MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
HASSIBA BENBOUALI UNIVERSITY OF CHLEF (UHBC), Algeria
Faculty of Exact and Computer Sciences
Department of Computer Science

THESIS

Submitted for the diploma of

DOCTORAT

Field: Computer Science

Speciality: Information Systems

By

SARA MEDJROUD

Title:

POI recommendation based on social trust in LBSN

Defended on 03/07/2025, before the committee composed of:

ARIDJ Mohamed	MCA	University of Chlef	President
DENNOUNI Nassim	MCA	Higher School of Management - Tlemcen	Supervisor
LOUKAM Mourad	MCA	University of Chlef	Co-Supervisor
NAHNOUH Chakib	MCA	University of Chlef	Examiner
MEZZOUDJ Freha	MCA	National Polytechnic School - Oran	Examiner
LOUAZANI Ahmed	MCA	University of Blida1	Examiner

Dedication

To my family, whose unwavering support and encouragement have been my greatest source of strength.

To my advisors and mentors, Mr. Dennouni Nassim and Mr. Loukam Mourad, for their invaluable guidance, support, and belief in my work throughout this journey.

To my friends and colleagues, for their insightful discussions and unwavering belief in my work.

And to all those who seek to push the boundaries of knowledge may innovation and curiosity always guide your path.

Acknowledgments

*First and foremost, I would like to express my deepest gratitude to **ALLAH**, the Most Gracious and the Most Merciful, for granting me the strength, patience, and guidance throughout my academic journey and the completion of this thesis. Without His divine blessings, none of this would have been possible.*

I would like to express my deepest gratitude to my advisor, Mr. Dennouni Nassim, and my co-advisor, Mr. Loukam Mourad, for their unwavering support, guidance, and constructive feedback throughout my research. Their expertise and dedication have been instrumental in shaping this thesis. I would also like to express my sincere appreciation to the members of my thesis committee for their time and willingness to evaluate my thesis.

A heartfelt thank you to my family for their unconditional support, patience, and encouragement, which have given me the strength to persevere through challenges.

I am also grateful to my colleagues and friends, who have shared this academic journey with me, offering both intellectual stimulation and moments of much-needed relief.

Finally, I acknowledge the institutions, research centers, and funding bodies that have supported this work, providing the necessary resources and opportunities for its completion.

To everyone who has contributed to this journey, directly or indirectly, thank you.

Contents

List of Figures

List of Tables

List of Equations

List of Algorithms

List of Abbreviations and Acronyms

RS	Recommender System
SR	Système de Recommandation
POI	Point of Interest
LBSN	Location-Based Social Network
HRCT	Hybrid Rating Check-in Trust
PRCT	Propagation of Rating/Check-in for implicit Trust
ITCRC	Implicit Trust based on Combining point of interest Ratings and user Check-ins
CF	Collaborative Filtering
LSA	Latent Semantic Analysis
MF	Matrix Factorization
SVD	Singular Value Decomposition
SVD++	It is an enhanced version of the original SVD model
PMF	Probabilistic Matrix Factorization
ALS	Alternating Least Squares
NMF	Non-Negative Matrix Factorization
VAE	Variational Autoencoders
DNN	Deep Neural Networks
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks

LSTM	Long Short-Term Memory
GNN	Graph Neural Networks
NLP	Natural Language Processing
NDCG	Normalized Discounted Cumulative Gain
DCG	Discounted Cumulative Gain
IDCG	Ideal Discounted Cumulative Gain
MRR	Mean Reciprocal Rank
STP	Social Trust Precision
UIRM	User-Item Rating Matrix
ITBRS	Implicit Trust-Based Recommender Systems
AI	Artificial Intelligence
UT	User Trust
ACF	Automatic Collaborative Filtering
MAE	Mean Absolute Error
MAUE	Mean Absolute User Error
RMSE	Root Mean Square Error
PCC	Pearson Correlation Coefficient
kNN	k-Nearest Neighbors
kNR	k-Nearest Recommenders
TARS	Trust-aware Recommender System
MSD	Mean Squared Differeces
ROC	Receiver Operating Characteristic

SPTW	Social Pertinent Trust Walker
JMSD	Jaccard Mean Squared Difference
BiTCF	Bi-Directional Trust Enhanced Collaborative Filtering model
TDMR	Trust Derivation Matrix based on Ratings
TDMC	Trust Derivation Matrix based on Check-ins
H-Trust	Hybrid trust matrix
RPMR	Rating Prediction Matrix based on Ratings
UPRM	User-POI Ratings Matrix
UPCM	User-POI Check-ins Matrix
RPMC	Rating Prediction Matrix based on Check-ins
TDM	Trust Derivation Matrix
RPMH	Rating Prediction Matrix based on H-Trust
R-Trust	Trust deduced from Ratings
C-Trust	Trust deduced from Check-ins
H-Trust	Hybrid Trust deduced from Rating & Check-in
AVG	Average
Dev	Standard Deviation
RC	Rating or Check-in
STR	Similarity Trust Rating
STC	Similarity Trust Check-in
RPM	Rating Prediction Matrix
UUTC	User-User Trust Computation

UUTP	User-User Trust Propagation
PTMR	Propagated Trust Matrix based on Rating
PTMC	Propagated Trust Matrix based on Check-in
SPR	Similarity Pearson Rating
SJR	Similarity Jaccard Rating
SCR	Similarity Cosine Rating
SOR	Similarity O'Donovan Rating
SPC	Similarity Pearson Check-in
SJC	Similarity Jaccard Check-in
SCC	Similarity Cosine Check-in
SOC	Similarity O’Donovan Check-in
HTM	Hybrid Trust Matrix
TR	Trust based on Rating
TC	Trust based on Check-in
TH	Trust based on Hybrid
OR	O'Donovan trust based on Rating
OC	O'Donovan trust based on Check-in
OH	O'Donovan trust based on Hybridization of rating and check-in

Abstract

This thesis addresses the challenges of point-of-interest (POI) recommendation systems in location-based social networks (LBSNs), such as Yelp or Foursquare, with a focus on data sparsity and cold-start problems. To overcome these challenges, the thesis proposes several approaches based on the exploitation of implicit trust between users. Unlike declared friendship links (explicit trust), implicit trust is inferred from users' behavior, particularly through their check-ins and ratings. Three main models were developed to integrate this trust into recommendation systems: (1) the HRCT model (Hybrid Rating Check-in Trust), which combines ratings and check-ins to build a denser trust matrix, there by reducing data sparsity and improving recommendation accuracy; (2) the PRCT model (Propagation of Rating/Check-in for implicit Trust), an extension of the HRCT model that applies a trust propagation mechanism within the social network, helping to mitigate cold-start issues; and (3) the ITCRC model (Implicit Trust based on Combining point of interest Ratings and user Check-ins), which incorporates trust directly into the POI prediction process. The experimental results, obtained from real-world datasets such as Yelp, showed that these models help to densify the similarity matrices and improve the accuracy of POI rating predictions based on user check-ins, while also reducing the impact of sparsity and the cold start problem. In particular, approaches that incorporate check-ins into the computation of the implicit trust matrix between users proved to be more effective than those based solely on ratings.

***Keywords*:** RS, LBSN, POI Recommendation, social trust, machine learning, implicit trust, trust propagation.

Résumé

Cette thèse aborde les défis des systèmes de recommandation (SR) de points d'intérêt (POI) dans les réseaux sociaux basés sur la localisation (LBSN), tels que Yelp ou Foursquare, en se concentrant sur les problèmes de sparsité des données et de démarrage à froid. Pour pallier ces défis, cette thèse propose plusieurs approches basées sur l'exploitation de la confiance implicite entre utilisateurs. Contrairement aux liens d'amitié déclarés (confiance explicite), la confiance implicite est déduite des comportements des utilisateurs, notamment à travers leurs check-ins et ratings. Trois modèles principaux ont été développés pour intégrer cette confiance dans les SRs : (1) le modèle HRCT (Hybrid Rating Check-in Trust) combinant ratings et check-ins pour construire une matrice de confiance plus dense, réduisant ainsi la sparsité des données et améliorant la précision des recommandations, (2) le modèle PRCT (Propagation of Rating/Check-in for implicit Trust), une extension du modèle HRCT qui applique un mécanisme de propagation de la confiance dans le réseau social, atténuant ainsi les problèmes liés au démarrage à froid et (3) le modèle ITCRC (Implicit Trust based on Combining point of interest Ratings and user Check-ins) qui intégre la confiance dans le processus de prédiction des POIs. Les résultats expérimentaux, obtenus à partir de jeux de données réels comme Yelp, ont montré que ces modèles permettent de densifier les matrices de similarité et d'améliorer la précision de la prédiction des notations POIs à partir des checkins des utilisateurs tout en réduisant l'impact de la sparsité et du démarrage à froid. En particulier, les approches basées sur les check-ins dans le calcul de la matrice des confiances implicites entre les utilisateurs se sont révélées plus efficaces que celles basés uniquement sur les évaluations.

Mots clés : SR, LBSN, recommandation de POI, confiance sociale, apprentissage automatique, confiance implicite, propagation de confiance.

ملخص

تعالج هذه الأطروحة التحديات المرتبطة بأنظمة التوصية بنقاط الاهتمام (POI) في شبكات التواصل الاجتماعي المعتمدة على الموقع(LBSNs) ، مثل Yelp أوFoursquare ، مع التركيز على مشكلتي تشتت البيانات (sparsity) والبداية الباردة (cold-start).ولمواجهة هذه التحديات، تقترح الأطروحة عدة منهجيات تعتمد على استغلال الثقة الضمنية بين المستخدمين. وعلى عكس روابط الصداقة المصرح بها (الثقة الصريحة)، فإن الثقة الضمنية تُستنتج من سلوك المستخدمين، وخاصة من خلال تسجيلاتهم (check-ins) وتقييماتهم (ratings).

تم تطوير ثلاثة نماذج رئيسية لدمج هذه الثقة في أنظمة التوصية:

1. نموذج **HRCT**(الثقة الهجينة بناءً على التقييمات وتسجيلات الدخول)، والذي يجمع بين التقييمات وتسجيلات الدخول لبناء مصفوفة ثقة أكثر كثافة، مما يساهم في تقليل تشتت البيانات وتحسين دقة التوصيات.
2. نموذج **PRCT**(نشر الثقة الضمنية عبر التقييمات وتسجيلات الدخول)، وهو امتداد لنموذج HRCT ويعتمد على آلية نشر الثقة داخل الشبكة الاجتماعية، مما يساعد على الحد من مشكلة البداية الباردة.
3. نموذج **ITCRC**(الثقة الضمنية بناءً على دمج تقييمات نقاط الاهتمام وتسجيلات دخول المستخدمين)، والذي يدمج الثقة مباشرة في عملية التنبؤ بنقاط الاهتمام.

وقد أظهرت النتائج التجريبية، المستخلصة من مجموعات بيانات حقيقية مثلYelp ، أن هذه النماذج تساهم في تكثيف مصفوفات التشابه وتحسين دقة التنبؤ بتقييمات نقاط الاهتمام بناءً على تسجيلات دخول المستخدمين، مع تقليل تأثير تشتت البيانات ومشكلة البداية الباردة. وعلى وجه الخصوص، أثبتت المنهجيات التي تعتمد على تسجيلات الدخول في حساب مصفوفة الثقة الضمنية بين المستخدمين فعاليتها أكثر من تلك التي تعتمد فقط على التقييمات.

الكلمات المفتاحية :نظام التوصية(RS) ، شبكات التواصل الاجتماعي المعتمدة على الموقع(LBSN) ، التوصية بنقاط الاهتمام (POI)، الثقة الاجتماعية، التعلم الآلي، الثقة الضمنية، نشر الثقة.

General Introduction

1. Context

Technological advancements and the rise of mobile devices have fostered the emergence of a new generation of information systems, particularly location-based social networks (LBSNs). These platforms allow users to share their experiences at visited locations, exchange reviews, and receive personalized recommendations. In this context, point-of-interest (POI) recommendation systems (RS) play a key role in facilitating the discovery of new places based on users' preferences and interaction history.

POI recommendations are particularly valuable in various fields such as smart tourism, urban navigation, and leisure services. However, traditional recommendation approaches face several challenges related to the evolving nature of interactions between users and points of interest.

With the rise of LBSNs such as Yelp, Foursquare, Facebook Places, etc., POI recommendation has become an essential tool for guiding users in exploring their surroundings. These platforms generate large volumes of heterogeneous data, including ratings, check-ins, textual reviews, and social interactions. However, this abundance of data comes with major challenges:

- **Information overload**: Faced with a multitude of options, users struggle to identify the most relevant places.
- **Dynamics of preferences**: Users' tastes vary according to contextual factors (weather conditions, time of day, company) and social factors (influence of friends).

Traditional recommendation systems, particularly collaborative filtering, struggle to address these challenges due to structural constraints such as data sparsity and the cold-start problem.

2. Problem Statement

Although traditional recommendation systems are effective, they have several limitations. Among the main challenges they face, we can distinguish:

- **Data sparsity**: The majority of users interact with a limited number of POIs, making it difficult to predict their preferences and limiting the reliability of recommendations.
- **Cold-start**: When a new user or a new POI enters the system, the lack of interaction history complicates the generation of relevant recommendations.
- **Lack of integration of social relationships**: Many recommendation systems overlook the social dimension of interactions, while trust between users can significantly influence their choices.

A promising approach to overcoming these challenges is to exploit social trust within LBSNs. This trust can be:

- **Explicit**, when users directly declare their level of trust towards other members.
- **Implicit**, when it is inferred from past interactions, such as POI ratings and check-ins.

This research focuses on the integration of implicit trust into POI recommendation systems in order to improve the relevance of suggestions. Despite recent advancements, two major obstacles remain in POI recommendation:

- **Data sparsity:** Limited interactions between users and POIs make the user-POI matrices extremely sparse, affecting the accuracy of similarity calculations.

- **Cold-start:**
  - New users: The lack of interaction history prevents the personalization of recommendations.
  - New POIs: The absence of ratings or check-ins complicates the evaluation of their relevance.

Moreover, traditional approaches tend to overlook the social and temporal dynamics of interactions. For example, collaborative filtering often relies on static similarity measures (such as Pearson or Cosine), while models based on social trust, although promising, lack the flexibility to capture implicit relationships derived from users' behaviors.

3. Motivation

The integration of implicit social trust into recommendation systems (RS) represents a promising approach to improving the quality of suggestions while addressing some of the limitations of traditional methods. Unlike explicit trust, which is based on declared friendship links, implicit trust is inferred from user behaviors (check-ins, ratings) and offers several advantages:

- **Reduction of sparsity:** By leveraging transitive relationships between users, it helps fill in incomplete data and enrich existing interactions.
- **Cold-start mitigation:** By relying on check-in/rating similarities between users without shared history, it allows for the inference of preferences even in the absence of explicit data.
- **Improved personalization**: It incorporates behavioral and contextual factors (spatial, temporal, social) to refine recommendations.

As users generally place more trust in recommendations coming from their social circles than in those generated anonymously, this approach helps optimize the relevance of suggestions. However, dynamically and contextually modeling this trust in a complex environment such as LBSNs represents a major challenge.

In this context, our work is based on the hypothesis that implicit trust relationships, inferred from user interactions, can enhance recommendation accuracy. To achieve this, we explore three complementary directions:

1. **Combine ratings and check-ins** to compute a measure of implicit trust between users.
2. **Propagate this trust** through transitive relationships in order to enrich the user-POI matrices and reduce sparsity.
3. **Integrate this hybrid trust** into the POI recommendation process to improve the relevance of suggestions.

4. Contributions

This research aims to propose solutions to overcome the limitations of traditional recommendation systems by leveraging implicit trust relationships within LBSNs. These solutions are designed to address challenges related to data sparsity, the cold-start problem, and the integration of social interactions into point-of-interest (POI) recommendation. For this reason, this thesis presents several major contributions, centered around the exploitation of implicit trust in recommendation systems.

1. Development of new models leveraging implicit trust:

- **HRCT Model (Hybrid Rating Check-in Trust):** A hybrid approach that combines ratings and check-ins to construct a denser and more accurate trust matrix, thereby improving the quality of recommendations.
- **PRCT Model (Propagation of Rating/Check-in for implicit Trust):** An extension of the HRCT model that integrates a trust propagation mechanism through social networks, helping to reduce sparsity and alleviate the cold-start problem.
- **ITCRC Model (Implicit Trust based on Combining point of interest Ratings and user Check-ins):** A model that combines various sources of information (ratings, check-ins) to optimize both the accuracy and relevance of recommendations.

2. In-depth analysis of the context and challenges related to POI recommendation:

- Study of location-based social networks (LBSNs) and existing recommendation techniques (collaborative filtering, contextualization).
- Critical review of models integrating social trust, identifying the distinction between techniques founded on explicitly stated trust and those inferred from user behavior or interactions.

3. Modeling and integration of implicit trust:

- Definition of a formal framework for inferring implicit trust from user behaviors (check-ins and ratings).
- Proposal of trust propagation algorithms to enrich interactions and reduce data sparsity.

