END FOR 19 END FOR 20 //Compute user-user trust $\text{TDMR}(a,b)$ using (8) 21 $\text{TDMR}(a,b) \leftarrow \text{CorrectSet}(b) / \text{RecSet}(b)$ 22 END FOR 23 //Compute Rating Prediction (RPMR) based on rating trust (TDMR) using (10) 24 For each user a in UPRM Do 25 For each POI x in UPRM Do 26 IF ($\text{UPRM}(a,x) = \text{empty}$) THEN 27 numerator $\leftarrow 0$ 28 denominator $\leftarrow 0$ 29 For user b in UPRM Do 30 IF $b \neq a$ AND $\text{isNeighbor}(a,b)$ THEN 31 //b is a neighbor of a 32 numerator $\leftarrow$ numerator + ($\text{Rate}(b,x) - \text{meanRate}(b)$) * $\text{TDMR}(a,b)$ 33 denominator $\leftarrow$ denominator + $\text{TDMR}(a,b)$ 34 END IF 35 END FOR 36 IF denominator $\neq 0$ THEN 37 $\text{RPMR}(a,x) \leftarrow \text{meanRate}(a) + (\text{numerator} / \text{denominator})$ 38 ELSE 39 $\text{RPMR}(a,x) \leftarrow \text{meanRate}(a)$ 40 END IF 41 END IF 42 END FOR 43 END FOR 44 END </pre>	Check-in Based INPUT: UPCM: User-POI Check-in Matrix OUTPUT: TDMC: Trust Derivation Matrix based on Check-in RPMC: Rating Prediction Matrix based on Check-in Var: PC, distance, Correct $\leftarrow$ empty lists <pre> 1 BEGIN 2 For each user b in UPCM Do 3 For each user a in UPCM AND $a \neq b$ Do 4 For each POI i in UPCM Do 5 //Compute predict check-in $PC(a,b,i)$ using (2) 6 $PC(a,b,i) \leftarrow \text{meanCheck}(a) + \text{Check}(b,i) - \text{meanCheck}(b)$ 7 //Compute $\text{Correct}(a,b,i)$ function using (3) 8 $\text{distance}(a,b,i) \leftarrow \text{Check}(a,i) - PC(a,b,i)$ 9 IF ($\text{distance}(a,b,i) = 0$) THEN 10 $\text{Correct}(a,b,i) \leftarrow 1$ 11 ELSE 12 $\text{Correct}(a,b,i) \leftarrow 0$ 13 END IF 14 //the set of user b's recommendations using (4) 15 $\text{RecSet}(b) \leftarrow \sum(\text{Correct}(a,b,i))$ 16 //the set of user b's correct recommendations using (5) 17 $\text{CorrectSet}(b) \leftarrow \sum(\text{Correct}(a,b,i)) \text{Correct}(a,b,i) = 1$ 18 END FOR 19 END FOR 20 //Compute user-user trust $\text{TDMC}(a,b)$ using (8) 21 $\text{TDMC}(a,b) \leftarrow \text{CorrectSet}(b) / \text{RecSet}(b)$ 22 END FOR 23 //Compute Rating Prediction (RPMC) based on check-in trust (TDMC) using (10) 24 For each user a in UPCM Do 25 For each POI x in UPCM Do 26 IF ($\text{UPCM}(a,x) = \text{empty}$) THEN 27 numerator $\leftarrow 0$ 28 denominator $\leftarrow 0$ 29 For user b in UPCM Do 30 IF $b \neq a$ AND $\text{isNeighbor}(a,b)$ THEN 31 //b is a neighbor of a 32 numerator = numerator + ($\text{Rate}(b,x) - \text{meanRate}(b)$) * $\text{TDMC}(a,b)$ 33 denominator = denominator + $\text{TDMC}(a,b)$ 34 END IF 35 END FOR 36 IF denominator $\neq 0$ THEN 37 $\text{RPMC}(a,x) \leftarrow \text{meanRate}(a) + \frac{\text{numerator}}{\text{denominator}}$ 38 ELSE 39 $\text{RPMC}(a,x) \leftarrow \text{meanRate}(a)$ 40 END IF 41 END IF 42 END FOR 43 END FOR 44 END </pre>
--	--

The first algorithm, denoted user-user trust computation (UUTC), integrates the STR approach and the STC approach to calculate similarities based on ratings and check-ins. For this reason, the UUTC algorithm takes as input the UPRM of size $n \times m$ (where n is the number of users and m is the number of POIs) to compute the TDM based on ratings (TDMR) of size $n \times n$ (where n is the number of users). This same Algorithm 2 uses the user-POI check-in matrix (UPCM) of size $n \times m$ (where n is the number of users and m is the number of POIs) as input to compute the TDM based on check-ins (TDMC) of size $n \times n$.

After filling the two derivation matrices, TDMR and TDMC, this algorithm can calculate the prediction matrices of size $n \times m$ (where n is the number of users and m is the number of POIs), referred to as

rating prediction matrix based on ratings (RPMR) and rating prediction matrix based on check-ins (RPMC), as indicated in Algorithm 1 above.