4. Experimental evaluation and validation of the proposed models:

- **HRCT:** Computation user-user similarities, and validation on real-world dataset using standard metrics (RMSE, PRECISION, RECALL).
- **PRCT:** Analysis of the impact of trust propagation on model robustness, and comparison with classical methods (Resnick, O'Donovan).
- **ITCRC:** Dynamic hybridization of information sources and comparative evaluation, demonstrating significant improvements in recommendation accuracy and better resilience to the cold-start problem.

Moreover, this thesis stands out through its multidimensional and adaptive approach, integrating:

- **Heterogeneous data** (ratings, check-ins, temporal and spatial context) to enrich recommendations.
- **Innovative algorithms** for trust propagation and similarity computation, enabling the exploitation of implicit relationships between users.
- **Rigorous evaluation** on real-world data (Yelp), demonstrating a significant reduction in RMSE and improved performance in terms of Precision and Recall.

5. Thesis Structure

This thesis is structured into six chapters, covering both the problem definition and the proposed solutions. The first three chapters (I, II, and III) form the theoretical part of this thesis. They introduce the fundamental concepts essential to understanding the research context, define the problem related to integrating implicit trust in LBSNs, and present a state-of-the-art review of POI recommendation approaches, including users' social context. Chapters IV, V, and VI are dedicated to the three main contributions of this thesis. Aside from the general introduction and conclusion, the structure of this document is organized as follows:

- **Chapter I** introduces the key concepts of LBSNs and their relationships with POIs. It then explores the main POI recommendation approaches, highlighting content-based filtering, collaborative filtering (similarity techniques and matrix factorization), as well as recent advances in deep learning. Finally, it discusses evaluation methods for recommender systems, emphasizing the importance of contextual and social factors for enhanced personalization.

- **Chapter II** defines social trust in LBSNs and highlights its importance in these networks, before examining the challenges of modeling trust and the role of social influence propagation in overcoming data limitations. This chapter then explores various modeling approaches for social trust, drawing on methods such as trust graphs, trust-based collaborative filtering, hybrid models, and advances in deep learning. It also highlights O'Donovan's approach to collaborative filtering, as well as trust propagation mechanisms like trust score calculation and path composition, which appear promising for optimizing recommendation personalization.

- **Chapter III** begins with a reminder of the link between LBSNs and POI recommendations, then examines different techniques that leverage social trust. It explores collaborative models integrating social filtering, the use of trust network graphs, and neighborhood-based approaches. Additionally, it analyzes the benefits of incorporating contextual data and hybrid models to refine recommendations. The chapter then provides a state-of-the-art review of existing works on leveraging social trust in recommender systems, distinguishing between approaches based on explicit and implicit trust. Finally, a critical discussion of these methods is presented, outlining their strengths and limitations to identify potential areas for improvement.

- **Chapter IV** presents the HRCT model (Hybrid Rating Check-in Trust), designed to enhance POI recommendation prediction by leveraging implicit trust and similarity measures. After defining and formulating the problem, the chapter details the mechanisms for computing implicit trust from ratings and check-ins, illustrated with concrete examples of trust matrix construction. It then introduces the proposed algorithms and describes the HRCT model in detail. Finally, an experimental evaluation is conducted to analyze the model's performance using various metrics and compare it with its variants and other existing recommendation approaches.

- **Chapter V** introduces the PRCT model (Propagation of Rating/Check-in for implicit Trust) which leverages implicit trust propagation to improve POI recommendation accuracy. Following a detailed problem definition and formulation, this chapter describes the methods for computing implicit trust and the mechanisms for propagating this trust within LBSNs. It then presents the proposed algorithms and details the PRCT model's operation, followed by an in-depth evaluation of its performance. Lastly, the experimental results are analyzed to study the impact of trust propagation on recommendation quality and data sparsity management.

- **Chapter VI**, the final chapter, presents the ITCRC model (Implicit Trust based on Combining point of interest Ratings and user Check-ins) which integrates trust into the prediction computation process to enhance POI recommendation quality. After defining and formulating the problem, the chapter builds upon O'Donovan's prediction formula to introduce methods for estimating ratings based on implicit trust relationships among users. It then thoroughly describes the proposed algorithms and the functioning of the ITCRC model, followed by an extensive performance evaluation. A critical discussion of the results concludes the chapter, synthesizing the model's contributions and identifying future directions for more personalized recommendations.

Part I
Theoretical Foundations
And
State of the Art

Chapter I:

POI Recommendation and LBSNs

I.1. Introduction

Location-Based Social Networks (LBSNs) have emerged as a valuable source of data to enhance the recommendation of Points of Interest (POIs). By leveraging social trust, these systems can refine recommendations by considering interactions and relationships between users. This chapter presents the fundamental concepts related to LBSNs, POI recommendation, the integration of social trust, and the evaluation of these systems.

I.2. LBSNs and POIs

LBSNs are digital platforms where users share their location and interact with other members based on their geographical position. They are characterized by several key aspects:

- **Geolocation**: Users record their visits to specific locations via check-ins (Zheng et al., 2009).
- **Social Relationships** : Interactions between users influence location recommendations (Ye et al., 2010).
- **Contextual Data**: Factors such as time, weather, or personal preferences affect recommendations (H. Gao et al., 2015).
- **Visit History**: Analyzing previously visited locations helps identify user preferences and predict future destinations.

LBSNs collect various information, including user check-in data, geographical coordinates of visited locations, and reviews or advice shared by users. They also enable the development of friendships and the sharing of real-time information. Interactions in these networks can be classified into two main categories:

- **Interactions between users and POIs**: This includes check-ins, ratings and reviews left by users.
- **Interactions between users**: Social relationships, such as friendships, influence recommendations.

To enrich the user experience, recommendation systems leverage these interactions to suggest new places based on users' preferences and behaviors. Several LBSN platforms are commonly used in POI recommendation research, including:

- **Foursquare:** One of the most popular LBSNs, where users check in, leave reviews, and receive personalized recommendations (D. Yang et al., 2013), (B. Liu & Xiong, 2013).
- **Yelp**: Known for its star ratings and detailed reviews of businesses.

- **Brightkite and Gowalla:** Two now-defunct LBSNs that provided widely-used datasets in research (Cho et al., 2011), (Scellato et al., 2011).
- **Other platforms:** Jiepang (the Chinese equivalent of Foursquare) (Lian et al., 2014), GeoLife, as well as data from Twitter (J.-D. Zhang & Chow, 2013) and TripAdvisor.

In what follows, Table 1 compares the main characteristics of these LBSN datasets to better illustrate their differences and similarities.

Table 1: Comparison of Main Characteristics of LBSN Datasets (Sánchez & Bellogín, 2022)

Platform	Status	Data Types	Accessibility	Use in Research
Foursquare	Active	Check-ins, places, users, timestamps	Public (via API, dumps)	POI recommendation, mobility analysis
Yelp	Active	Ratings, textual reviews, check-ins	Public (via Yelp dataset)	Recommender systems, trust modeling
Brightkite	Defunct (2011)	Check-ins, user IDs, locations	Public (archived datasets)	Mobility studies, community detection
Gowalla	Defunct (2012)	Trajectories, check-ins, POIs, users	Public (archived datasets)	Trajectory mining, user behavior analysis
Jiepang	Largely inactive	Check-ins, social links, photos	Limited	Cultural and regional analysis
GeoLife	Inactive (data collection ended)	Precise GPS trajectories (real-time)	Public (Microsoft Research)	Trajectory prediction, mobility analysis
Twitter	Active	Geotagged tweets, text, user information	Public (via API)	Event detection, sentiment and mobility analysis
TripAdvisor	Active	Reviews, ratings, location metadata (restaurants, hotels)	Semi-public (scraping/API)	Tourism recommendation, opinion mining

In Location-Based Social Networks (LBSNs), Points of Interest (POIs) represent physical places such as restaurants, monuments, parks, or shops, which structure users' interactions. These POIs are at the heart of LBSN dynamics, as they serve as the basis for check-ins, reviews, recommendations, and geolocated shares. For example, studies like that of Zheng et al. (Zheng et al., 2011) show that POIs allow for the analysis of urban behaviors and the personalization of recommendations by cross-referencing spatial and social data. Users frequently evaluate these places, thereby creating critical data sets for recommendation algorithms (Bao et al., 2012). However, managing POIs presents challenges, particularly in validating their authenticity or handling biases related to non-representative contributions. Thus, POIs are not only geographical markers but key elements for understanding the relationship between individuals, their environment, and trust dynamics within LBSNs.

In our thesis work, we have chosen Yelp as the reference platform due to the simultaneous availability of ratings and check-ins, making it a rich and relevant dataset for analyzing POI recommendations (Medjroud et al., 2025a).

I.3. POI Recommendation

POI recommendation systems aim to suggest relevant locations (POIs) to users based on various factors. Among the most common approaches, we distinguish three categories (see Figure 1):

- **Content-based approaches**: These approaches analyze the characteristics of visited POIs to identify similar locations (H. Zhang et al., 2021), (Baral et al., 2018), (Yin et al., 2013).
- **Collaborative Filtering approaches**: These approaches leverage the preferences of users with similar profiles to make recommendations (Cheng et al., 2021).
- **Contextual recommendation approaches**: These approaches integrate contextual data (time, weather, current location) to refine POI suggestions (Wan et al., 2022), (Y. Li et al., 2019).

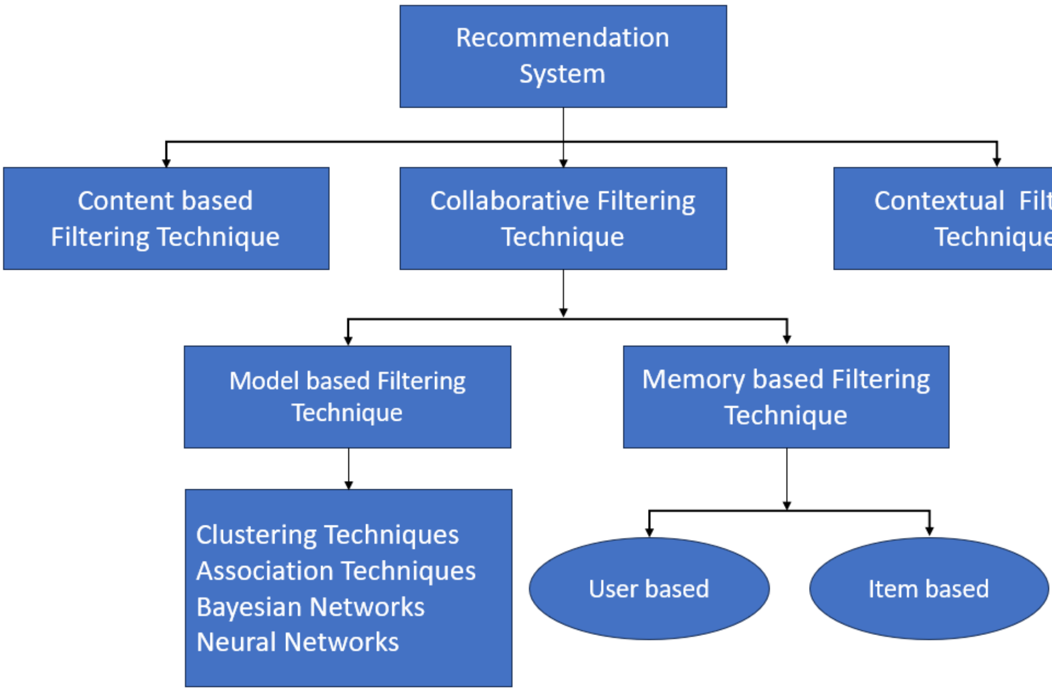

Figure 1: The Recommendation System Types

I.3.1. Content-based Filtering Approaches for POIs

Content-based filtering is a recommendation approach that relies on analyzing the characteristics of Points of Interest (POIs) to identify locations similar to those already visited by a user. This method assumes that if a user likes a certain type of POI, they are likely to enjoy other places sharing similar attributes. This approach involves four steps:

(1) **Representation of POIs through Features**: Each POI is described using a set of attributes, such as:

- **Category**: Restaurant, museum, hotel, park, shopping center, etc.
- **Keywords and descriptions**: Textual information about the place, often extracted from user reviews.
- **User ratings**: The average score assigned to the POI.
- **Geographical location**: The position of the POI, which can be used to recommend nearby locations.
- **Offered services**: Free Wi-Fi, parking, accessibility, etc.

(2) **Building the User Profile**: A user profile is generated by analyzing the POIs the user has previously visited or rated. This profile can be represented as a preference vector, where each dimension corresponds to a characteristic of the POI (for example, a user who frequently visits vegetarian restaurants will receive more recommendations for similar places).

(3) **Calculating Similarity between POIs**: Once the user profile is defined, the algorithm searches for POIs with similar characteristics. Different methods can be used to measure this similarity:

- **Cosine similarity**: Measures the angle between two POI vectors based on their textual features.
- **Latent Semantic Analysis (LSA)**: Used to extract relationships between keywords and improve the understanding of textual content.
- **Neural networks and deep learning**: Used to capture complex relationships between POIs by leveraging word embeddings and advanced vector representations.

(4) **Generating Recommendations**: The algorithm selects the POIs most similar to those favored by the user and ranks them according to their relevance. Weighting can be applied to prioritize certain features (for example, giving more importance to the POI category than its textual description).

Content-based filtering offers several advantages that make it a valuable approach for POI recommendation. Firstly, it provides explainability of recommendations, allowing users to understand why a particular place is being suggested. Additionally, it ensures effective personalization by tailoring recommendations to each user's specific preferences. This method is also particularly useful for new users, as it can generate relevant suggestions even without social interactions, based solely on the few places the user has visited or rated. However, this approach also comes with certain limitations. It notably suffers from the over-specialization problem, where users are likely to receive suggestions that are too similar to their past choices, thus reducing the diversity of recommendations. Moreover, it requires rich and detailed descriptions of POIs to operate effectively; in the absence of sufficient data, the quality and relevance of recommendations may be compromised. Finally, unlike collaborative filtering, this method ignores the behavior of other users, which may limit the discovery of new popular places and reduce the social dimension of the recommendation process. Thus, although content-based filtering is effective in recommending POIs similar to those already visited, its limitations make it necessary to combine it with other approaches; such as collaborative filtering or context-aware recommendation; to enrich the user experience and ensure greater diversity in the suggestions.

I.3.2. Collaborative Filtering

Collaborative filtering is a widely used approach in recommender systems, leveraging past interactions between users and Points of Interest (POIs) to generate relevant suggestions.

It is based on the assumption that users who have expressed similar preferences in the past will likely have similar tastes in the future. This technique is divided into two main categories: memory-based methods and model-based methods. Memory-based methods, also known as neighborhood approaches, work by measuring the similarity between users (user-based) or between POIs (item-based). They identify groups of users with similar preferences to recommend places based on the choices of their peers. Although these methods are simple and interpretable, they often suffer from data sparsity and scalability issues, especially when applied to large datasets. On the other hand, model-based methods use advanced techniques such as matrix factorization and deep learning to identify latent patterns in user-POI interactions. These methods improve recommendation accuracy by capturing implicit relationships between users and items, helping mitigate sparsity. However, they require prior training and may be more computationally expensive. Despite their differences, both approaches are often combined in hybrid models to leverage their respective strengths and compensate for their limitations.

I.3.2.1. Item-Based Collaborative Filtering (POI-POI Similarity)

POI-POI similarity-based collaborative filtering (also called Item-Based Collaborative Filtering) relies on analyzing relationships between POIs to recommend places similar to those a user has already visited (Sarwar et al., 2001), as shown in the Figure 2 below.

Unlike user-based collaborative filtering; which recommends POIs based on the preferences of users with similar profiles; this approach identifies POIs that are frequently visited by users who exhibit similar behaviors. The item-based collaborative filtering process involves several steps:

(1) Construction of the POI similarity matrix: A matrix is built by analyzing user interactions with various POIs. Two POIs are considered similar if they are often visited by the same users. Several metrics can be used to measure this similarity, such as:

- **Cosine similarity:** Measures the degree of resemblance between two POIs based on the users who have visited them (Gupta et al., 2020).
- **Pearson Correlation Coefficient:** Analyzes the correlation between ratings assigned to the POIs by users (Bobadilla et al., 2010).
- **Jaccard index:** Calculates the overlap rate between the sets of users who have visited each POI (H. Liu et al., 2014).
- **Slope One algorithm:** Exploits rating differences between POIs visited by users to estimate the likelihood of a user liking a given POI (Lemire & Maclachlan, 2005).

(2) Identification of similar POIs: Once the similarity matrix is built, for each POI visited by a user, a set of similar POIs is identified.

(3) Generation of recommendations: The algorithm recommends to the user the POIs with high similarity to those they have already visited, weighting the suggestions based on the similarity scores obtained in the matrix.