To mitigate cold start issues and data sparsity, the PRCT model can utilize the user-user trust propagation (UUTP) algorithm to enhance the TDMR and TDMC matrices using the principle of propagation. This algorithm increases the density of TDMR and TDMC matrices by leveraging indirect trust links between users to obtain the propagated trust matrix based on ratings (PTMR) matrix of size $n \times n$ (where n is the number of users) and propagated trust matrix based on check-ins (PTMC) matrix of size $n \times n$ (where n is the number of users), as indicated in Algorithm 2.

Algorithm 2. User-User Trust Propagation

	Rating Based	Check-in Based
INPUT	TDMR: Trust Derivation Matrix based on Rating	TDMC: Trust Derivation Matrix based on Check-in
OUTPUT	PTMR: Propagated Trust Matrix based on Rating	PTMC: Propagated Trust Matrix based on Check-in
BEGIN	//Compute user-user trust propagation with one intermediate user using (11)	
2	For each user a in TDMR Do	For each user a in TDMC Do
3	For each user c in TDMR Do	For each user c in TDMC Do
4	IF (TDMR(a, c) = empty) THEN	IF (TDMC(a, c) = empty) THEN
	//Search for all possible intermediate users b	
5	For each user b in TDMR Do	For each user b in TDMC Do
6	IF (TDMR(a, b) AND TDMR(b, c)) THEN	IF (TDMC(a, b) AND TDMC(b, c)) THEN
7	$PTMR(a, c) \leftarrow (commonRate(a, b) * TDMR(a, b) + commonRate(b, c) * TDMR(b, c)) / (commonRate(a, b) + commonRate(b, c))$	$PTMC(a, c) \leftarrow (commonCheck(a, b) * TDMC(a, b) + commonCheck(b, c) * TDMC(b, c)) / (commonCheck(a, b) + commonCheck(b, c))$
8	END IF	END IF
9	END FOR	END FOR
	//If there are multiple intermediate users b , perform aggregation	
10	IF (length(PTMR(a, c)) > 1) THEN	IF (length(PTMC(a, c)) > 1) THEN
11	$PTMR(a, c) \leftarrow mean(PTMR(a, c))$	$PTMC(a, c) \leftarrow mean(PTMC(a, c))$
12	END IF	END IF
13	END FOR	END FOR
14	END FOR	END FOR
END		

3.2. Functioning of the PRCT model

Figure 1 illustrates how the PRCT model works, which is based on four main steps, summarized as follows,

- After filtering the **review.js** file from the Yelp dataset to extract only the ratings and check-ins, the UUTC algorithm calculates the trust between users derived from (1) ratings, referred to as STR, and (2) the similarity inferred from check-ins, referred to as STC.
- The PRCT model can ignore the propagation principle and proceed directly to the prediction calculations. This process is performed without using the UUTP algorithm.
- The PRCT model can adopt the propagation principle (two hops) to further enrich the content of the similarity matrices between users. These matrices are calculated using the UUTP algorithm.
- Calculate predictions based on the similarity matrices between users derived from the propagation principle.

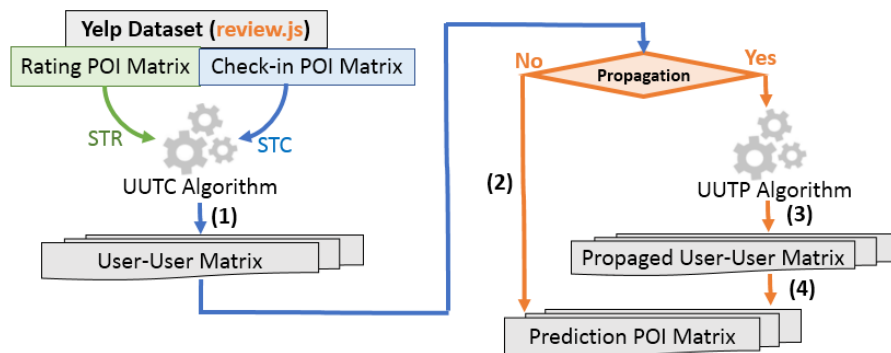


Figure 1. Functional description of the PRCT model

3.3. Evaluation of the PRCT model

In Figure 2, we explain how to evaluate the PRCT model by comparing these two approaches (STR and STC) with other types of approaches from the literature, such as (1) Pearson similarity [25], (2) Jaccard similarity [26], (3) Cosine similarity [27], and (4) trust-based similarity defined by O'Donovan [20].