POI-POI similarity-based collaborative filtering presents several advantages that make it particularly effective for the recommendation of points of interest. It is stable and robust, because unlike user-user similarity-based methods, it does not require frequent updates when new users join the system. Indeed, the relationships between POIs evolve more slowly, which allows for a more coherent modeling of user preferences. Moreover, this approach is highly scalable, as it is less affected by the exponential growth of the number of users, making it particularly suitable for large databases. Another notable advantage is that it works well for users with few interactions, as the recommendations rely more on the popularity of POIs and global trends rather than on a limited personal history. However, this method also presents certain limitations. One of the main ones is the lack of personalization, as the suggestions are based on similarity between POIs rather than on the specific preferences of a user, which may reduce their relevance for profiles with very particular tastes. Furthermore, it heavily depends on interaction data, which means that the more users visit different POIs, the larger and more complex the similarity matrix becomes, thus increasing computational and storage needs. Finally, this approach encounters difficulties in recommending new POIs that have not yet been sufficiently visited by users; a problem known as the cold start. To overcome these limitations, POI-POI collaborative filtering is often combined with other techniques, such as content-based recommendation (to enhance personalization) or contextual approaches (to introduce situational relevance), in order to improve the overall relevance and user satisfaction.

I.3.2.2. User-User Collaborative Filtering

User-user collaborative filtering is based on the idea that users who have exhibited similar behaviors in the past will likely have similar preferences in the future (Zhi-Dan Zhao & Ming-Sheng Shang, 2010), as shown in the following Figure 2:

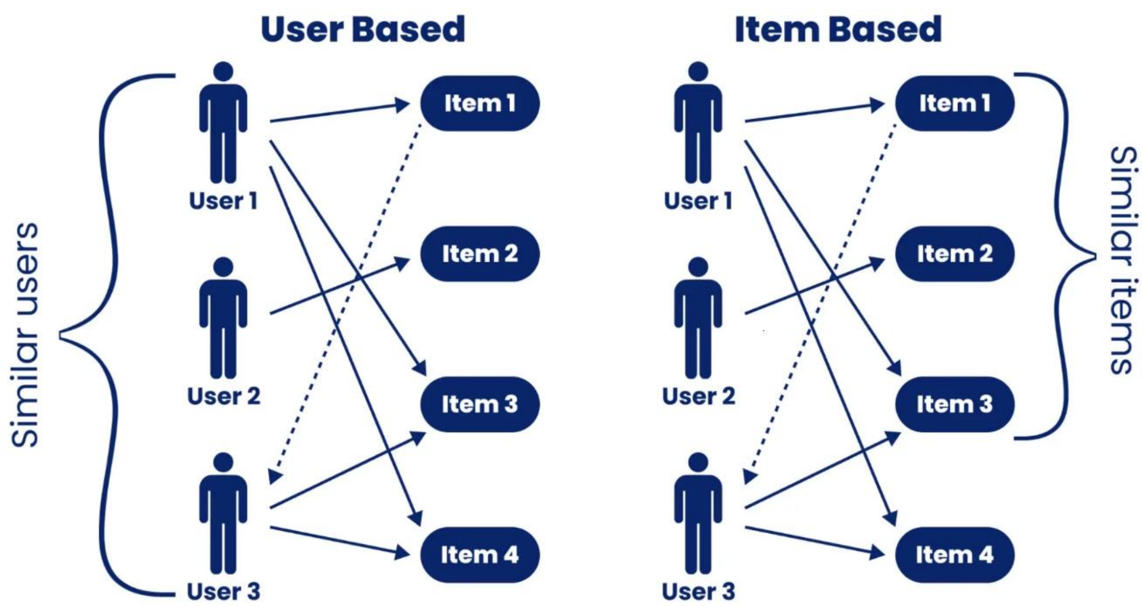

Figure 2: Item-based Vs User-based Collaborative Filtering

This approach involves identifying groups of users who share common habits and recommending to a given user the POIs preferred by their nearest neighbors. Its operation is based on four key steps:

(1) **Construction of the user-POI matrix**

- Each user is represented by a vector containing the POIs they have visited and the ratings they have given (if available).
- A user-POI matrix is built, where each row corresponds to a user and each column to a POI.

(2) **Calculating similarity between users:** To compare users with one another, various similarity measures are used, such as:

- **Cosine similarity:** Measures the angle between users' rating vectors.
- **Pearson correlation:** Identifies common trends by neutralizing differences in individual rating scales.
- **Jaccard index:** Evaluates the overlap between the sets of POIs visited by two users.
- **Slope One algorithm:** Exploits rating differences between POIs visited by users to estimate the likelihood that a user will enjoy a given POI.

(3) **Identifying similar users (neighborhood):** For a target user, the algorithm selects a group of neighboring users with the highest similarity scores. These neighbors serve as a reference for generating recommendations.

(4) **Rating prediction and recommendation:** By analyzing the POIs visited by the neighbors but not yet explored by the target user, the algorithm predicts the potential interest the user might have in those places. The POIs most highly rated by similar users are prioritized in the recommendation.

For example, if three users U1, U2, and U3 rated three POIs (A, B, C), as shown in Table 2 below:

Table 2: Example of the evaluation of three POIs by three users

User-POI	POI A	POI B	POI C
U1	5	4	?
U2	4	3	2
U3	5	4	3

- We calculate the cosine similarity between U1 and U3, and between U1 and U2.
- If U3 is the most similar to U1, then the rating given by U3 to POI C (3) will be used to estimate U1's preference for this POI.
- The algorithm then recommends POI C to U1.

This method offers several advantages, including improved recommendation accuracy by leveraging comparable user profiles, adaptability to evolving preferences, and effectiveness for popular POIs that have been widely visited. However, it also presents certain limitations, such as the sparsity problem, which makes it difficult to identify relevant neighbors when few interactions are available, limited scalability when dealing with a large number of users, and the challenge of recommending POIs to new users who have not yet performed any interactions (cold start problem). To overcome these challenges, this approach is often combined with other techniques, such as content-based recommendation or hybrid models (J. Wang et al., 2006), in order to enhance the diversity and relevance of the recommendations.

I.3.2.3. Model-Based Collaborative Filtering using Matrix Factorization

Collaborative filtering based on matrix factorization (Matrix Factorization - MF) (Xin Luo et al., 2014) is a powerful approach for recommending Points of Interest (POIs), particularly effective for handling large-scale datasets and mitigating the sparsity problem in user-POI interactions. Unlike memory-based methods (user-based and item-based), which rely on direct similarity computations, matrix factorization leverages mathematical models to learn latent representations of users and POIs, thus enabling more accurate and generalizable recommendations.

The main objective of matrix factorization is to decompose the user-POI interaction matrix into two smaller latent factor matrices:

- A matrix **U**, representing user preferences as low-dimensional vectors.
- A matrix **V**, representing POI characteristics in the same latent space.

The original matrix is thus approximated by the product of these two matrices (see Equation (1)):

Equation 1: Approximation of the Rating Matrix Using Latent Features

$$R \approx U \times V^{\mathrm{T}}$$

Where:

- **R** is the user-POI interaction matrix (e.g., ratings, check-ins).
- **U** is the matrix of users' latent representations.
- **V** is the matrix of POIs' latent representations.

This decomposition makes it possible to predict unknown values in the matrix R, i.e., to recommend POIs not yet visited but likely to be of interest to a user.

Various matrix factorization techniques are used in POI recommendation, each with specific characteristics tailored to different contexts. Singular Value Decomposition (SVD) (Anwar et al., 2021) is a classic method that decomposes the user-POI matrix into three matrices (U, Σ, V), thereby reducing dimensionality by capturing the principal components of interactions. However, since this technique requires a dense matrix, it is often improved with SVD++, which incorporates implicit interactions such as check-ins and likes. Another approach, Probabilistic Matrix Factorization (PMF) (Salakhutdinov & Mnih, 2008), introduces a probabilistic model to estimate latent factors, offering better handling of missing data; a recurring issue in POI recommendation systems. The Alternating Least Squares (ALS) (Takács & Tikk, 2012) method minimizes reconstruction error by alternating the optimization of matrices U and V, making it particularly suitable for large-scale datasets and parallel computations, such as those performed with Apache Spark. Finally, Non-Negative Matrix Factorization (NMF) (H. Li et al., 2019) stands out for its constraint that forces the values in matrices U and V to remain positive, thereby allowing for more intuitive interpretation of relationships between users and POIs. These various techniques (see Table 3) are often combined or adapted to improve the accuracy and relevance of recommendations based on the specific characteristics of user-POI data.

Table 3: Matrix Factorization Techniques Used in POI Recommendation

Technique	Description	Advantages	Disadvantages
Singular Value Decomposition (SVD)	Decomposes the user-POI matrix into three matrices (U, Σ, V), capturing the main components of the interactions.	Reduces dimensionality and improves recommendation accuracy.	Requires a dense matrix, often combined with SVD++ to handle implicit interactions.
Probabilistic Matrix Factorization (PMF)	Introduces a probabilistic model to estimate latent factors for users and POIs.	Handles missing data better, which is common in POI recommendation.	More complex to train and requires careful tuning of hyperparameters.
Alternating Least Squares (ALS)	Minimizes reconstruction error by alternating the optimization of matrices U and V.	Highly scalable, used for large datasets, and compatible with distributed systems (e.g., Apache Spark).	May perform less effectively on highly sparse datasets.
Non-Negative Matrix Factorization (NMF)	Constrains the values of U and V matrices to be positive, making latent factors easier to interpret.	Offers better interpretability of user-POI relationships.	Less flexible than other approaches and requires good data preprocessing.

Matrix Factorization-Based Collaborative Filtering for POI Recommendation Matrix factorization-based collaborative filtering plays a key role in POI (Point of Interest) recommendation, particularly in addressing several common challenges. First, it enables advanced personalization by extracting latent representations of users and POIs. Through this approach, the model can identify places that match a user's implicit preferences; even when no direct interaction between the user and those POIs is recorded in the interaction matrix. Next, matrix factorization provides an effective solution to the sparsity problem, which is especially prevalent in Location-Based Social Networks (LBSNs), where most users interact with only a limited number of POIs. By leveraging latent factors, matrix factorization allows for the prediction of missing values, thus increasing both the coverage and accuracy of recommendations. Finally, this approach can incorporate contextual information to refine suggestions. Some variations of matrix factorization take into account temporal, geographic, or social factors, enabling the system to adapt recommendations based on the time of day, geographic proximity, or the user's social interactions. These capabilities make matrix factorization a powerful and versatile tool for improving the quality and relevance of POI recommendations. The following Table 4 highlights the strengths and limitations of matrix factorization in POI recommendation, emphasizing its effectiveness for handling large-scale datasets and addressing cold start problems for new users or POIs.

Table 4: Strengths and Limitations of Matrix Factorization

Aspect	Advantages	Limitations
Capturing Complex Relationships	Identifies implicit links between users and POIs, even without direct similarities.	The learned latent factors are not always easily interpretable.
Handling Data Sparsity	Generates predictions even for rarely visited POIs, improving recommendation coverage.	Requires prior training, which can be time- and resource-intensive.
Scalability and Performance	Can be optimized to handle large-scale datasets, including millions of interactions.	Less effective in cold-start scenarios, as it relies on a minimum number of interactions to model users and POIs.

I.3.2.4. Deep Learning-Based Collaborative Filtering

Deep learning has revolutionized the field of recommendation by enabling the modeling of complex relationships between users and points of interest (POIs). Unlike traditional collaborative filtering approaches, which rely on explicit similarity calculations or matrix factorization, deep learning models capture non-linear representations and exploit large amounts of data (ratings, check-ins, temporal and social contexts) to improve the relevance of recommendations.

Deep learning-based collaborative filtering uses neural networks to extract complex features from user-POI interactions. This approach follows several key steps:

1. **Encoding User-POI Interactions:** Each user and each POI are represented as dense vectors in a latent space learned by a deep learning model. These representations are enriched with additional data, such as textual reviews, temporal and social factors, to improve recommendation accuracy.
2. **Learning Representations:** The model is trained on large datasets by learning to predict user-POI interactions (e.g., rating a POI or checking into one). Techniques like supervised learning (rating prediction) or unsupervised learning (embedding learning) are used.
3. **Prediction and Recommendation Generation:** Once the model is trained, it can recommend the most relevant POIs by analyzing similarities in the latent space and considering contextual factors. Recommendations are refined based on the user's preferences and past behavior.

Deep learning models applied to POI recommendation (Islam et al., 2022) use various neural architectures to capture complex relationships between users and points of interest. Variational Autoencoders (VAE) (Girin et al., 2020) compress user-POI interactions and generate recommendations by modeling the distribution of user preferences. Deep Neural Networks (DNN) (Qian et al., 2014), on the other hand, use multiple layers of neurons to learn these complex relationships and integrate diverse data, such as check-ins, textual reviews, and contextual information. Other models, such as Convolutional Neural Networks (CNN) (Nilla & Setiawan, 2024), are particularly effective at analyzing images associated with POIs, such as tourist attractions or restaurants, and enrich recommendations by combining visual and geospatial features. To capture the temporal dimension of interactions, Recurrent Neural Networks (RNN) (Cui et al., 2020) and Long Short-Term Memory (LSTM) (Devooght & Bersini, 2017) networks are used to optimize recommendations based on past visit sequences, allowing suggestions to adapt according to users' routes and habits. Finally, Graph Neural Networks (GNN) (C. Gao et al., 2022), (Wu et al., 2023), (C. Gao et al., 2023) exploit the structure of social networks and user-POI interactions in the form of graphs, capturing richer and more dynamic relationships between users, their friends, and visited POIs. These different approaches, often combined, enhance the relevance and personalization of recommendations by considering a wide range of factors influencing users' choices.

Table 5 provides a summary of the main deep learning approaches used to improve the accuracy and personalization of POI recommendations.

I.3.3. Contextual POI Recommendation

Contextual POI recommendation aims to enhance the relevance of suggestions by incorporating contextual factors that influence user choices. Unlike traditional approaches that rely solely on past interactions, this method considers dynamic elements such as the time of day, current location, weather, ongoing events, and the user’s social relationships. In the context of Location-Based Social Networks (LBSN), the integration of social context plays a key role in optimizing recommendations, as user decisions are often influenced by their social circle and interactions with other network members.

Table 5: Deep Learning Approaches Used in POI Recommendation

Model	Description	Use in POI Recommendation
Variational Autoencoders (VAE)	Compression of user-POI interactions and modeling preferences.	Generation of recommendations by learning the distribution of user preferences.
Deep Neural Networks (DNN)	Use of multiple neural layers to learn complex relationships.	Integration of heterogeneous data such as check-ins, textual reviews, and context.
Convolutional Neural Networks (CNN)	Analysis of visual and geospatial features.	Use of POI images (e.g., tourist attractions, restaurants) to refine recommendations.
Recurrent Neural Networks (RNN) and LSTM	Modeling of temporal sequences of user-POI interactions.	Adaptation of recommendations based on users' routes and habits.
Graph Neural Networks (GNN)	Exploitation of relationships in graph form.	Capture of social interactions and links between users and POIs for richer and more dynamic recommendations.

I.3.3.1. Contextual Factors of POIs

Contextual recommendation is based on the assumption that user preferences are not static but evolve based on various contextual factors. These factors can be classified into several categories:

1. **Spatial and Geographical Context**

The geographical factor is one of the most decisive elements in POI recommendation. It encompasses several sub-factors:

- **Proximity of the POI:** Users are more likely to visit POIs located near their current position. For example, a user looking for a restaurant will receive recommendations prioritizing the closest establishments.
- **User Mobility:** Some recommendation models analyze the past movements of users to predict their future destinations. For instance, if a user frequently visits specific neighborhoods, recommendations can prioritize those areas.
- **Accessibility:** Recommendations take into account the ease of access to POIs, including available transportation methods, travel distance, and the existence of physical barriers (e.g., stairs, lack of parking).

Example application: A tourist exploring a city will receive recommendations for POIs within a limited radius of their location, prioritizing easily accessible places.

2. **Temporal Context**

Time is an essential factor that influences the relevance of recommendations in several dimensions:

- **Time of day:** Some POIs are more popular at specific times (e.g., cafes in the morning, cinemas in the evening). Recommendations should adapt to users' temporal preferences (Q. Yuan et al., 2013), (H. Gao et al., 2013), (Rahmani et al., 2022).
- **Day of the week:** User habits can vary depending on the day (e.g., business restaurants on weekdays, leisure spots on weekends).
- **Season and weather conditions:** A user will not receive the same recommendations in winter and summer (e.g., beach recommendations in summer and ski resort suggestions in winter) (Braunhofer et al., 2013), (Trattner et al., 2018).
- **Local events:** The presence of a festival, concert, or market influences POI choices. Recommendation systems can adjust their suggestions based on ongoing events (Macedo et al., 2015).

Example application: A user looking for a restaurant at 11 p.m. will be prioritized with recommendations for establishments still open at that hour.

3. **Environmental Context**

The immediate environment can influence the choice of POIs, particularly through:

- **Weather conditions**: Recommendations vary according to the weather. For example, a user won't be suggested an outdoor park when it's raining, but rather a museum or a shopping mall (Trattner et al., 2016).