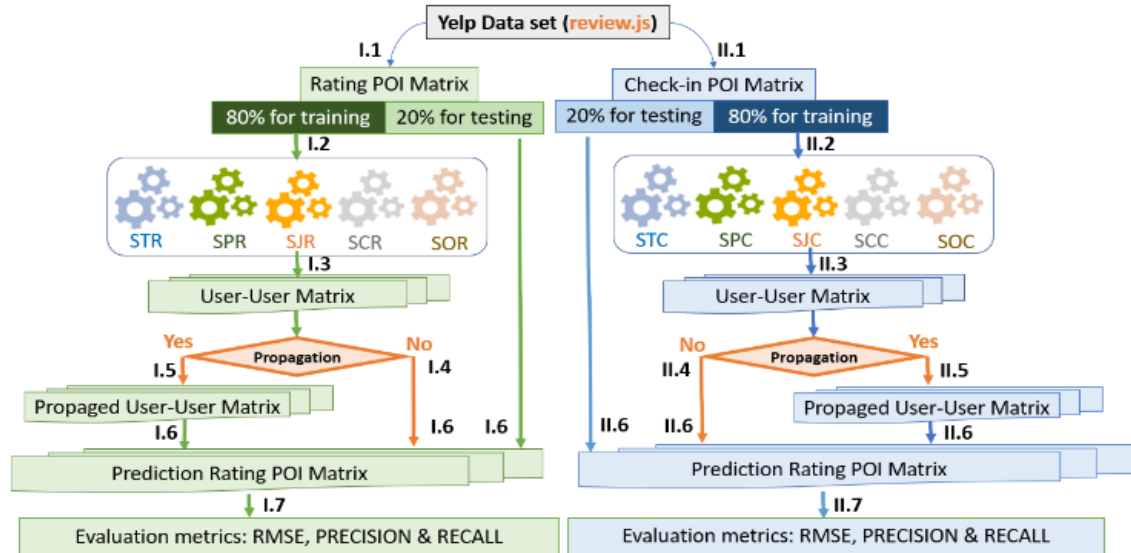


Figure 2. Description of the evaluation process for the PRCT model

Figure 2 consists of two parts: the first part addresses ratings and comprises 7 main steps (from I.1 to I.7 in Figure 2), while the second part focuses on the utilization of check-ins and also includes 7 main steps (from II.1 to II.7 in Figure 2). Each step of this model is described using the data it manipulates and the algorithms it employs. These steps can be divided into two categories as follows,

Part I: Evaluation of our STR approach based on ratings,

I.1. Filter the Yelp dataset to extract only the ratings made by users on the POIs, and then split this dataset into two parts: 80% for model training and 20% for experimental testing.

I.2. Introduce the training part (80% of the Yelp dataset of ratings made by users on the POIs) as input for calculating the similarities between users using the STR approach.

I.3. Calculate the similarities between users using five types of similarities: (1) STR similarity, (2) Pearson similarity derived from ratings, noted similarity Pearson rating (SPR), (3) Jaccard similarity from ratings, noted similarity Jaccard rating (SJR), (4) Cosine similarity from ratings, noted similarity Cosine rating (SCR), and (5) trust-based similarity defined by O'Donovan from ratings, noted similarity O'Donovan rating (SOR).

I.4. Ignore the propagation principle.

I.5. Use the propagation principle (two hops) to further enrich the content of the similarity matrices between users.

I.6. Calculate predictions from the 20% of the dataset reserved for testing using the similarity matrices between users, obtained through STR, SPR, SJR, SCR, and SOR, which incorporate the similarity propagation principle with or without hops.

I.7. Evaluate the predictions calculated in I.6 using the RMSE and Precision/Recall parameters.

Part II: Evaluation of our STC approach based on check-ins,

II.1. Filter the Yelp dataset to extract only the check-ins made by users at POIs, then split this dataset into two parts: 80% for model training and 20% for experimental testing.

II.2. Introduce the training part (80% of the Yelp dataset of check-ins made by users on the POIs) as input for calculating the similarities between users using the STC approach.

II.3. Calculate the similarities between users using five types of similarities: (1) STC similarity, (2) Pearson similarity derived from check-ins, denoted as similarity Pearson check-in (SPC), (3) Jaccard similarity from check-ins, denoted as similarity Jaccard check-in (SJC), (4) Cosine similarity from check-ins, denoted as similarity cosine check-in (SCC), and (5) the trust-based similarity defined by O'Donovan, which we have adapted to the context of check-ins (see Algorithm 3), denoted as similarity O'Donovan check-in (SOC).

II.4. Ignore the principle of propagation.

II.5. Use the principle of propagation (two hops) to further enrich the content of the similarity matrices between users.

II.6. Calculate predictions from the 20% of the dataset reserved for testing using the similarity matrices between users, obtained via STC, SPC, SJC, SCC, and SOC, which incorporate the principle of similarity propagation with or without hops.

II.7. Evaluate the predictions calculated in II.6 using the RMSE and Precision/Recall parameters.

Algorithm 3. O'donovan trust computation adapted for check-in data

INPUT: UPCM: User-POI check-in matrix
OUTPUT: TProfile: Trust profile matrix based on check-in
Var: PC, distance, Correct $\leftarrow$ empty lists

```

1  BEGIN
2  For each user  $b$  in UPCM Do
3  For each user  $a$  in UPCM AND  $a \neq b$  Do
4  For each POI  $i$  in UPCM Do
    //Compute predict check-in  $PC(a,b,i)$  using (2)
5   $PC(a,b,i) \leftarrow \text{meanCheck}(a) + \text{Check}(b,i) - \text{meanCheck}(b)$ 
    //Compute  $\text{Correct}(a,b,i)$  function using (3)
6   $\text{distance}(a,b,i) \leftarrow | \text{Check}(a,i) - PC(a,b,i) |$ 
7  IF ( $\text{distance}(a,b,i) = 0$ ) THEN
8   $\text{Correct}(a,b,i) \leftarrow 1$ 
9  ELSE
10  $\text{Correct}(a,b,i) \leftarrow 0$ 
11 END IF
    //the set of user  $b$ 's recommendations using (4)
12  $\text{RecSet}(b) \leftarrow \sum(\text{Correct}(a,b,i))$ 
    //the set of user  $b$ 's correct recommendations using (5)
13  $\text{CorrectSet}(b) \leftarrow \sum(\text{Correct}(a,b,i) | \text{Correct}(a,b,i) = 1)$ 
14 END FOR
15 END FOR
    //Compute Profile-Level Trust  $TProfile(a,b)$  using (6)
16  $TProfile(a,b) \leftarrow \text{CorrectSet}(b) / \text{RecSet}(b);$ 
17 END FOR
18 END
```

To calculate the parameters (RMSE, Precision, And Recall) for comparing the PRCT model (STR and STC) with other approaches (Jaccard, Cosine, PCC, and O'Donovan), we used the Yelp dataset described in Table 1 and a series of hyperparameters defined in Table 2. This dataset contains user interactions with the POIs through ratings and check-ins, and these hyperparameters concern the settings to be adopted for all comparisons made in section 4.

To assess the performance of the PRCT model, we employed the previously described dataset and hyperparameters (see Table 1 and Table 2), using RMSE, Precision, and Recall as evaluation metrics. First, RMSE was applied to measure the accuracy of rating predictions produced by the two PRCT variants-STR and STC-and to compare their performance with existing POI recommendation methods in the literature. Then, Precision and Recall were used to gain deeper insights into the quality of the recommendations generated by both variants. To compute these metrics, we loaded the Yelp review dataset, extracted key attributes (user ID, business ID, star ratings, and comments), and filtered the data based on specific user and rating criteria. Following preprocessing, the dataset was split into training and test sets, and trust matrices were constructed from both rating and check-in information (see Figure 3).

Table 1. Description of data set Yelp

Field	Value	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..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 2. List of the PRCT hyperparameters

Hyperparameter	Value	Explanation
ε	0.9	The trust threshold
n_{hop}	2	The number of hops (propagation)
m_{inter}	1	The number of intermediate users (propagation)
T_{train}	80%	The train subset
T_{test}	20%	The test subset

```

json_file_path = "data/yelp_academic_dataset_review.json"
# Define parameters
NUM_USERS_TO_SELECT = 1000 # Number of users to select
MIN_NUM_RATINGS = 3 # Minimum number of ratings per user
MIN_COMMON_RATED_ITEMS = 2 # Minimum number of common rated items

# Read the JSON file
data = []

with open(json_file_path, "r", encoding="utf-8") as json:
    for line in json:
        review = json.loads(line)
        selected_review = {
            "user_id": review["user_id"],
            "business_id": review["business_id"],
            "stars": review["stars"],
            "useful": review["useful"],
            "funny": review["funny"],
            "cool": review["cool"]
        }
        data.append(selected_review)

# New Dataframe for : user-user-trust
df_user_trustC = pd.DataFrame(columns=['truster', 'trustee', 'trust'])
for Cusertr in Cindex_prediction:
    CRecSet_sum = 0
    cCorrectSet_sum = 0
    c_ui = initial_checkin_prediction_df.loc[Cusertr, 'User Actif']
    c_uj = initial_checkin_prediction_df.loc[Cusertr, 'User Predict']
    for Cusertr in Cindex_prediction:
        truster = initial_checkin_prediction_df.loc[Cusertr, 'User Actif']
        trustee = initial_checkin_prediction_df.loc[Cusertr, 'User Predict']
        if truster == c_ui and trustee == c_uj:
            CRecSet_sum += 1
            if initial_checkin_prediction_df.loc[Cusertr, 'Correct'] == 1:
                cCorrectSet_sum += 1
    user_trustC = cCorrectSet_sum / CRecSet_sum
    cadd_row = pd.DataFrame({'truster': [c_ui], 'trustee': [c_uj], 'trust': [user_trustC]})
    # append the new row to the dataframe
    df_user_trustC = pd.concat([df_user_trustC, cadd_row], ignore_index=True)

```

Figure 3. Preprocessing, feature selection, and trust matrix computation from the yelp review dataset using ratings and check-ins

Once the trust matrices are computed, we proceed to predict user ratings, followed by an evaluation of the results using RMSE, Precision, and Recall (see Figure 4).