- **Crowdedness of a location**: Some users prefer lively places, while others seek quieter environments. By leveraging real-time data (e.g., Google Maps), systems can recommend POIs based on their current occupancy rate (Migliorini et al., 2021).
- **Local trends**: Certain POIs may gain temporary popularity (e.g., a trendy new café or a restaurant going viral on social media). Integrating trends into the recommendation engine can attract users to popular and trending places.

Application example: A user searching for an outdoor activity on a sunny day will receive suggestions for parks, beaches, or hiking trails, whereas in case of rain, the recommendations will shift toward indoor activities.

4. **Social Context**

The social context plays a major role in Location-Based Social Networks (LBSNs), as it strongly influences user preferences:

- **User relationships**: Recommendations can be refined by taking into account the POIs visited by close friends or by users with similar profiles (Huang et al., 2022).
- **Friend influence**: Users are more likely to follow recommendations from people they know or consider trustworthy. For instance, if several friends have visited and liked a particular restaurant, it will be highlighted in the recommendations (Seyedhoseinzadeh et al., 2022).
- **Interest groups and communities**: Some users share common interests within specific groups (e.g., food enthusiasts, hiking lovers). Recommendations can be adapted to these collective preferences to suggest POIs aligned with the group's interests (Kim et al., 2010).

Application example: A user who enjoys Japanese cuisine and is part of a sushi fan group on an LBSN will receive recommendations based on the reviews and check-ins of group members.

Integrating contextual factors into POI recommendation significantly enhances the relevance of suggestions by tailoring them to users' real needs. However, managing these different contexts presents technical challenges, particularly in terms of real-time processing, data privacy, and handling diverse user preferences.

Among these factors, our thesis focuses specifically on social context, which is especially relevant in LBSNs, where users actively interact with POIs, and share their experiences discovering new places.

I.3.3.2. The Social Context of Users

The social context is a key factor in POI recommendation systems. In LBSNs, users establish friendships, leave reviews, perform check-ins, and share recommendations with their social circles. This social dimension can be integrated into recommendations in several ways:

1. **Influence of Friends and Social Relationships**

Users are more likely to visit a POI if their close friends or people they follow have recommended or visited it. Social propagation models leverage this influence by weighting recommendations according to the degree of social proximity between users. For instance, if a user shares multiple check-ins with a friend, the POIs visited by that friend will be considered more relevant (Z. Wang et al., 2015), (Z. Sun et al., 2015), (Alahmadi & Zeng, 2015).

2. **Propagation of Social Trust**

Implicit trust between users is a determining factor in social recommendation. By analyzing the frequency of interactions and behavioral similarity, models can identify trust relationships and adjust recommendations accordingly. A high level of social trust between two users means that POIs visited and liked by one are more likely to interest the other (Z. Zhang & Liu, 2015).

3. **Analysis of Reviews and Shared Content**

Textual reviews, comments, and ratings left by users greatly influence the perception of a POI. Recommendation models based on semantic analysis (e.g., Natural Language Processing - NLP) extract sentiments and trends from reviews to improve the relevance of recommendations. A POI recommended by several trusted friends with positive reviews will be highlighted more prominently (Chen et al., 2015).

4. **Influence of Interest Groups and Communities**

LBSN users often join thematic groups related to their interests (e.g., food lovers, urban explorers, history enthusiasts). By integrating these community affiliations, recommendations can be refined to suggest POIs that align with the group's collective interests (K.-Y. Wang et al., 2013), (Leng et al., 2022).

5. **Use of Social Graphs and Network-Based Recommendations**

The relationships between users and POIs can be modeled as social graphs where nodes represent users and POIs, and edges capture interactions (friendship, check-ins, reviews, recommendations). Graph Neural Networks (GNNs) exploit these graphs to identify social recommendation patterns, considering strong ties between users and their visit habits (Fan et al., 2019), (Yu et al., 2024).

From the following, a summary in table (Table 6) form of the different aspects of integrating social context into POI recommendation:

Table 6: Integration of Social Context in POI Recommendation

Aspect	Description	Impact on POI Recommendation
Influence of Friends and Social Ties	Users are more likely to visit a POI if it has been recommended or visited by close friends or people they follow.	Recommendation weighting based on social proximity. POIs visited by close friends are considered more relevant.
Propagation of Social Trust	Analysis of interaction frequency and behavioral similarity between users.	The stronger the social trust between users, the more likely POIs visited by one will be recommended to the other.
Review and Shared Content Analysis	Textual reviews, comments, and ratings strongly influence how a POI is perceived.	Semantic analysis (e.g., NLP) identifies sentiment and trends to recommend POIs with positive and reliable feedback.
Influence of Interest Groups and Communities	Users join groups related to their interests (e.g., gastronomy, tourism, history).	Refined recommendations tailored to the collective preferences of a group.
Use of Social Graphs and Networks	Modeling interactions as graphs linking users and POIs.	GNNs leverage these links to identify social recommendation patterns and improve relevance.

I.3.4. Discussions

The three recommendation approaches; content-based, collaborative filtering, and context-aware recommendation; each have advantages and limitations depending on the application context. The content-based approach analyzes the features of POIs already visited by a user in order to suggest similar places. It is effective when users have sufficient interaction history, but it suffers from the problem of over-specialization, limiting the discovery of new locations. In contrast, collaborative filtering leverages the preferences of users with similar profiles to generate recommendations, enabling more diverse suggestions and avoiding overfitting. However, this method is sensitive to the data sparsity problem, as it relies heavily on the volume of available interactions. Finally, context-aware recommendation enhances the process by incorporating external factors such as time, weather, or the user's current location. Although it improves the relevance of recommendations by considering immediate context, it requires real-time access to various data sources, which can make its implementation more complex. By combining these approaches into hybrid systems, it is possible to leverage their respective strengths while mitigating their limitations. Hybrid recommendation systems make it possible to benefit from the advantages of different approaches while reducing their respective drawbacks.

By combining, for example, collaborative filtering and content-based recommendation, it is possible to improve the relevance of suggestions by leveraging both user preferences and the characteristics of POIs. However, POI recommendation remains a major challenge due to several factors. First, data sparsity is an obstacle, as many users interact with only a limited number of POIs, making it difficult to accurately model their preferences. Next, the variability of preferences adds further complexity, since users' choices are not fixed and evolve depending on context and time. Finally, integrating social trust represents a considerable challenge, as it is difficult to effectively model the influence of social relationships on recommendations.

Therefore, improving POI recommendation systems relies on the development of hybrid and adaptive approaches capable of better leveraging user interactions, contextual data, and social relationships.

Moreover, contextual recommendation based on social context offers several advantages that enhance the relevance and personalization of suggestions. By taking into account users' dynamic preferences, it allows recommendations to be adapted according to recent interactions and emerging trends. The integration of social relationships and implicit trust between users strengthens the accuracy of recommendations, making the suggested POIs more appealing and better aligned with each individual's preferences. In addition, this approach facilitates the discovery of new places by leveraging the influence of friends and communities, thereby encouraging the exploration of unfamiliar destinations. Finally, by analyzing local trends and the crowd levels of places, it adjusts recommendations based on temporal and environmental context, providing a more immersive and relevant experience.

However, this approach raises several challenges and limitations. One of the main concerns involves social data privacy, as using user relationships and interactions may lead to ethical questions regarding personal data protection. In addition, the complexity of the models is another major challenge, since integrating multiple contextual dimensions (social, temporal, geographic) increases the computational load and requires sophisticated algorithms. Another risk is related to social bias, where recommendations may be overly influenced by a user's limited social circle, thereby reducing suggestion diversity. Finally, the cold-start problem remains a significant issue: in the absence of established social links or interaction history, it becomes difficult to provide accurate recommendations to new users. Despite these challenges, leveraging social context remains a promising direction to refine and enhance POI recommendation, especially when combined with other approaches such as collaborative filtering and deep learning.

I.4. Evaluation of Social Recommender Systems in LBSNs

The assessment of POI recommendation systems that incorporate social trust involves both quantitative and qualitative measures. These metrics facilitate the examination of recommendation accuracy, the pertinence of suggested locations, and the influence of social connections within the recommendation process. Common evaluation strategies for recommendations produced in Location-Based Social Networks (LBSNs) generally include an analysis of these aspects.

I.4.1. Classical Evaluation Metrics for POI Recommender Systems

Classical metrics are used to measure the system's ability to recommend relevant POIs. These metrics evaluate the system's capacity to correctly predict the POIs that a user might like. For instance, the parameter Precision@K measures the proportion of POIs that are actually appreciated among the top K recommendations (Werneck et al., 2021), as shown in Equation (2) below.

Equation 2: Top-K Precision for Evaluating Recommendation Accuracy

$$Precision@K = \frac{|POI_{recommended} \cap POI_{visited}|}{K}$$

On the other hand, the Recall@K parameter indicates the proportion of relevant POIs that have been recommended out of all the possible relevant POIs, as shown in Equation (3) below (D. Li et al., 2020).

Equation 3: Recall@K: Measuring the Coverage of Relevant POIs in Top-K Recommendations

$$Recall@K = \frac{|POI_{recommended} \cap POI_{\{visited\}}|}{|POI_{\{visited\}}|}$$

Finally, the F1-Score provides a balanced measure that combines the Precision and Recall metrics, as shown in Equation (4) below (Tamm et al., 2021), (A. Sun, 2023).

Equation 4: F1@K: Harmonic Mean of Precision and Recall for Top-K POI Recommendations

$$F1@K = 2 * \frac{(Precision@K * Recall@K)}{(Precision@K + Recall@K)}$$

1.4.2. Ranking Metrics for Recommendations

These metrics evaluate how the system ranks the POIs in its recommendations. For example, the NDCG (Normalized Discounted Cumulative Gain) parameter assesses the relevance order of the recommendations by giving more weight to POIs that are higher in the list, as shown in Equation (5) below (Jeunen et al., 2024), (R. Z. Li et al., 2021).

Equation 5: NDCG@K: Normalized Discounted Cumulative Gain for Ranking Quality in Top-K Recommendations

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

Where:

- **DCG@K** is the cumulative gain at rank K, adjusted by a logarithmic discount based on the rank of the items, as shown in Equation (6) below (Rampisela et al., 2023).

Equation 6: DCG@K: Discounted Cumulative Gain for Evaluating Ranking Quality

$$DCG@K = \sum_{i=1}^{K} \frac{rel_i}{log_2(i+1)}$$

Where:

- rel_i : represents the relevance value of the item located at the *i-th* position in the ranked list.
- $log_2(i+1)$: introduces a logarithmic discount that reduces the contribution of items appearing lower in the ranking, reflecting their diminished impact.

- **IDCG@K** is the maximum possible cumulative gain at rank K, which is computed by sorting the items based on the highest relevance scores, as shown in Equation (7) below (Burges et al., 2005).

Equation 7: IDCG@K: Ideal Discounted Cumulative Gain for Optimal Ranking Evaluation

$$IDCG@K = \sum_{i=1}^{|REL_K|} \frac{rel_i}{log_2(i+1)}$$

Where:

- $|REL_K|$: the number of relevant items among the top K (usually equals K if we assume we always evaluate K items).
- rel_i : the sorted relevance scores in descending order; this represents the ideal ranking.

Moreover, the Mean Reciprocal Rank (MRR) metric evaluates the effectiveness of recommendations by computing the reciprocal of the rank at which the first relevant POI appears in the result list, as shown in Equation (8) below (Moriya & Jones, 2023), (Schwartz, 2021).

Equation 8: MRR: Mean Reciprocal Rank for Evaluating the First Relevant Recommendation

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

1.4.3. Metrics Specific to Trust-Based Systems

Because these systems make use of social connections among users, tailored evaluation metrics are essential to assess how social networks influence the quality of recommendations.

One such metric, Social Trust Precision, examines whether the suggested POIs align with those previously visited by users considered trustworthy, as shown in Equation (9) below (Y. Zhang & Pennacchiotti, 2013).

Equation 9: STP@K: Socially-Trusted Precision in Top-K Recommendations

$$STP@K = \frac{|POI_{\{recommended\}} \cap POI_{\{visited\_by\_friends\}}|}{K}$$

Additionally, the Social Recall parameter measures the proportion of relevant POIs recommended by considering the preferences of friends, as shown in Equation (10) below (Davtalab & Alesheikh, 2021).

Equation 10: STR@K: Social Trust Recall at K for POI Recommendations

$$STR@K = \frac{|POI_{\{recommended\}} \cap POI_{\{visited\_by\_friends\}}|}{|POI_{\{visited\_by\_friends\}}|}$$

Finally, to measure the quality of trust propagation, the Propagation Influence Score parameter measures the impact of indirect recommendations (e.g., recommendations of a POI visited by a friend of a friend), and the Graph Influence Ratio parameter analyzes the propagation of recommendations within the social graph of LBSN users.

1.4.4. Experimental Evaluation Approaches

A. Offline Evaluation

This approach involves testing the system on a historical dataset where user-POI interactions and social links are known (Hidasi & Czapp, 2023), (Aouali et al., 2022). The steps are as follows:

1. **Data Partitioning**: Split the check-ins into training (80%) and testing (20%) sets.
2. **Social Graph Construction**: Define the friendship and trust links between users.
3. **Recommendation Generation**: Compare with the actual check-ins of users in the test data.
4. **Metric Calculation**: Use both classical and social metrics to assess the quality of the recommendations.

B. Online Evaluation (A/B Testing)

Online evaluation is conducted through real user experimentation on a recommendation platform (Kasalický et al., 2023), (Y. Wang & Ba, 2023), (D. Xu et al., 2021).

1. **Group A (Reference)**: Receives classic recommendations based solely on check-ins.
2. **Group B (Test - Social Trust)**: Receives recommendations that integrate social trust.
3. **Behavior Analysis**: Compare click-through rates, actual visit rates, and user satisfaction.

C. User Studies and Qualitative Experiences

Surveys and questionnaires can be conducted to measure the perceived user experience (Pu et al., 2011), (Zangerle & Bauer, 2023).

- **Satisfaction Rate**: Measures user satisfaction with the recommendations received.
- **Sentiment Analysis**: Analyzes user comments and reviews to evaluate whether users found the recommendations relevant.

1.4.5. Discussions

The evaluation of social trust-based POI recommendation systems combines classic metrics (Precision, Recall, NDCG) with metrics specific to social interactions (social consistency, trust propagation, recommendation diversity). Evaluation approaches include offline tests based on real datasets (e.g., Foursquare, Gowalla, Yelp), online experiments (A/B

Testing), and user studies to measure the perceived effectiveness of the system. The goal is to ensure that recommendations leverage social trust to improve the relevance, diversity, and acceptability of the POIs suggested to users of LBSNs.

I.5. Conclusion

This chapter introduced the fundamental concepts related to LBSNs, POI recommendation, the integration of social trust, and the evaluation of POI predictions. Understanding these concepts is essential for developing an effective recommendation system that leverages the trust relationships between users. The following chapters will delve deeper into these concepts by exploring the methodologies and approaches adopted to improve the accuracy and relevance of recommendations.

Chapter II:

The Problem of Social Trust in LBSNs

II.1. Introduction

Location-Based Social Networks (LBSNs) provide personalized recommendations based on social interactions and location data (H. Wang et al., 2013). However, the integration of social trust into these systems presents several challenges. Trust between users plays a crucial role in the quality and relevance of recommendations, but it is complex to model and utilize effectively. This chapter explores the various issues related to social trust in LBSNs, its determinants, and its implications for the recommendation of Points of Interest (POIs).

II.2. Definition and Importance of Social Trust in LBSNs

Social trust refers to the perceived level of reliability, honesty, or relevance between users (Shareef et al., 2020) in a social network . In the context of Location-Based Social Networks (LBSNs), it plays a crucial role in how users share, evaluate, and recommend Points of Interest (POIs) (Bao et al., 2015). Social trust reflects the extent to which one user believes that another user's behavior or preferences are dependable, even without a personal connection. For instance, if User A follows User B and notices that they frequently check in at places User A also enjoys, this can foster a sense of trust in User B's future visits or recommendations.

Social trust can be categorized as either explicit, such as direct friendships, following relationships, or endorsements, or implicit, which is inferred from shared interactions, similar reviews, or overlapping check-in histories (Ouyang et al., 2016) (Demirci & Karagoz, 2022). This trust significantly influences recommendation relevance, as users tend to give more weight to suggestions coming from trusted friends or like-minded individuals.

II.3. Challenges of Social Trust in LBSNs

Integrating social trust into POI recommendation systems introduces several key challenges:

II.3.1 Trust Modeling Problem

Trust between users is a subjective and dynamic concept (Tong et al., 2013). It is difficult to define a precise mathematical model that captures the multiple dimensions of trust (e.g., interest similarity, frequency of interactions, reliability of reviews) (Tang et al., 2013), (Massa & Avesani, 2007).

II.3.2 Limited Data and Sparsity Problem

Most users interact with a limited number of friends and POIs, resulting in a very sparse trust matrix. This low density complicates the use of trust in recommendation algorithms (Ye et al., 2011), (J.-D. Zhang & Chow, 2015).

II.3.3 Trust Propagation and Social Influence

Some models attempt to infer trust between users who do not know each other directly (e.g., through friends of friends) (Pal & Jenamani, 2019). However, the diminishing effect of trust across indirect connections makes this propagation more complex (Ma et al., 2011), (B. Yang et al., 2017).

II.3.4 Manipulation and Review Reliability

Social trust-based systems are vulnerable to attacks such as fake reviews, fake check-ins, or the manipulation of the popularity of certain POIs (Mukherjee et al., 2021), (Ott et al., 2011).

II.3.5 Privacy and Data Protection

Leveraging social trust often involves analyzing user interactions and personal data, which raises privacy and ethical concerns (Shokri et al., 2011), (Krumm, 2009).

II.4. Modeling and Evaluation of Social Trust

Several methods have been developed to integrate social trust into POI recommendation systems:

II.4.1 Graph-Based Trust Approaches

Social relationships can be modeled as graphs, where users are nodes and the links represent the level of trust. Propagation algorithms (e.g., Social PageRank) are used to weight recommendations based on the degree of trust (Golbeck, 2006), (Zhou et al., 2010).

II.4.2 Trust-Based Collaborative Filtering

Adding trust factors to classic collaborative filtering models improves the relevance of recommendations by prioritizing interactions between trusted users (Ma et al., 2009), (G. Guo et al., 2015).

II.4.3 Hybrid Models

Some systems combine social trust with other recommendation methods (collaborative filtering, deep learning) to reduce the impact of sparsity and review manipulation problems (B. Liu et al., 2013).

II.4.4 Deep Learning and Social Trust

Graph Neural Networks (GNNs) and Autoencoders are used to capture latent representations of social trust and improve the accuracy of recommendations (S. Zhang et al., 2020), (M. Li et al., 2024), (Zhao et al., 2014).

The evaluation of these previously mentioned social trust-based recommendation systems relies on several indicators:

- **Precision and Recall (@K)**: Measure the system's ability to recommend POIs actually visited by users (Herlocker et al., 2004).
- **Social Trust Precision**: Indicates whether the recommended POIs are influenced by trusted users.
- **Graph Influence Ratio**: Analyzes the propagation of recommendations within the social graph (X. Wang et al., 2019).
- **User Satisfaction**: Measures the acceptability of recommendations through qualitative surveys.

II.5. O'DONOVAN Approach and Collaborative Filtering

The approach proposed by O'Donovan and Smyth (O'Donovan & Smyth, 2005) falls within the scope of collaborative filtering, integrating the notion of implicit trust to enhance the quality of recommendations. Unlike traditional collaborative filtering methods, which rely solely on similarity measures between users (e.g., Pearson, Cosine), this approach introduces a mechanism for evaluating the reliability of recommendations. It assigns a trust score to users based on the consistency and accuracy of their past predictions. Thus, recommendations are no longer based only on profile similarity but also on the trustworthiness of information sources, helping to reduce errors caused by biased or irrelevant ratings. This approach marks a significant advance in mitigating data sparsity and cold-start problems, by promoting interactions between trusted users.

O'Donovan and Smyth distinguish between two types of profiles in the context of a recommendation session or a rating prediction task:

- The *consumer* refers to the profile receiving the predicted rating for an item.
- The *producer* refers to the profile selected as a recommendation partner for the consumer and involved in the recommendation session.

To generate a predicted rating for item *i* for a consumer *c*, O'Donovan and Smyth typically rely on several producer profiles. Their individual recommendations are then aggregated using an appropriate function, such as the Resnick formula (see Equation (11)).

Until now, collaborative filtering systems have largely relied on what can be called the similarity hypothesis: that similar profiles (in terms of rating history) make good recommendation partners.

The reference algorithm of the O'Donovan and Smyth approach uses the standard Resnick prediction formula (Resnick et al., 1994), reproduced below as Equation (11):

- *c(i)* represents the predicted rating for item *i* in the consumer profile *c*.
- *p(i)* denotes the rating assigned to item *i* by a producer profile *p*.
- Additionally, $\bar{c}$ and $\bar{p}$ denote the average ratings of profiles *c* and *p*, respectively.

The weighting factor $sim(c,p)$ is a measure of similarity between profiles *c* and *p*, traditionally calculated using the Pearson correlation coefficient.

Equation 11: Rating Prediction (Resnick's prediction formula)

$$c(i) = \bar{c} + \frac{\sum_{p \in P(i)} (p(i) - \bar{p}) * sim(c,p)}{\sum_{p \in P(i)} |sim(c,p)|}$$

This benchmark is used because it allows for easy comparison with existing systems. Moreover, Resnick's prediction formula reduces the influence of a partner's prediction based on their degree of similarity with the target user, so that the most similar partners have a greater impact on the final rating prediction.

O'Donovan and Smyth assume that profile similarity is only one of many factors that can influence recommendation and prediction, then the reliability of a partner profile in providing accurate recommendations in the past constitutes another important factor, which they call trust.

Intuitively, if a profile has made many accurate predictions in the past, it can be considered more reliable than another profile that has made many erroneous predictions. In what follows, O'Donovan and Smyth define two trust models and show how they can be easily integrated into the operation of a standard collaborative filtering recommendation system.

II.5.1. Profile-Level & Item-Level Trust

A rating prediction *p(i)* made by producer *p* for item *i* is deemed accurate if it falls within a margin of ε from the actual rating *c(i)* assigned by the consumer *c*, as illustrated in Equation (12).

Of course, when a producer participates in the recommendation process, they are usually accompanied by several other recommendation partners. Therefore, it can be difficult to determine whether the accuracy of the final recommendation is due to the contribution of p. Consequently, to evaluate the accuracy of p's recommendation, we perform the recommendation process separately using p as the sole recommendation partner for c.

Equation (13) and Figure 3 illustrate how each triplet (i, p, c) yields a binary outcome, either success or failure, based on whether the predicted rating falls within ε of the rating given by the consumer to the item.

In a real-time recommendation system, producer trust values could be easily generated on the fly by comparing the predicted rating (based solely on a producer's profile) with the actual rating entered by the user.

Equation 12: Rating Prediction Validity Within Error Threshold

$$Correct(i, p, c) \Leftrightarrow |p(i) - c(i)| < \varepsilon$$

Equation 13: Binary Evaluation of Prediction Correctness

$$T_p(i, c) = Correct(i, p, c)$$

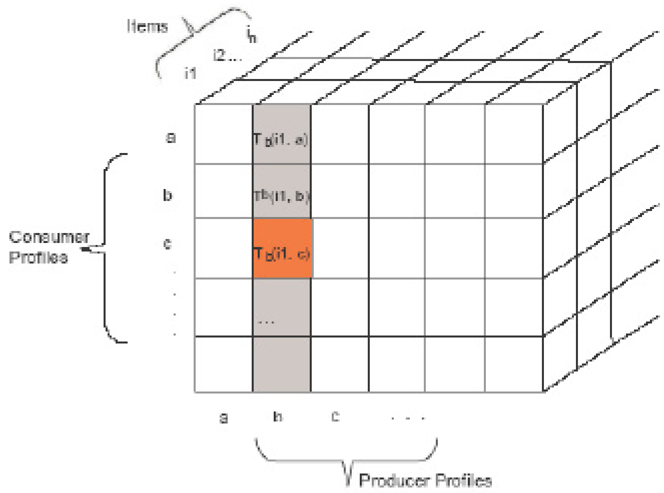

Figure 3: Calculation of Trust Scores from Rating Data (O'Donovan & Smyth, 2005)

From this, two basic trust metrics are defined, based on the relative number of correct recommendations made by a given producer.

The full set of recommendations in which a given producer was involved, $RecSet(p)$, is defined by Equation (14).

The subset of those recommendations that are correct, $CorrectSet(p)$, is defined by Equation (15).

The values i represent the items, and the values c correspond to the predicted ratings.

Equation 14: Definition of the Recommendation Set for a Producer

$$RecSet(p) = \{(c_1, i_1), \ldots, (c_n, i_n)\}$$

Equation 15: Definition of the Correct Recommendation Subset for a Producer

$$CorrectSet(p) = \{(c_k, i_k) \in RecSet(p) : Correct(i_k, p, c_k)\}$$

The profile-level trust, $Trust^P$, for a producer corresponds to the percentage of correct recommendations that the producer has contributed to, see Equation (16).

For example, if a producer participated in 100 recommendations, meaning they served as a recommendation partner 100 times, and for 40 of those recommendations they were able to predict a correct rating, then their profile-level trust score is 0,4.

Equation 16: Profile-Level Trust for a Producer

$$Trust^P(p) = \frac{|CorrectSet(p)|}{|RecSet(p)|}$$

Obviously, profile-level trust is a rather coarse measure of trust, as it applies to the entire profile.

In reality, a producer's profile may be more reliable for predicting ratings of certain items than others.

Therefore, we can define a finer-grained trust metric at the item-level, $Trust^I$, as shown in Equation (17). This metric measures the percentage of correct recommendations for a given item i.

Equation 17: Item-Level Trust for a Producer

$$Trust^I(p, i) = \frac{|\{(c_k, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_k, i_k) \in RecSet(p) : i_k = i\}|}$$

II.5.2. Trust-Based Recommendation

After estimating a profile's trust (or its trust with respect to a specific item), this trust needs to be integrated into the recommendation process.

The simplest approach is to adopt Resnick's prediction strategy (see Equation (11)).

Then, two adaptations are considered: trust-based weighting and trust-based filtering. These two approaches can be used with either profile-level or item-level trust metrics. In our case, the Resnick formula is used.

II.5.2.1. Trust-Based Weighting

The simplest way to integrate trust into the recommendation process is perhaps to combine trust and similarity to produce a composite weighting factor, which can be used in the Resnick formula (see Equation (18)).

Equation 18: Resnick Formula Adapted for Trust-based Weighting

$$c(i) = \bar{c} + \frac{\sum_{p \in P(i)} (p(i) - \bar{p}) * w(c,p,i)}{\sum_{p \in P(i)} |w(c,p,i)|}$$

For example, when predicting the rating of item *i* for consumer *c*, we could compute the arithmetic mean of the trust value (either at the profile or item level) and the similarity value for each producer profile.

Alternatively, this approach can be modified by using the harmonic mean of trust and similarity, see Equation (19), which combines profile similarity with item-level trust in this case.

Equation 19: The Harmonic Mean of Trust and Similarity

$$w(c,p,i) = \frac{2(sim(c,p))(Trust^I(p,i))}{sim(c,p) + Trust^I(p,i)}$$

The advantage of using the harmonic mean is that it is robust to large differences between inputs, meaning a high weight will only be computed if both trust and similarity scores are high.

The harmonic mean method may be preferred over techniques such as addition, subtraction, or multiplication because it produced the best results during their preliminary optimization tests.

II.5.2.2. Trust-Based Filtering

Trust can also be considered as a way to filter profiles before making recommendations, so that only the most trustworthy profiles are allowed to participate in the prediction process.

For example, Equation (20) presents a modified version of Resnick's formula, where producer profiles are included in the recommendation process only if their trust value exceeds a predefined threshold.

Equation 20: Resnick Formula Adapted for Trust-Based Filtering

$$c(i) = \bar{c} + \frac{\sum_{p \in P^T(i)} (p(i) - \bar{p}) * sim(c,p)}{\sum_{p \in P^T(i)} |sim(c,p)|}$$

Equation (21) illustrates this approach using item-level trust ($Trust^{I}(p,i)$), but it can easily be adapted to use profile-level trust instead. In this way, the standard Resnick method is applied only to the most trustworthy profiles.

Equation 21: Filtering Producers by Item-Level Trust Threshold

$$P_i^T = \{p \in P(i) : Trust^{I}(p,i) > T\}$$

II.5.2.3. Combining Trust-Based Weighting & Filtering

The Trust-Based Weighting and Trust-Based Filtering approaches can be combined to improve the accuracy of recommendations. First, profiles are filtered based on their trust level, allowing only the most dependable users to be included. In the prediction stage, these users' trust scores are then combined with their similarity to the target consumer, which helps refine and improve the relevance of the recommendations.

In the following, Equation (22) illustrates this approach by combining filtering and weighting using item-level trust ($Trust^{I}$).

Equation 22: Resnick Formula Adapted for Combining Trust-Based Filtering and Weighting

$$c(i) = \bar{c} + \frac{\sum_{p \in P^{T}(i)} (p(i) - \bar{p}) * w(c,p,i)}{\sum_{p \in P^{T}(i)} |w(c,p,i)|}$$

II.6. Trust Propagation

Trust propagation in a location-based social network (LBSN) is an approach that leverages both implicit and explicit trust relationships between users to enhance the recommendation of points of interest (POIs). In these networks, trust is not limited to direct connections but can be transmitted through transitive relationships. Consider the following scenario: if user A places trust in user B, and user B in turn trusts user C, it follows that user A might also develop a certain level of trust towards user C, even without direct interaction. This mechanism broadens the scope of recommendations by utilizing indirect social links, thereby reducing data sparsity and mitigating the cold-start problem.

II.6.1. Trust Score Calculation through Propagation

By integrating trust propagation into the recommendation process, systems become more robust and personalized, taking into account users' social affinities and implicit behaviors. Due to the large number of items in a recommendation system, the rating matrix tends to be very sparse. This sparsity often results in two users having no commonly rated items, which leads to the absence of direct trust relationships between them.

The issue can be resolved through trust propagation, which enables the inference of indirect relationships. Trust propagation assumes that within the trust network, there exists a trust path between a source user u_s and a target user u_t.

Suppose there is an intermediate user u_m on the trust path linking u_s to u_t. The inferred trust score of u_t from the perspective of u_s via u_m is calculated as the weighted average of the two direct trust relationships: us → u_m and u_m → u_t (Hwang & Chen, 2007), as shown in Equation (23).

Equation 23: Weighted Trust Inference Through an Intermediate Node

$$t_{s\to t} = t_{s\to m} \oplus t_{m\to t} = \frac{n(I_s \cap I_m)t_{s\to m} + n(I_m \cap I_t)t_{m\to t}}{n(I_s \cap I_m) + n(I_m \cap I_t)}$$

The logic behind this calculation is that the more co-rated items two users have, the more reliable their direct relationship is, and thus it deserves a higher weight. The propagation operator can be applied repeatedly to compute the indirect trust relationship between any two users in the trust network.

II.6.2. Path Composition

It is possible that multiple paths exist between two users in the trust network. Each path contributes its own inferred trust score, and these scores are considered independent of one another. To combine these trust scores into a single composite measure, one common method is to calculate the average of all inferred trust scores provided by each alternative path.

In the following section, an example is presented to illustrate this method.

Table 7: Example of User-Item Rating Matrix

UIRM	i1	i2	i3	i4	i5	i6	$\overline{r_{ux}}$
u1		5	2		3		10/3
u2	4			3		4	11/3
u3		1	2			2	5/3
u4	5			3	2		10/3
u5		5	5			3	13/3

Using the previously mentioned Equation (11), we estimate the predicted rating for item *i* by consumer user *c*, inferred through their producer *p*.

For example, PredictedRate(u1, u3, i2) = meanRate(u1) + Rate(u3,i2) – meanRate(u3) = 8/3

After applying Equation (11) on the entire User-Item Rating Matrix (see Table 7), we obtain the matrix UPRM_Predicted which contains the predicted ratings below (see Table 8):

Table 8: The matrix UPRM_Predicted containing the predicted ratings

UPRM_Predicted	i1		i2		i3		i4		i5		i6	
u1	u1	?	u1	5	u1	2	u1	?	u1	3	u1	?
	u2	?	u2	?	u2	?	u2	?	u2	?	u2	?
	u3	?	u3	8/3	u3	11/3	u3	?	u3	?	u3	?
	u4	?	u4	?	u4	?	u4	?	u4	2	u4	?
	u5	?	u5	4	u5	4	u5	?	u5	?	u5	?
u2	u1	?	u1	?	u1	?	u1	?	u1	?	u1	?
	u2	4	u2	?	u2	?	u2	3	u2	?	u2	4
	u3	?	u3	?	u3	?	u3	?	u3	?	u3	4
	u4	16/3	u4	?	u4	?	u4	10/3	u4	?	u4	?
	u5	?	u5	?	u5	?	u5	?	u5	?	u5	7/3
u3	u1	?	u1	10/3	u1	1/3	u1	?	u1	?	u1	?
	u2	?	u2	?	u2	?	u2	?	u2	?	u2	2
	u3	?	u3	1	u3	2	u3	?	u3	?	u3	2
	u4	?	u4	?	u4	?	u4	?	u4	?	u4	?
	u5	?	u5	7/3	u5	7/3	u5	?	u5	?	u5	1/3
u4	u1	?	u1	?	u1	?	u1	?	u1	3	u1	?
	u2	11/3	u2	?	u2	?	u2	8/3	u2	?	u2	?
	u3	?	u3	?	u3	?	u3	?	u3	?	u3	?
	u4	5	u4	?	u4	?	u4	3	u4	2	u4	?
	u5	?	u5	?	u5	?	u5	?	u5	?	u5	?
u5	u1	?	u1	18/3	u1	3	u1	?	u1	?	u1	?
	u2	?	u2	?	u2	?	u2	?	u2	?	u2	14/3
	u3	?	u3	11/3	u3	14/3	u3	?	u3	?	u3	14/3
	u4	?	u4	?	u4	?	u4	?	u4	?	u4	?
	u5	?	u5	5	u5	5	u5	?	u5	?	u5	3

Using the Equation (12) mentioned earlier, we can calculate the difference between the actual rating given by consumer *c* and the predicted rating by producer *p* for that consumer user *c*.