4. RESULTS AND DISCUSSION

In this section, a comparison between the variants of the PRCT model is made using the Yelp dataset divided into two parts: 80% for training and 20% for testing. Then, the STR, STC, SOR, and SOC algorithms use this training set to construct the trust matrices of the PRCT model and predict the ratings of the POIs in the testing set. The performance of this model is evaluated using the RMSE [28] and Precision [29]/Recall [30] metrics to estimate the accuracy of the POI recommendations. The PRCT model is then compared to other POI recommendation models using different similarities such as Jaccard, Cosine, and Pearson. Finally, a study on the impact of propagation on sparsity [31] and the quality of recommendations is conducted on the trust matrices of the PRCT model and the prediction matrices.

```

def rating_prediction(test_rate_dataset, trust_matrix):
    # create an empty dataframe
    df_predictions_2 = pd.DataFrame(columns=['UserActif', 'POItoPredict', 'Prediction'])
    sumOnly_2 = 0
    sumRate_2 = 0
    for i_2 in test_rate_dataset.index:
        for j_2 in test_rate_dataset.columns:
            if np.isnan(test_rate_dataset.loc[i_2, j_2]):
                user_a_2 = i_2 # aktif User
                item_x_2 = j_2 # POI to predict
                mean_user_a_2 = user_mean(test_rate_dataset, user_a_2) # Call mean function
                list_neighbors_2 = user_neighbor(test_rate_dataset, user_a_2, item_x_2) # Call find neighbors function
                if not list_neighbors_2.empty: # if there is one neighbor or more
                    for n_2 in list_neighbors_2.index:
                        neighbor_2 = list_neighbors_2.loc[n_2, 'Neighbor']
                        rate_neighbor_2 = list_neighbors_2.loc[n_2, 'Rate']
                        # The mean of the user neighbor
                        mean_user_neighbor_2 = user_mean(test_rate_dataset, neighbor_2)
                        rate_mean_2 = rate_neighbor_2 - mean_user_neighbor_2
                        # Search of the trust User-User Value between (c,p)
                        sim_2 = sim_value(trust_matrix, user_a_2, neighbor_2)
                        if sim_2 != 'vide' and sim_2 != 0 and sim_2 is not None:
                            sumOnly_2 += abs(sim_2)
                            ss_2 = rate_mean_2 * sim_2
                            sumRate_2 += ss_2
                    if sumOnly_2 != 0:
                        prediction_2 = mean_user_a_2 + (sumRate_2 / sumOnly_2)

rmse = calculate_rmse(prediction_trust_rate, test_matrix_rates)
mae = calculate_mae(prediction_trust_rate, test_matrix_rates)
precision = precision_pivot(prediction_trust_rate, test_matrix_rates,
                             threshold, threshold_recommend)
recall = recall_pivot(prediction_trust_rate, test_matrix_rates,
                      threshold_predict_item, threshold_relevant_item)

```

Figure 4. Rating prediction and evaluation based on trust matrices

4.1. Comparison between the variants of the PRCT model

In this subsection, we compare the different variants of the PRCT model, such as the STR/STC approaches with or without propagation, and the two approaches SOR and SOC. Table 3 illustrates that RSs based on trust inferred from user ratings generally achieve better performance in terms of RMSE, Precision, and Recall compared to those relying on trust derived from check-ins. However, the application of trust propagation, while helpful in addressing sparsity issues, appears to negatively impact the overall accuracy of the recommendations.

In Table 3, we observe that the prediction algorithm based on ratings, called STR, outperforms the STC, STR algorithms with propagation, and STC without propagation in terms of Precision and Recall. Additionally, it is noteworthy that the STC and STR approaches (with or without propagation) of the PRCT model outperform the traditional approach of O'Donovan, which also utilizes ratings and check-ins, in terms of Precision and Recall.

Table 3. Comparison between the variants of the PRCT model using Avg Precision, Avg Recall, and Avg F1 (Avg = Average)

Approach	Avg Precision	Avg Recall	Avg F1 Score
STR	0,91542	0,65369	0,76273
STC	0,91332	0,65040	0,75976
STR with propagation	0,91098	0,64097	0,75249
STC with propagation	0,90827	0,63913	0,75029
SOR	0,90827	0,63913	0,75029
SOC	0,90827	0,63913	0,75029

4.2. Comparison without propagation between the PRCT model and other types of user-user similarities

In this subsection, we compared the STR approach of the PRCT model with the other approaches SPR, SCR, SJR, and SOR without incorporating the principle of propagation. Next, we also conducted a comparison between the STC approach of the PRCT model and the approaches SPC, SCC, SJC, and SOC without incorporating the principle of propagation. Finally, we summarized the results obtained from these comparisons.

4.2.1. Comparison between STR and the approaches SPR, SCR, SJR, and SOR

In Table 4, we observe that the SPR approach outperforms all rating-based approaches in terms of RMSE and Recall; however, the STR approach of the PRCT model appears to be better in terms of Precision. Table 4 supports our claim that the proposed trust-based approach (STR) outperforms the similarity-based CF method in terms of accuracy, as evidenced by higher Precision and Recall values.

Table 4. Comparison between STR and the SPR, SCR, SJR, and SOR approaches using Avg RMSE, Avg Precision, and Avg Recall

Type of Similarity	Avg. RMSE	Avg. Precision	Avg. Recall
STR	0,9440	0,9154	0,6537
SPR	0,8866	0,9131	0,6577
SCR	0,9170	0,9133	0,6504
SJR	0,9234	0,9118	0,6482
SOR	0,9420	0,9083	0,6391

4.2.2. Comparison between STC and the approaches SPC, SCC, SJC, and SOC

In Table 5, we observe that the SPC approach outperforms all approaches based on check-ins in terms of RMSE and Recall; however, the SCC approach appears to be better in terms of Precision. Table 5 demonstrates that the proposed check-in-based approach (STC) outperforms the profile-level methods inspired by O'Donovan in terms of both Precision and Recall.

4.2.3. Summary of comparison between the PRCT model and other types of similarities

In Table 6, we observe that overall, the approaches based on check-ins demonstrate better performance than those based on ratings, in terms of RMSE, Precision, and Recall.

4.3. Comparison with propagation between the PRCT model and other types of user-user similarities

In this subsection, we examined the impact of propagation on the PRCT model and other approaches such as SPR, SCR, SJR, SOR, SPC, SCC, SJC, and SOC. We then conducted a comparison of these approaches in terms of RMSE, Precision, and Recall. Finally, we summarized the results obtained from these comparisons.

Table 5. Comparison between STC and the SPC, SCC, SJC, and SOC approaches using Avg RMSE, Avg Precision, and Avg Recall

Type of Similarity	Avg. RMSE	Avg. Precision	Avg. Recall
STC	0,9465	0,9133	0,6504
SPC	0,8256	0,9193	0,6578
SCC	0,9212	0,9200	0,6528
SJC	0,9263	0,9180	0,6493
SOC	0,9428	0,9083	0,6391

Table 6. Summary comparison between the PRCT model and other types of similarities using Avg Rmse, Avg Precision, and Avg Recall

Type of Similarity	Avg RMSE	Avg Precision	Avg Recall
STR	0,9440	0,9154	0,6537
SPR	0,8866	0,9131	0,6577
SCR	0,9170	0,9133	0,6504
SJR	0,9234	0,9118	0,6482
SOR	0,9420	0,9083	0,6391
STC	0,9465	0,9133	0,6504
SPC	0,8256	0,9193	0,6578
SCC	0,9212	0,9200	0,6528
SJC	0,9263	0,9180	0,6493
SOC	0,9428	0,9083	0,6391

4.3.1. Study of the impact of propagation on sparsity

In Table 7, we observe that the "SCR with propagation" and "SJR with propagation" approaches display the highest density levels for both the similarity matrices and the prediction matrices. There are two types of propagation: one based on ratings and the other on check-ins. It is observed that trust propagation is more effective with check-ins (STC) than with similarity-based approaches (SPC, SCC, SJC). On the other hand, in the case of ratings, propagation methods based on similarity (SCR and SJR) outperform the trust-based approach (STR).

Table 7. Effect of propagation on the sparsity of similarity and prediction matrices

Type of Similarity	AVG Sim Matrix Sparsity	AVG Prediction Sparsity
STR with propagation	16,2218	19,2890
SPR with propagation	25,8464	45,0272
SCR with propagation	15,0845	8,4727
SJR with propagation	15,0845	8,4727
STC with propagation	16,1745	14,3736
SPC with propagation	36,2918	79,2718
SCC with propagation	21,8464	22,6063
SJC with propagation	24,4845	13,1672

4.3.2. Comparison of approaches with and without propagation

In Table 8, we observe that, generally, the approaches with propagation decrease the performance of the RS in terms of RMSE, Precision, and Recall. However, some approaches, such as SCC and SJC, prove to be more robust than others. For example, propagation has no impact on the SCC approach in terms of RMSE, Precision, and Recall, while the SJC approach remains stable in terms of Recall.

4.3.3. Summary of comparisons

In Table 9, we observe that propagation in check-in-based approaches provides better performance than in rating-based approaches, in terms of RMSE, Precision, and Recall. However, propagation in rating-based approaches results in better outcomes regarding the sparsity of the similarity and prediction matrices compared to check-in-based methods.

Table 8. Effect of propagation on Avg RMSE, Avg Precision, and Avg Recall

Type of Similarity	Avg RMSE	Avg Precision	Avg Recall
STR	0,9440	0,9154	0,6537
STC	0,9465	0,9133	0,6504
SPR	0,8866	0,9131	0,6577
SCR	0,9170	0,9133	0,6504
SJR	0,9234	0,9118	0,6482
SPC	0,8256	0,9193	0,6578
SCC	0,9212	0,9200	0,6528
SJC	0,9263	0,9180	0,6493
AVG without propagation	0,9113	0,9155	0,6525
STR with propagation	0,9751	0,9110	0,6410
STC with propagation	0,9737	0,9083	0,6391
SPR with propagation	0,9199	0,9086	0,6500
SCR with propagation	0,9433	0,9083	0,6391
SJR with propagation	0,9496	0,9074	0,6405
SPC with propagation	0,8301	0,9162	0,6516
SCC with propagation	0,9212	0,9200	0,6528
SJC with propagation	0,9490	0,9106	0,6493
AVG with propagation	0,9327	0,9113	0,6454

Table 9. Comparison between rating-based approaches and check-in-based approaches