For example, to calculate the difference between the actual rating given by consumer *c*: u1 for item i2 and the predicted rating by producer *p*: u3 for the same item i2, we perform the following calculation:

$$| \text{ActualRate}(u1, i2) - \text{PredictedRate}(u1, u3, i2) | = | 5 – 8/3 | = 7/3 < \varepsilon$$

After applying Equation (12) to the entire matrix UPRM_Predicted, we obtain DeviationMatrix below, which contains the prediction accuracy deviations (see Table 9).

Table 9: DeviationMatrix containing the differences between the actual ratings and the predicted ratings

<table>
<tr><th>DeviationMatrix</th><th colspan="2">i1</th><th colspan="2">i2</th><th colspan="2">i3</th><th colspan="2">i4</th><th colspan="2">i5</th><th colspan="2">i6</th></tr>
<tr><td rowspan="5">u1</td><td>u1</td><td>?</td><td>u1</td><td>|5-5|</td><td>u1</td><td>|2-2|</td><td>u1</td><td>?</td><td>u1</td><td>|3-3|</td><td>u1</td><td>?</td></tr>
<tr><td>u2</td><td>?</td><td>u2</td><td>|5-?|</td><td>u2</td><td>|2-?|</td><td>u2</td><td>?</td><td>u2</td><td>|3-?|</td><td>u2</td><td>?</td></tr>
<tr><td>u3</td><td>?</td><td>u3</td><td>|5-8/3|</td><td>u3</td><td>|2-11/3|</td><td>u3</td><td>?</td><td>u3</td><td>|3-?|</td><td>u3</td><td>?</td></tr>
<tr><td>u4</td><td>?</td><td>u4</td><td>|5-?|</td><td>u4</td><td>|2-?|</td><td>u4</td><td>?</td><td>u4</td><td>|3-2|</td><td>u4</td><td>?</td></tr>
<tr><td>u5</td><td>?</td><td>u5</td><td>|5-4|</td><td>u5</td><td>|2-4|</td><td>u5</td><td>?</td><td>u5</td><td>|3-?|</td><td>u5</td><td>?</td></tr>
<tr><td rowspan="5">u2</td><td>u1</td><td>|4-?|</td><td>u1</td><td>?</td><td>u1</td><td>?</td><td>u1</td><td>|3-?|</td><td>u1</td><td>?</td><td>u1</td><td>|4-?|</td></tr>
<tr><td>u2</td><td>|4-4|</td><td>u2</td><td>?</td><td>u2</td><td>?</td><td>u2</td><td>|3-3|</td><td>u2</td><td>?</td><td>u2</td><td>|4-4|</td></tr>
<tr><td>u3</td><td>|4-?|</td><td>u3</td><td>?</td><td>u3</td><td>?</td><td>u3</td><td>|3-?|</td><td>u3</td><td>?</td><td>u3</td><td>|4-4|</td></tr>
<tr><td>u4</td><td>|4-16/3|</td><td>u4</td><td>?</td><td>u4</td><td>?</td><td>u4</td><td>|3-10/3|</td><td>u4</td><td>?</td><td>u4</td><td>|4-?|</td></tr>
<tr><td>u5</td><td>|4-?|</td><td>u5</td><td>?</td><td>u5</td><td>?</td><td>u5</td><td>|3-?|</td><td>u5</td><td>?</td><td>u5</td><td>|4-7/3|</td></tr>
<tr><td rowspan="5">u3</td><td>u1</td><td>?</td><td>u1</td><td>|1-10/3|</td><td>u1</td><td>|2-1/3|</td><td>u1</td><td>?</td><td>u1</td><td>?</td><td>u1</td><td>|2-?|</td></tr>
<tr><td>u2</td><td>?</td><td>u2</td><td>|1-?|</td><td>u2</td><td>|2-?|</td><td>u2</td><td>?</td><td>u2</td><td>?</td><td>u2</td><td>|2-6/3|</td></tr>
<tr><td>u3</td><td>?</td><td>u3</td><td>|1-1|</td><td>u3</td><td>|2-2|</td><td>u3</td><td>?</td><td>u3</td><td>?</td><td>u3</td><td>|2-2|</td></tr>
<tr><td>u4</td><td>?</td><td>u4</td><td>|1-?|</td><td>u4</td><td>|2-?|</td><td>u4</td><td>?</td><td>u4</td><td>?</td><td>u4</td><td>|2-?|</td></tr>
<tr><td>u5</td><td>?</td><td>u5</td><td>|1-7/3|</td><td>u5</td><td>|2-7/3|</td><td>u5</td><td>?</td><td>u5</td><td>?</td><td>u5</td><td>|2-1/3|</td></tr>
<tr><td rowspan="5">u4</td><td>u1</td><td>|5-?|</td><td>u1</td><td>?</td><td>u1</td><td>?</td><td>u1</td><td>|3-?|</td><td>u1</td><td>|2-3|</td><td>u1</td><td>?</td></tr>
<tr><td>u2</td><td>|5-11/3|</td><td>u2</td><td>?</td><td>u2</td><td>?</td><td>u2</td><td>|3-8/3|</td><td>u2</td><td>|2-?|</td><td>u2</td><td>?</td></tr>
<tr><td>u3</td><td>|5-?|</td><td>u3</td><td>?</td><td>u3</td><td>?</td><td>u3</td><td>|3-?|</td><td>u3</td><td>|2-?|</td><td>u3</td><td>?</td></tr>
<tr><td>u4</td><td>|5-5|</td><td>u4</td><td>?</td><td>u4</td><td>?</td><td>u4</td><td>|3-3|</td><td>u4</td><td>|2-2|</td><td>u4</td><td>?</td></tr>
<tr><td>u5</td><td>|5-?|</td><td>u5</td><td>?</td><td>u5</td><td>?</td><td>u5</td><td>|3-?|</td><td>u5</td><td>|2-?|</td><td>u5</td><td>?</td></tr>
<tr><td rowspan="5">u5</td><td>u1</td><td>?</td><td>u1</td><td>|5-18/3|</td><td>u1</td><td>|5-3|</td><td>u1</td><td>?</td><td>u1</td><td>?</td><td>u1</td><td>|3-?|</td></tr>
<tr><td>u2</td><td>?</td><td>u2</td><td>|5-?|</td><td>u2</td><td>|5-?|</td><td>u2</td><td>?</td><td>u2</td><td>?</td><td>u2</td><td>|3-14/3|</td></tr>
<tr><td>u3</td><td>?</td><td>u3</td><td>|5-11/3|</td><td>u3</td><td>|5-14/3|</td><td>u3</td><td>?</td><td>u3</td><td>?</td><td>u3</td><td>|3-14/3|</td></tr>
<tr><td>u4</td><td>?</td><td>u4</td><td>|5-?|</td><td>u4</td><td>|5-?|</td><td>u4</td><td>?</td><td>u4</td><td>?</td><td>u4</td><td>|3-?|</td></tr>
<tr><td>u5</td><td>?</td><td>u5</td><td>|5-5|</td><td>u5</td><td>|5-5|</td><td>u5</td><td>?</td><td>u5</td><td>?</td><td>u5</td><td>|3-3|</td></tr>
</table>

Based on DeviationMatrix and Equations (12) and (13), we calculate a binary success or failure score that estimates the closeness between the actual rating given by the consumer (denoted as c) and the rating generated by a producer user (denoted as p) for a given item (denoted as i) as follows.

$$Correct(i,p,c) \Leftrightarrow |\ \text{PredictedRate}(i, p, c) - \text{ActualRate}(i, c)\ | < \varepsilon$$

$$\text{IF } |c_i - p_i| < \varepsilon \text{ THEN } Correct(i,p,c) = 1 \text{ ELSE } Correct(i,p,c) = 0$$

If this score $|c_i - p_i|$ is less than ε (a precision parameter), the function $Correct(i,p,c)$ is equal to 1 (good precision). Otherwise, if this score is greater than or equal to ε, this same function is equal to 0. In this way, we obtain the SuccessFailureMatrix below, which contains the success and failure scores of the predictions (see Table 10):

Table 10: The SuccessFailureMatrix containing the success and failure scores of the predictions

SuccessFailureMatrix	i1		i2		i3		i4		i5		i6	
u1	u1	?	u1	1	u1	1	u1	?	u1	1	u1	?
	u2	?	u2	?	u2	?	u2	?	u2	?	u2	?
	u3	?	u3	0	u3	0	u3	?	u3	?	u3	?
	u4	?	u4	?	u4	?	u4	?	u4	1	u4	?
	u5	?	u5	1	u5	0	u5	?	u5	?	u5	?
u2	u1	?	u1	?	u1	?	u1	?	u1	?	u1	?
	u2	1	u2	?	u2	?	u2	1	u2	?	u2	1
	u3	?	u3	?	u3	?	u3	?	u3	?	u3	1
	u4	0	u4	?	u4	?	u4	1	u4	?	u4	?
	u5	?	u5	?	u5	?	u5	?	u5	?	u5	0
u3	u1	?	u1	0	u1	0	u1	?	u1	?	u1	?
	u2	?	u2	?	u2	?	u2	?	u2	?	u2	1
	u3	?	u3	1	u3	1	u3	?	u3	?	u3	1
	u4	?	u4	?	u4	?	u4	?	u4	?	u4	?
	u5	?	u5	0	u5	1	u5	?	u5	?	u5	0
u4	u1	?	u1	?	u1	?	u1	?	u1	1	u1	?
	u2	0	u2	?	u2	?	u2	1	u2	?	u2	?
	u3	?	u3	?	u3	?	u3	?	u3	?	u3	?
	u4	1	u4	?	u4	?	u4	1	u4	1	u4	?
	u5	?	u5	?	u5	?	u5	?	u5	?	u5	?
u5	u1	?	u1	1	u1	0	u1	?	u1	?	u1	?
	u2	?	u2	?	u2	?	u2	?	u2	?	u2	0
	u3	?	u3	0	u3	1	u3	?	u3	?	u3	0
	u4	?	u4	?	u4	?	u4	?	u4	?	u4	?
	u5	?	u5	1	u5	1	u5	?	u5	?	u5	1

To calculate the trust at the profile-level, $Trust^P$, based on Equation (16), we use Equations (14) and (15) mentioned earlier. These two formulas allow us to find the subsets $RecSet(User)$ and $CorrectSet(User)$.

For example, to calculate how many times $u5$ has participated in predictions for the system's users, and how many correct predictions $u5$ made out of the total number where $u5$ was involved:

Using Equation (14), we can calculate $RecSet(u5)$ as follows:

$$RecSet(u5) = \{(4, i_2), (4, i_3)\,, \left(\frac{7}{3}, i_6\right), \left(\frac{7}{3}, i_2\right), \left(\frac{7}{3}, i_3\right), \left(\frac{1}{3}, i_6\right)\}$$

Using Equation (15), we can calculate $CorrectSet(u5)$ as follows:

$$CorrectSet(u5) = \{(4, i_2), \left(\frac{7}{3}, i_3\right)\}$$

Using Equation (16), we can calculate $Trust^P(u5)$ as follows:

$$|CorrectSet(u5)| = 2$$

$$|RecSet(u5)| = 6$$

$$Trust^P(u5) = \frac{2}{6} = \frac{1}{3} = 0.33$$

In the same way, we calculate the set of values of the profile-level trust matrix, $Trust^P$, for all the other users (producers) in the system. This leads to the following Table 11.

Table 11: The matrix representing trust at the profile-level

user	$Trust^P$
u1	0.4
u2	0.5
u3	0.3
u4	0.6
u5	0.3

II.7. Conclusion

The exploitation of social trust in LBSNs (Location-Based Social Networks) is an effective strategy to improve the relevance of POI (Points of Interest) recommendations. However, several challenges remain, including trust modeling, the protection of personal data, and the management of manipulations. The integration of hybrid models and deep learning-based approaches presents a promising avenue for optimizing the personalization of recommendations while maintaining the reliability of social interactions.

Chapter III:

POI Recommendation Based on Social Trust in LBSNs

III.1. Introduction

Point of Interest (POI) recommendation in Location-Based Social Networks (LBSNs) has become a major research area due to the growing popularity of platforms such as Foursquare, Yelp, and Facebook Places. While the first chapter introduced preliminary notions; including geolocation concepts, classical recommendation models, and evaluation metrics; this chapter focuses on the state-of-the-art recommendation techniques that integrate the notion of social trust. The goal is to present existing approaches, their advantages and their limitations, as well as to identify challenges and future research directions.

III.2. LBSNs and POI Recommendation

LBSNs combine location data with social interactions, thereby offering a wealth of contextual information. In this context, POI recommendation aims to suggest places to users that are likely to match their preferences based on:

- **Geographical aspects**: proximity, visit frequency, local popularity.
- **Temporal aspects**: peak hours, seasonal trends.
- **Social aspects**: relationships between users, peer influence, and implicit recommendations derived from their behaviors.

The social dimension, particularly the notion of trust, has proven essential in addressing issues such as data sparsity and the cold-start problem. Indeed, incorporating trust derived from social interactions helps enhance the relevance of recommendations by leveraging indirect signals of user preferences.

III.3. Collaborative Models Integrating Social Filtering

Traditional collaborative approaches, such as collaborative filtering, have been extended to incorporate social trust measures. These models rely on the assumption that users have preferences similar to those of their friends or trusted contacts. Among the notable methods are:

III.3.1. Social regularization in matrix factorization

This method incorporates a trust-based regularization term, which brings closer the latent representations of users who share trust relationships. This enhances robustness in the face of data sparsity.

III.3.2. Trust propagation models

By leveraging social relationship networks, these approaches propagate trust information through the links of the social graph to estimate implicit similarities between users.

III.3.3. Trust Network Graphs

Social networks naturally lend themselves to graph-based modeling. In this context, several studies exploit:

III.3.3.1. Graph-based recommendation models

These models represent users, POIs, and trust relationships as nodes and edges, thus enabling the application of traversal algorithms (e.g., adapted PageRank) or graph-based learning techniques.

III.3.3.2. Random walk methods and neighborhood aggregation approaches:

These techniques allow the extraction of proximity and trust signals from the structure of the network, thus providing recommendations that take social topology into account.

III.3.3.3. Integration of Contextual and Hybrid Data

In addition to social aspects, POI recommendation requires consideration of a multitude of contextual factors. Hybrid approaches combine:

- **Geographical and temporal data**: for example, by using spatial distribution models or clustering methods to identify areas of interest.
- **Textual content and images**: analyzing content associated with POIs (such as reviews and descriptions) can enrich the profile of the locations and refine the measurement of trust.
- **Deep learning techniques**: architectures such as Convolutional Neural Networks (CNNs) or Graph Neural Networks (GNNs) are increasingly being used to fuse various sources of information into an end-to-end model.

These hybrid approaches not only enable a better capture of user preferences, but also allow for adjusting the impact of trust signals according to the context.

III.4. Discussions and Critical Analyses

Methods that incorporate social trust offer several advantages:

- **Improvement of accuracy**: By leveraging additional signals from social relationships, these methods are able to better predict user preferences.
- **Reduction of data sparsity**: Trust diffusion helps to address the lack of explicit data in certain cases (cold-start).
- **Adaptability**: The integration of multiple contextual data sources provides flexibility to model diverse user behaviors.

However, several challenges remain:

- **Trust data quality and noise**: Social interactions do not always represent true trust, which can introduce bias into the recommendation process.
- **Computational complexity**: The use of graph-based models or deep learning architectures can significantly increase computation time and system latency.
- **Privacy issues:** Using social and location-based data raises significant concerns regarding user privacy and data security.

These limitations highlight the need for robust approaches and noise-filtering mechanisms in order to improve the reliability of recommendations.

For these reasons, current research focused on integrating social trust into POI recommendation opens up several avenues for improvement:

- **Finer hybrid models:** the dynamic fusion of social, geographic, and temporal signals through attention mechanisms could enable even more personalized recommendations.
- **Use of advanced deep learning techniques:** Graph Neural Networks (GNNs) and Transformer-based models adapted to heterogeneous data show promise in better capturing the complex interactions between users and POIs.
- **Privacy-centered approaches:** Developing privacy-preserving recommendation models (e.g., through federated learning) is essential to meet regulatory requirements and user expectations.

III.5. State of the art on works integrating social trust

This study explores two trust-based approaches to improve the accuracy of recommendations: explicit trust, based on direct user statements, and implicit trust, inferred from their past behaviors. The analysis highlights the value of implicit trust, which reduces the need for active user participation and mitigates the cold-start problem caused by data sparsity. Artificial intelligence plays a key role in leveraging tourists' histories, analyzing their preferences and habits to recommend relevant places.