Type of Similarity	Avg RMSE	Avg Precision	Avg Recall	Avg Sim Matrix Sparsity	Avg Prediction Sparsity
STR with propagation	0,9751	0,9109	0,6410	16,2218	19,2890
SPR with propagation	0,9199	0,9086	0,6500	25,8464	45,0272
SCR with propagation	0,9433	0,9083	0,6391	15,0845	8,4727
SJR with propagation	0,9496	0,9074	0,6405	15,0845	8,4727
AVG_STR,SPR,SCR,SJR	0,9469	0,9088	0,6426	16,1745	20,3154
STC with propagation	0,9737	0,9082	0,6391	16,1745	14,3736
SPC with propagation	0,8301	0,9162	0,6516	36,2918	79,2718
SCC with propagation	0,9212	0,9200	0,6528	21,8464	22,6063
SJC with propagation	0,9490	0,9106	0,6493	24,4845	13,1672
AVG_STC,SPC,SCC,SJC	0,9185	0,91376	0,6482	24,6993	32,3547

4.4. Global summary and discussion of results

In summary, the approaches based on check-ins provide superior performance in terms of RMSE, Precision, and Recall compared to those based on ratings. However, while propagation enhances the sparsity of the matrices, it does not always lead to significant improvements in terms of Precision and Recall. Approaches like SCC and SJC stand out for their robustness and stability in the face of propagation, making them particularly valuable in certain use cases.

The proposed approach is particularly well-suited for the context of exploring a new city, especially when the user and point of interest (POI) database is still being developed. However, several important limitations must be considered. First, ratings and check-ins can be deliberately biased by certain users, which may affect the reliability of the inferred trust. Second, leveraging check-in history requires permission to share personal data, raising privacy concerns. Finally, the approach involves a certain level of computational complexity, particularly related to the dynamic updating of trust matrices when a new user or POI is added to the system. This may impact performance in rapidly changing environments.

5. CONCLUSION

In this paper, we proposed and studied the model PRCT to enhance POI recommendations in LBSNs by leveraging the implicit trust between users, inferred from their interactions with POIs through ratings and check-ins. Our main objective was to explore the impact of trust propagation on the accuracy and quality of recommendations, particularly in scenarios characterized by sparse similarity matrices. The results obtained from the Yelp dataset indicate that the rating-based approach (STR) of the PRCT model generally outperforms other variants in terms of Precision and Recall. However, the check-in-based approach of the PRCT model (STC) demonstrate particularly strong performance in terms of RMSE, providing greater robustness in the process of recommending POIs. Moreover, while trust propagation enhances the density of similarity matrices, it does not always lead to significant improvements (Precision parameter), except in certain cases where it is well-managed, such as with the SCC and SJC approaches. Overall, this article demonstrates the effectiveness of trust propagation in enriching similarity matrices, mitigating data sparsity and improving recommendation reliability. However, it is crucial to carefully adjust the propagation methods and select the most suitable approaches to maximize recommendation performance.

Future work could focus on optimizing these parameters as well as may explore hybrid approaches that integrate various similarity measures based on user interactions, such as ratings and check-ins, while fusing different trust sources to better capture user intent. Additionally, incorporating richer contextual information: such as temporal dynamics, user profiles, or real-time behavioral signals: could further personalize recommendations and increase their relevance in dynamic environments like LBSNs.

In practical applications, this work could benefit POI-based services in tourism, local business discovery, and smart city services, where trust and context-aware recommendations play a pivotal role in enhancing user satisfaction. Moreover, in scenarios like sparse trust matrices or cold-start problems, trust can be effectively inferred from user ratings or check-ins, enabling the generation of more personalized and accurate recommendations. Our findings confirm that adaptive trust propagation strategies can improve recommendation robustness and help mitigate the challenges posed by data sparsity.

FUNDING INFORMATION

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.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sara Medjroud	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Nassim Dennouni	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓		✓
Mourad Loukam			✓				✓			✓		✓	✓	

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nterpretation

R : **R**esources

D : **D**ata Curation

O : **O**riginal Draft

E : **E**xperimentation

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Visualization The data that support the findings of this study are available from the corresponding author, [Sara MEDJROUD], upon reasonable request.

REFERENCES

- [1] X. Zhou, J. Tian, J. Peng, and M. Su, "A smart tourism recommendation algorithm based on cellular geospatial clustering and multivariate weighted collaborative filtering," *ISPRS International Journal of Geo-Information*, vol. 10, no. 9, 2021, doi: 10.3390/ijgi10090628.
- [2] B. Xiao and I. Benbasat, "An empirical examination of the influence of biased personalized product recommendations on consumers' decision making outcomes," *Decision Support Systems*, vol. 110, pp. 46–57, 2018, doi: 10.1016/j.dss.2018.03.005.
- [3] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 397–405, 2009, doi: 10.1145/1557019.1557067.
- [4] G. Guo, J. Zhang, and D. Thalmann, "A simple but effective method to incorporate trusted neighbors in recommender systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7379 LNCS, pp. 114–125, 2012, doi: 10.1007/978-3-642-31454-4_10.
- [5] L. Yang and X. Niu, "A genre trust model for defending shilling attacks in recommender systems," *Complex and Intelligent Systems*, vol. 9, no. 3, pp. 2929–2942, 2023, doi: 10.1007/s40747-021-00357-2.
- [6] F. Zhang, H. Wang, and H. Yi, "An Adaptive Recommendation Method Based on Small-World Implicit Trust Network," *Journal of Computers*, vol. 9, no. 3, 2014, doi: 10.4304/jcp.9.3.618-625.
- [7] F. Roy, S. M. Sarwar, and M. Hasan, "User similarity computation for collaborative filtering using dynamic implicit trust," *Communications in Computer and Information Science*, vol. 542, pp. 224–235, 2015, doi: 10.1007/978-3-319-26123-2_22.
- [8] A. Zahir, Y. Yuan, and K. Moniz, "Agreereltrust-a simple implicit trust inference model for memory-based collaborative filtering recommendation systems," *Electronics (Switzerland)*, vol. 8, no. 4, 2019, doi: 10.3390/electronics8040427.

- [9] S. Zhao, I. King, and M. R. Lyu, "A survey of point-of-interest recommendation in location-based social networks," *ACM International Conference Proceeding Series*, pp. 185–192, Jul. 2016, doi: 10.1145/3428658.3430970.
- [10] 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, 2021, doi: 10.1016/j.elerap.2021.101060.
- [11] R. Logesh and V. Subramaniaswamy, "A Reliable Point of Interest Recommendation based on Trust Relevancy between Users," *Wireless Personal Communications*, vol. 97, no. 2, pp. 2751–2780, 2017, doi: 10.1007/s11277-017-4633-1.
- [12] J. Zhu, Q. Ming, and Y. Liu, "Trust-distrust-aware point-of-interest recommendation in location-based social network," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10874 LNCS, pp. 709–719, 2018, doi: 10.1007/978-3-319-94268-1_58.
- [13] 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, no. 1, 2019, doi: 10.1186/s13638-019-1388-2.
- [14] 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, 2020, doi: 10.1109/TII.2019.2958696.
- [15] K. Rmoku, B. Selimi, and L. Ahmedi, "Application of Trust in Recommender Systems—Utilizing Naive Bayes Classifier," *Computation*, vol. 10, no. 1, 2022, doi: 10.3390/computation10010006.
- [16] J. An, W. Jiang, and G. Li, "Bidirectional Trust-Enhanced Collaborative Filtering for Point-of-Interest Recommendation," *Sensors*, vol. 23, no. 8, 2023, doi: 10.3390/s23084140.
- [17] 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, doi: 10.1016/j.procs.2020.11.017.
- [18] Q. Shambour and J. Lu, "An effective recommender system by unifying user and item trust information for B2B applications," *Journal of Computer and System Sciences*, vol. 81, no. 7, pp. 1110–1126, 2015, doi: 10.1016/j.jcss.2014.12.029.
- [19] H. Zhao, Y. Zhang, and Y. Xiao, "A new collaborative filtering algorithm with combination of explicit trust and implicit trust," in *13th International Conference on Computer Science and Education, ICCSE 2018*, 2018, pp. 319–323, doi: 10.1109/ICCSE.2018.8468763.
- [20] J. O'Donovan and B. Smyth, "Trust in recommender systems," *International Conference on Intelligent User Interfaces, Proceedings IUI*, pp. 167–174, 2005, doi: 10.1145/1040830.1040870.
- [21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994*, pp. 175–186, 1994, doi: 10.1145/192844.192905.
- [22] C. S. Hwang and Y. P. Chen, "Using trust in collaborative filtering recommendation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4570 LNAI, pp. 1052–1060, 2007, doi: 10.1007/978-3-540-73325-6_105.
- [23] 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, doi: 10.1002/int.20495.
- [24] S. Medjrout, N. Dennouni, M. Hadj Henni, and D. Bettache, "Towards a new POI Recommendation Approach based on Implicit Trust between users," *Proceedings of the 2022 1st International Conference on Big Data, IoT, Web Intelligence and Applications, BIWA 2022*, pp. 19–24, 2022, doi: 10.1109/BIWA57631.2022.10038190.
- [25] S. Aygun and S. Okyay, "Improving the pearson similarity equation for recommender systems by age parameter," 2015, doi: 10.1109/AIEEE.2015.7367282.
- [26] 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, doi: 10.1016/j.ins.2019.01.023.
- [27] 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, doi: 10.35940/ijeat.e9666.069520.
- [28] M. Jahrer, A. Töschner, and R. Legenstein, "Combining predictions for accurate recommender systems," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 693–701, 2010, doi: 10.1145/1835804.1835893.
- [29] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 39–46, 2010, doi: 10.1145/1864708.1864721.
- [30] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2019, doi: 10.1007/s13042-017-0762-9.
- [31] D. Anand and K. K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5101–5109, 2011, doi: 10.1016/j.eswa.2010.09.141.

BIOGRAPHIES OF AUTHORS



Sara Medjrout    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). She can be contacted at email: s.medjrout@univ-chlef.dz.



Dr. 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 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. He can be contacted at email: dennouninas@gmail.com.



Dr. 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. He can be contacted at email: m.loukam@univ-chlef.dz.