Implicit Trust-Based Recommender Systems (ITBRS) use past actions, such as ratings and check-ins, to establish trust relationships between users. These approaches, often based on collaborative filtering, rely on abundant implicit data such as clicks and viewing times. Probabilistic models, such as matrix factorization, help improve the effectiveness of recommendations. Recent developments include hybrid systems that combine implicit trust with explicit collaborative filtering, integrating contextual information.

This study compared explicit and implicit approaches, highlighting the advantage of the latter in addressing data sparsity. Finally, it emphasizes the potential of trust propagation to overcome these limitations and proposes new AI-based perspectives to optimize tourist recommendations.

III.5.1. Recommender Systems Based on Explicit Trust

Studies based on explicit trust declarations between users are generally classified according to several criteria, including the method used, the dataset employed, the evaluation metrics applied, and whether or not the principle of trust propagation is adopted (see Table 12).

Golbeck and Hendler (Golbeck & Hendler, 2006) developed FilmTrust, a platform combining social networks and the Semantic Web to integrate User Trust (UT) into movie recommendations. Explicit trust ratings are leveraged to compute similarities between users. Then, a comparison is drawn between the user's actual rating, the average rating of the movie, and the rating predicted by the automatic collaborative filtering (ACF) algorithm. The prediction evaluation is also carried out by comparing these results with those obtained using the nearest neighbor algorithm based on Pearson correlation.

Massa and Avesani (Massa & Avesani, 2004) designed an algorithm that enables the propagation of trust within a network to identify users whom an active user can trust. This algorithm leverages the explicit trust relationships from the Epinions.com platform, where users indicate the level of trust they assign to other members. These trust levels influence the perceived relevance of the provided reviews. The evaluation of this approach is based on metrics such as Mean Absolute Error (MAE), Mean Absolute User Error (MAUE), and user coverage and ratings.

In a similar vein, Jamali and Ester (Jamali & Ester, 2009) observed that ratings provided by highly trusted friends for items similar to a target item are more relevant. They developed TrustWalker, a random walk technique that combines trust-based and item-based recommendations in order to reduce the impact of noisy data while ensuring a sufficient number of ratings. Experiments conducted on the Epinions dataset demonstrated the effectiveness of TrustWalker, evaluated using RMSE and a coverage metric, and compared to methods such as TidalTrust, MoleTrust, CF Pearson, and item-based models.

Moreover, Jamali and Ester (Jamali & Ester, 2010) introduced SocialMF, a matrix factorization-based approach that incorporates trust propagation mechanisms to enhance recommendations. This method was evaluated on the Flixster and Epinions datasets using RMSE. Compared to existing models (basic matrix factorization, user-based collaborative filtering), SocialMF demonstrated a significant reduction in errors, particularly for new users.

Finally, Guo et al. (G. Guo et al., 2012) proposed Merge, an approach aimed at explicitly integrating trusted neighbors identified by users into recommendation systems to alleviate the cold-start problem. Experiments conducted on the FilmTrust, Flixster, and Epinions datasets evaluated Merge in comparison to the TrustAll method and collaborative filtering using Pearson Correlation Coefficient (PCC). The evaluation, performed using performance metrics such as Mean Absolute Error (MAE), showed that Merge improves accuracy without requiring trust propagation.

Table 12: Studies Based on Explicit Trust

Work	Method	Dataset	Evaluation Metrics	Trust Propagation
(Golbeck & Hendler, 2006)	Automatic Collaborative Filtering (ACF)	FilmTrust	Comparison of ratings using ACF and PCC (MAE)	Yes
(Massa & Avesani, 2004)	Trust Propagation	Epinions	MAE, MAUE, Ratings coverage, Users coverage	Yes
(Jamali & Ester, 2009)	Random Walk	Epinions	RMSE, Coverage	No
(Jamali & Ester, 2010)	Matrix Factorization	Epinions, Flixster	RMSE	Yes
(G. Guo et al., 2012)	User-identified Trust Neighbors	FilmTrust, Flixster, Epinions	MAE, Rating Coverage	Yes

III.5.2. Recommender Systems Based on Implicit Trust

This section presents a state-of-the-art review of research that leverages implicit trust relationships between users. The contributions are categorized based on criteria such as the method used, the dataset employed, the evaluation metrics implemented, and whether trust propagation is adopted.

Pitsilis and Marshall (Pitsilis & Marshall, 2008) consider trust as a form of opinion that is always subjective and uncertain. Each opinion is represented as a three-dimensional metric comprising belief, disbelief, and uncertainty. Uncertainty is modeled based on prediction error, while belief and disbelief levels are derived from the correlation between each pair of users. The system shows performance comparable to the beta distribution approach. However, no comparison with traditional collaborative filtering (CF) was conducted.

On the other hand, Papagelis et al. (Papagelis et al., 2005) propose an alternative approach to addressing the sparsity problem by introducing a method that defines transitive properties between users within the context of a social network. Their approach focuses on developing a computational model that explores transitive user similarities based on inferred trust, in order to tackle the sparsity issue. This approach assumes that interactions are based on rating activity and that it is possible to express similarity conditions between pairs of users, according to the subset of their co-rated items. These similarity conditions are considered as links in a social network, where node associations are inferred through the expression of established trust between users.

To address the shortcomings of the two previously presented methods, O'Donovan and Smyth (O'Donovan & Smyth, 2005) assume that profile similarity is only one of many possible factors that could be used to influence recommendation and prediction. They believe that another factor can be defined: "trust". If a profile has made many accurate recommendation predictions in the past, it can be considered more reliable than another profile that has made many poor predictions. For this reason, O'Donovan and Smyth define two trust models and show how they can be easily integrated into the mechanics of a standard collaborative filtering RS.

To enhance existing trust-based collaborative filtering (CF) techniques, Hwang and Chen (Hwang & Chen, 2007) propose a mechanism to derive the trust score directly from rating data based on the accuracy of users' past predictions. Hwang and Chen introduce two trust metrics: the global trust metric and the local trust metric. In particular, they discuss local trust metrics that can be incorporated into the collaborative filtering (CF) process and effectively propagated through the trust network to infer indirect relationships.

Lathia et al. (Lathia et al., 2008) present a variation of k-Nearest Neighbors (kNN), the k-Nearest Recommenders Trust algorithm (or kNR), by evaluating the usefulness of the rating information received. This algorithm proposes a top-k list of users who have the necessary information to make a prediction. If the distance between the user's rating and the recommender's rating increases, trust decreases linearly.

Existing work on Trust-aware Recommender Systems (TARS) suffers from the issue of labeling trust statements by users. For this reason, Yuan et al. (W. Yuan et al., 2010) propose a new model called implicit TARS, which leverages implicit trust networks. This model uses binary trust measures and similarities between users to address the issue of explicit trust, which is effort-intensive due to the need to compare the ratings users assign to common items.

The work of Shambour and Lu (Shambour & Lu, 2011, 2012, 2015) uses the Mean Squared Differeces (MSD) method to calculate the accuracy of a user's past predictions in order to measure their reliability. After calculating the direct implicit trust scores, trust propagation leverages indirect trust relationships to compute trust scores between users who are not directly connected.

Unlike most implicit trust inference methods that use ratings as the main source of trust, the model by Zahir et al. (Zahir et al., 2019) adds the direction of rating agreement between users. They propose AgreeRelTrust, in which trust is based on users' positive and negative agreements as well as their relative activity, and does not require prior predictions. Zahir et al. hypothesize that people are more likely to trust recommendations from individuals who have shared similar opinions in the past. Agreement can be defined as the ratio of the number of agreed ratings to the total number of co-rated items (considering both positive and negative agreements).

In these studies, it is evident that recommendations based on implicit trust represent a promising research direction, as they help overcome, on the one hand, the limitations related to explicit trust declarations by users, and on the other hand, significantly alleviate the cold-start problem. For these reasons, we intend to explore this type of approach, which appears particularly effective in the recommendation of POIs in the tourism sector, where tourists without prior profiles are very common.

Finally, Shambour et al. propose a hybrid recommendation approach based on trust between users and trust between items. This approach also employs trust propagation to establish new relationships between users who do not have a direct trust connection (Shambour & Lu, 2015).

The aforementioned studies highlight the importance of trust relationships and propagation in the recommendation process as shown in Table 13 below.

III.5.3. POI Recommendation Based on Explicit Trust

In the field of Point-of-Interest (POI) recommendation, leveraging trust information between users has proven to be a valuable strategy for enhancing recommendation quality. Explicit trust relationships, in which users directly express trust or distrust towards others, provide rich and interpretable data that can be integrated into recommendation models. Recent research has explored various ways to exploit this trust data, often combining it with user preferences and contextual factors such as geography. The following works demonstrate how integrating explicit trust, along with other social and behavioral signals, can lead to more accurate and socially-aware POI recommendation systems (see Table 14).

In their work, Zhu et al. (Zhu et al., 2018) introduce a hybrid recommendation framework that integrates user preferences, social trust-distrust relationships, and geographical influence to enhance Point-of-Interest (POI) recommendations. A key innovation of their approach is the utilization of a modified normalized Jaccard coefficient to calculate both trust and distrust scores, effectively capturing direct trust between users and indirect trust obtained through propagation, including distrust links. This methodology allows for a more nuanced understanding of user relationships, which is particularly beneficial in social networks where trust dynamics are complex. By incorporating these elements, the algorithm adjusts propagation parameters to refine recommendation accuracy, demonstrating improved performance over traditional trust-based approaches.

Table 13: Studies Based on Implicit Trust

Work	Method	Dataset	Evaluation Metrics	Trust Propagation
(Pitsilis & Marshall, 2008)	CF	MovieLens	Mean Error %	No
(O'Donovan & Smyth, 2005)	CF	MovieLens	Mean Error Rate	No
(Papagelis et al., 2005)	CF	Movie recommendation system (MRS)	Statistical accuracy (MAE), Decision-support accuracy (ROC)	Yes
(Hwang & Chen, 2007)	CF	MovieLens	Mean Absolute Error, Coverage	Yes
(Lathia et al., 2008)	CF	MovieLens	MAE, RMSE, Coverage	No
(Lifen, 2008)	CF	MovieLens	Divergence between predicted rating and real rate, percentage of successful estimation	Yes
(W. Yuan et al., 2010)	TARS	Epinions	MAE, Coverage	Yes
(Shambour & Lu, 2011) (TeCF)	User-based (CF)	MovieLens	MAE, Coverage	Yes
(Bedi & Sharma, 2012)	TARS	Jester, MovieLens	Precision, Recall, F-measure	No
(Shambour & Lu, 2012)	CF, User and item-based	MovieLens, Yahoo! Webscope R4	Mean Absolute Error (MAE), Coverage	Yes
(F. Zhang et al., 2014) (ARA)	CF	MovieLens	MAE, F-measure	Yes
(Shambour & Lu, 2015) (HUIT)	CF, User and item-based	MovieLens, Yahoo! Webscope R4, FilmTrust	MAE, Coverage	Yes
(Roy et al., 2015)	CF	MovieLens	MAE	Yes
(Zahir et al., 2019)	Memory-based CF	GroupLens, MovieLens, Jester	RMSE, MAE	No

Zhu et al. address the limitations of traditional recommendation systems in Location-Based Social Networks (LBSNs), which often rely solely on user similarity, popularity, or geographical influence, neglecting the role of social trust. To enhance recommendation accuracy, they introduce a novel approach that identifies "trust clusters", which are groups of users with similar trust relationships. These clusters are utilized in a trust prediction method, combining trust values and user similarities to recommend friends to target users. For Point-of-Interest (POI) recommendations, they develop a hybrid framework integrating user preferences, geographical influence, and trust relationships. Experimental evaluations on real-world datasets from Foursquare and Instagram demonstrate that this trust cluster-based recommendation approach outperforms baseline methods in terms of PRECISION and RECALL (Zhu et al., 2019).

Logesh and Subramaniyaswamy (Logesh & Subramaniyaswamy, 2017) use a POI recommendation algorithm called Social Pertinent Trust Walker (SPTW), which is based on trust levels between users calculated using matrix factorization. This algorithm is an extended version of the work by Jamali and Ester (Jamali & Ester, 2009), designed to effectively recommend locations by integrating user similarities, trust relationships, and place categories.

Table 14: Comparative Table of the Different Mentioned Approaches

Work	Approach	Methodology	Factors Considered	Objective
(Zhu et al., 2018)	Hybrid Algorithm	Use of a normalized Jaccard coefficient to measure trust/distrust and trust propagation	Direct and indirect trust, user preferences, geographical influence	Provide recommendations by adjusting trust/distrust propagation
(Zhu et al., 2019)	Trust Clusters and Trust Prediction	Integration of clusters in a trust prediction method	Similarity between individuals, trust values, geographical influence, user preferences	Recommend friends and POIs
(Logesh & Subramaniyaswamy, 2017)	Social Pertinent Trust Walker (SPTW)	Matrix factorization to calculate trust levels	Similarity between users, trust relationships, place categories	Effectively recommend locations

III.5.4. POI Recommendation Based on Implicit Trust

The recommendation of points of interest (POIs) represents a crucial component of Location-Based Social Networks (LBSNs), as it provides significant value in terms of personalization and adaptability. However, traditional collaborative filtering methods used for this type of recommendation face the cold-start problem (in the case of a new user or a new POI) and data sparsity (when a user has few interactions with POIs). Furthermore, traditional recommender systems often ignore existing social relationships between users in LBSNs, thereby missing the opportunity to provide more reliable POI recommendations. To address these issues, several methods in the literature use trust between users in LBSNs to mitigate the impact of data sparsity on the accuracy of POI recommendations.

Wang et al. emphasize the role of personality traits in determining the level of trust between users, suggesting that users with similar traits are more likely to trust each other. This trust can be used to compute similarity metrics that are enhanced by trust for more accurate POI recommendations. Additionally, they introduce a novel approach where trust links are inferred when users co-visit the same location within a specific time frame. This technique integrates both geographical and temporal factors into the recommendation process, allowing for more context-aware POI suggestions. By leveraging these dual influences, their method enhances the ability to make relevant recommendations in dynamic social settings (W. Wang et al., 2020).

Ekaterina et al. address the dynamic nature of review-based trust in POI recommendation systems, arguing that older reviews, written several years ago, are generally less informative than recent ones. As a result, the trust in review authors should vary depending on the recency of the reviews, influencing the overall recommendation process. Their work highlights the importance of considering the temporal aspect of reviews to better assess the relevance and trustworthiness of review-based information in POI recommendations (Ekaterina et al., 2020).

Xu et al. (C. Xu et al., 2021) propose an innovative Points of Interest (POI) recommendation method that leverages matrix factorization to integrate multiple factors such as user preferences, trust relationships, check-in time, and geographical location. Their approach utilizes the principle of social relationship propagation within Location-Based Social Networks (LBSNs) to infer trust between users, relying on both direct and inferred social connections. By incorporating the Jaccard Mean Squared Difference (JMSD) to calculate similarity, the method effectively measures direct trust scores and estimates indirect trust values through propagation, improving recommendation accuracy, particularly in sparse data scenarios. This hybrid approach highlights the importance of combining social and spatiotemporal factors in enhancing the quality of POI recommendations.

An et al. tackle the issues of data sparsity and trust estimation in POI recommendation systems by proposing the Bi-directional Trust enhanced Collaborative Filtering (BiTCF) model. This approach integrates a temporal similarity measure within a weighted matrix factorization framework to predict missing user preferences. In addition, BiTCF estimates both direct and indirect trust relationships between users by incorporating POI category, temporal, geographical, and textual factors. By capturing the dynamic nature of user trust and considering multiple contextual influences, BiTCF enhances the accuracy and relevance of recommendations. Their method shows significant improvements through extensive experiments on real-world datasets like Gowalla and Foursquare (An et al., 2023).

Previous studies fall within the scope of implicit trust, inferred from user interactions with LBSN-type platforms. These studies rely either on models such as matrix factorization or simply on collaborative filtering techniques using various types of similarity measures. In this thesis, a new similarity measure is proposed to estimate similarity scores between different users of an LBSN during its testing phase. This score is calculated based on POI ratings and user check-ins. Table 15 below compares some literature works with our approach.

III.6. Analysis and Discussions

The recommendation approaches studied are based on two types of trust between users: explicit trust and implicit trust. These approaches are evaluated using performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and other indicators used to assess their effectiveness. The first family of approaches relies on explicit trust, based on direct declarations between users and its diffusion within a network. In contrast, the second family favors implicit trust, inferred from past interactions between users, and relies on its propagation to enhance the relevance of recommendations.

Table 15: Comparative Grid of Related Works and Our Approach

Work	**Approach**	**Methodology**	**Factors Considered**	**Objective**
(W. Wang et al., 2020)	Trust Based on Co-visits	Deduction of trust links between users based on joint visits to a place	Co-visits to a POI, geographical and temporal influence	Generate POI recommendations based on real interactions between users
(Ekaterina et al., 2020)	Temporal Trust	Weighting reviews based on their age	Date of review writing	Adjust trust in reviews to improve POI recommendations
(C. Xu et al., 2021)	Matrix Factorization	Calculating similarity with Jaccard Mean Squared Difference (JMSD)	Preferences, social relationships, spatiotemporal factors	Estimate direct and indirect trust values through propagation
(An et al., 2023)	Matrix Factorization with Temporal Similarity	Using a temporal similarity measure and contextual factors to fill in missing preferences	Temporal, geographical, textual factors, POI categories	Mitigate the data sparsity problem and improve user preference prediction
Our work	CF Memory-based using Trust	Deduction of trust based on ratings and check-ins	POI ratings, user check-ins	Generate POI recommendations based on historical user behavior

This chapter particularly highlights the importance of implicit trust relationships and the impact of their propagation compared to approaches based on explicit trust. It also examines the growing integration of Artificial Intelligence (AI) in the field of smart tourism, with a focus on the use of Implicit Trust-Based Recommender Systems (ITBRS). These systems leverage implicit trust relationships between tourists, inferred from their interactions such as ratings, check-ins, and clicks. The analysis emphasizes the advantages of implicit trust, notably its ability to reduce the need for active user participation and to mitigate the cold-start problem related to data sparsity.

Finally, to the best of our knowledge, no previous work has implicitly utilized user check-ins to calculate trust within the context of LBSNs, nor has it proposed an approach that conducts multi-level offline evaluations of POI recommendations using metrics such as RMSE and PRECISION/RECALL, computed over the progressive evolution of an LBSN's historical data. In this context, our research focuses on a k-nearest neighbors-based heuristic aimed at constructing trust relationships inferred from both POI ratings and user check-ins within a LBSN.

Conclusion

This chapter provided an overview of the various POI recommendation techniques in LBSNs, with a particular focus on the integration of social trust. From collaborative methods to hybrid approaches and graph-based models, each method offers specific advantages for enhancing the relevance of recommendations. However, challenges remain, particularly regarding data quality, computational complexity, and privacy concerns. In this context, the next chapter will introduce our first thesis contribution, detailing a POI recommendation system model that leverages implicit trust, inferred from user check-ins and ratings.

Part II

Contributions And Experiments

Chapter IV:

The HRCT model integrating similarities for predictions

IV.1. Introduction

With the rapid evolution of information and communication technologies, the rise of location-based social networks (LBSNs) has transformed the way users interact with their environment. These platforms, such as Yelp, Foursquare, Gowalla, and Brightkite, allow users to share their experiences by posting reviews, ratings, and check-ins at various points of interest (POIs). However, the diversity and rapid growth of POIs, along with the exponential increase in user interactions, are making traditional recommendation methods less and less effective, as they suffer from issues such as data sparsity and the cold start problem.

In this chapter, we focus on recommendation systems (RSs) based on implicit trust, a promising alternative that leverages trust relationships inferred from user interactions with POIs (check-ins, ratings), without requiring active participation to explicitly express their level of trust towards other users.

IV.2. Problem Definition

Traditional recommendation systems face several major challenges within the context of LBSNs:

1. **Data Sparsity**: POI rating matrices are often sparse, which affects the quality of recommendations.
2. **Cold Start**: New users do not have enough history to benefit from accurate recommendations.
3. **Preference Dynamics**: User interests change rapidly, making static predictive models less relevant.
4. **Limitations of Explicit Trust-Based Methods**: Few users provide explicit trust scores, making these models difficult to leverage effectively

To address these limitations, we propose a POI recommendation system based on implicit trust (Medjroud et al., 2025b). This system infers trust between users from their interactions with POIs by using three types of matrices:

- **TDMR (Trust Derivation Matrix based on Rating)**: Based on users' ratings of POIs.
- **TDMC (Trust Derivation Matrix based on Check-in)**: Based on the frequency and diversity of check-ins.
- **H-Trust Matrix**: A hybrid matrix that combines TDMR and TDMC to provide a denser and more accurate representation of implicit trust between users, leveraging both rating and check-in behaviors for improved recommendation performance.

Our main objective is to evaluate the effectiveness of this RS in terms of recommendation accuracy and its ability to reduce data sparsity, compared to traditional collaborative filtering techniques based on Pearson, Cosine, and Jaccard similarity measures.

IV.3. Problem Formulation

This section describes the process of inferring implicit trust between users based on their ratings and check-ins of POIs. To do so, we first present the formulas used to calculate this trust. Then, we illustrate, through an example, the construction of the user-user trust matrix from the User-POI-rating matrix. Finally, we explain, with another example, how to construct the user-user trust matrix from the User-POI-check-in matrix.

IV.3.1. Implicit Trust Calculation

O'Donovan and Smith (O'Donovan & Smyth, 2005) define trust based on the reliability of a partner's profile in providing accurate recommendations in the past. For example, a profile that has made numerous accurate recommendation predictions can be considered more trustworthy than one that has made mostly poor predictions. This type of prediction can be calculated using Equation (24) (Resnick et al., 1994):

Equation 24: Rating Prediction based on Resnick Formula Adapted for LBSN

$$P_{a,i} = \overline{r_a} + \frac{\sum_{b=1}^{N}(r_{b,i} - \overline{r_b}) * sim(a,b)}{\sum_{b=1}^{N}|sim(a,b)|}$$

Where:

- $P_{a,i}$: The predicted rating for user *a* on POI *i*.
- $\overline{r_a}$: The average ratings of user *a* across all POIs.
- $r_{b,i}$: The actual rating that user *b* assigned to POI *i*.
- $sim(a,b)$: The similarity measure between user *a* and user *b*.
- N : The set of neighbors of user *a*.

However, to calculate the rating prediction of user *a* for a given POI *i* based on user *b* considered as the only recommender (O'Donovan & Smyth, 2005); Equation (25), derived from Equation (24) can be used (Hwang & Chen, 2007; Shambour & Lu, 2011, 2012, 2015):

Equation 25: Rating Prediction Based on a Single Recommender's Influence Adapted for LBSN

$$P_{a,i}^{b} = \overline{r_a} + (r_{b,i} - \overline{r_b})$$

Where:

- $P_{a,i}^{b}$: The predicted rating for user *a* on POI *i* based on user *b*.
- $\overline{r_a}$: The average ratings of user *a* across all POIs.

- $r_{b,i}$: The actual rating given to POI *i* by user *b*.
- $\overline{r_b}$: The average ratings of user *b* across all POIs.

According to O'Donovan and Smith, the prediction of a rating for a user a on a POI *i* based on recommender *b* is considered "*correct*" 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 (26):

Equation 26: Rating based Correct Function Adapted for LBSN

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

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

Next, O'Donovan and Smith use Equation (27) below to define $RecSet(b)$ as the complete set of recommendations in which recommender *b* was involved:

Equation 27: The Set of Recommendations Adapted for LBSN

$$RecSet(b) = \{(P^b_{1,1}, r_{1,1}), \dots, (P^b_{m,n}, r_{m,n})\}$$

Where:

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

- $r_{j,k}$ represents the real rating of POI k (where k varies from 1 to *n*) given by a user *j* (where *j* varies from 1 to *m*).

From $RecSet(b)$, the subset of correct recommendations, denoted as $CorrectSet(b)$, is calculated using Equation (28) as shown below (O'Donovan & Smyth, 2005).

Equation 28: The set of correct recommendations Adapted for LBSN

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

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 Equation (29) as shown below (O'Donovan & Smyth, 2005).

Equation 29: Profile-Level Trust Adapted for LBSN

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

From Equation (29), a more refined trust metric at the item-level (POI-level), denoted $Trust^{I}$, can be defined to measure the percentage of correct recommendations for POI i obtained by a recommender b out of all its recommendations, as indicated in formula (30) (O'Donovan & Smyth, 2005).

Equation 30: Item-Level Trust Adapted for LBSN

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

Equation (29) 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 POIs (Hwang & Chen, 2007), (Shambour & Lu, 2011). On the other hand, Equation (30) highlights the reputation of a given user among all users based on its common ratings for a specific POI.

In this same context, drawing inspiration from the work of (Zahir et al., 2019), the trust of a given user a in another user b (recommender) based on their common ratings for all POIs can be defined using Equation (31) (Medjroud et al., 2022):

Equation 31: User-User Trust Based on Rating

$$Trust^{U}(a \to b) = \frac{card\left\{\left(P_{j,k}^{b}, r_{j,k}\right) \in CorrectSet(b) : j=a\right\}}{card\left\{\left(P_{j,k}^{b}, r_{j,k}\right) \in RecSet(b) : j=a\right\}}$$

Where $Trust^{U}(a \to b)$ represents the trust user a has in user b (recommender), determined as the percentage of correct recommendations in which recommender b participated with user a based on their common ratings of all POIs.

From Equation (31), the trust of user a in recommender b for a particular POI 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 POI i, as indicated in Equation (32) below:

Equation 32: Item-Specific Trust Estimation Between Users

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

In the following, we utilized Equation (31) to infer the implicit trust between users from their POI ratings.

This type of trust will be used to compute the rating prediction using Equation (33).

Equation 33: Rating Prediction Based on Rating-Trust Relationships Between Users

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N}\left(r_{b,x} - \overline{r_b}\right) * Trust^{U}\,(a \to b)}{\sum_{b=1}^{N} Trust^{U}\,(a \to b)}$$

$Trust^U(a \rightarrow b)$: user-user trust derived from ratings.

In this chapter, we used Equation (25) 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 Equation (34) below.

Equation 34: Check-in Prediction Based on the Influence of a Single Recommender, Adapted for LBSNs

$$C^b_{a,i} = \bar{c_a} + \left(c_{b,i} - \bar{c_b}\right)$$

Where:

- $C^b_{a,i}$: The predicted check-in for item *i* by user *a*, based on user *b*.
- $\bar{c_a}$: The average number of check-ins of user *a*.
- $c_{b,i} \in \{0,1\}$: Indicates whether user *b* has checked at POI *i*.
- $\bar{c_b}$: The average number of check-ins of user *b*.

Equation (26) above can be applied in the case of check-ins to obtain Equation (35) below:

Equation 35: Check-in-Based Correction Function Adapted for LBSNs

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

Replacing the ratings with the check-ins in Equations (27) and (28) above, $RecSet\_C(b)$, which represents the complete set of recommendations, is given by Equation (36) below, and $CorrectSet\_C(b)$, indicating the subset of correct recommendations, is given by Equation (37) below.

Equation 36: The set of recommendations in case of check-ins

$$RecSet\_C(b) = \left\{\left(C^b_{1,1}, c_{1,1}\right), \ldots, \left(C^b_{m,n}, c_{m,n}\right)\right\}$$

Where:

- $C^b_{j,k}$ represents the prediction made by recommender *b* for the check-in that a user *j* (where *j* ranges from 1 to *m*) will make for an item *k* (where *k* ranges from 1 to *n*).

- $c_{j,k}$ represents the actual check-in of item *k* (where *k* varies from 1 to *n*) made by a user *j* (where *j* varies from 1 to *m*).

From $RecSet\_C(b)$, the subset of correct recommendations denoted $CorrectSet\_C(b)$ is calculated using Equation (37).

Equation 37: The set of correct recommendations in case of check-ins

$$CorrectSet\_C(b) = \left\{\left(C^b_{j,k}, c_{j,k}\right) \epsilon\ RecSet\_C(b) : Correct\_C(k, b, C^b_{j,k})\right\}$$

Then, using the check-ins, the derivation of user *a*'s trust towards user *b* can be inferred from Equation (31) above and is applied by replacing the ratings with the check-ins to obtain Equation (38) below:

Equation 38: User-User Trust Based on Check-in

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

Finally, note that Equation (39) below, deduced from Equation (33) above, can be utilized to calculate the rating predictions of POIs from user check-in data.

Equation 39: Rating Prediction based on Check-in-Trust between users

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N}(r_{b,x} - \overline{r_b}) * Trust\_C^U\ (a \to b\ )}{\sum_{b=1}^{N} Trust\_C^U\ (a \to b\ )}$$

$Trust\_C^U(a \to b)$: user-user trust derived from check-ins.

From Equations (33) and (39), Equation (40) can also be expressed in an alternative form:

Equation 40: Rating Prediction based on Trust and Similarity between users

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N}\left(r_{b,x} - \overline{r_b}\right) * w(a,b\ )}{\sum_{b=1}^{N} w(a,b\ )}$$

$$w(a,b\ ) = \begin{cases} Trust^U\ (a \to b\ ) : trust\ based\ on\ rating \\ Trust\_C^U\ (a \to b\ ) : trust\ based\ on\ check-in \end{cases}$$

When combining the HCRT model with other similarity models, we use the following harmonic mean of trust and similarity Equation (41) to integrate both aspects into the rating prediction process:

Equation 41: The Harmonic Mean of Trust and Similarity in Our Context

$$w(a,b) = \frac{2(sim(a,b))(Trust^U(a,b))}{sim(a,b) + Trust^U(a,b)}$$

Where :

- $sim(a,b)$: similarity which can be Pearson, Cosine, or Jaccard between user *a* and user *b*.
- $Trust^U(a,b)$: user-user trust deduced from rating, check-in, or hybrid; user *a* towards user *b*.

IV.3.2. Example of Trust Matrix Calculation Based on Ratings

To clearly explain how to derive trust between users; noted as TDMR (Trust Derivation Matrix based on Ratings); and how to compute the RPMR (Rating Prediction Matrix based on Ratings), based on their ratings of POIs, we found it necessary to provide more detail on the calculation steps through an example using the UPRM (User-POI Ratings Matrix) as shown in Table 16 below.

Notably, the same steps can be applied to compute the TDMC (Trust Derivation Matrix based on Check-ins), using the UPCM (User-POI Check-ins Matrix) see Table 22, which contains the POI check-ins made by users; and the RPMC (Rating Prediction Matrix based on Check-ins), using the UPRM (User-POI Ratings Matrix).

Table 16: Example of UPRM Matrix

UPRM	POI1	POI2	POI3	POI4	POI5	POI6	$\overline{rate_u}$
U1		5	2		3		10/3
U2	4			3		4	11/3
U3		1	2			2	5/3
U4	5			3	2		10/3
U5		5	5			3	13/3

Using the previously mentioned Equation (25), we calculate the predicted rating of POI i for the active user a inferred from their neighbor b.

PredictedRating(a, b, i) = meanRate(a) + Rate(b, i) – meanRate(b)

For example, PredictedRating(U1, U3, POI2) = meanRate(U1) + Rate(U3, POI2) – meanRate(U3) = 8/3

After applying Equation (25) to the entire UPRM matrix (see Table 16), we obtain matrix M2, which contains the predicted ratings, as shown in Table 17 below:

Table 17: Matrix M2, which contains the predicted ratings

M2	POI1		POI2		POI3		POI4		POI5		POI6	
U1	U1	?	U1	5	U1	2	U1	?	U1	3	U1	?
	U2	?	U2	?	U2	?	U2	?	U2	?	U2	?
	U3	?	U3	8/3	U3	11/3	U3	?	U3	?	U3	?
	U4	?	U4	?	U4	?	U4	?	U4	2	U4	?
	U5	?	U5	4	U5	4	U5	?	U5	?	U5	?
U2	U1	?	U1	?	U1	?	U1	?	U1	?	U1	?
	U2	4	U2	?	U2	?	U2	3	U2	?	U2	4
	U3	?	U3	?	U3	?	U3	?	U3	?	U3	4
	U4	16/3	U4	?	U4	?	U4	10/3	U4	?	U4	?
	U5	?	U5	?	U5	?	U5	?	U5	?	U5	7/3
U3	U1	?	U1	10/3	U1	1/3	U1	?	U1	?	U1	?
	U2	?	U2	?	U2	?	U2	?	U2	?	U2	2
	U3	?	U3	1	U3	2	U3	?	U3	?	U3	2
	U4	?	U4	?	U4	?	U4	?	U4	?	U4	?
	U5	?	U5	7/3	U5	7/3	U5	?	U5	?	U5	1/3
U4	U1	?	U1	?	U1	?	U1	?	U1	3	U1	?
	U2	11/3	U2	?	U2	?	U2	8/3	U2	?	U2	?
	U3	?	U3	?	U3	?	U3	?	U3	?	U3	?
	U4	5	U4	?	U4	?	U4	3	U4	2	U4	?
	U5	?	U5	?	U5	?	U5	?	U5	?	U5	?
U5	U1	?	U1	18/3	U1	3	U1	?	U1	?	U1	?
	U2	?	U2	?	U2	?	U2	?	U2	?	U2	14/3
	U3	?	U3	11/3	U3	14/3	U3	?	U3	?	U3	14/3
	U4	?	U4	?	U4	?	U4	?	U4	?	U4	?
	U5	?	U5	5	U5	5	U5	?	U5	?	U5	3

Using the previously mentioned Equation (26), we can calculate the difference between the real rating provided by the active user *a* and the predicted rating from the neighboring user *b* for this same active user *a*.

For example, to calculate the difference between the actual rating assigned by the active user U1 for POI2 and the predicted rating from the neighboring user U3 for the same POI2, we perform the following calculation:

| PredictedRating(U1, U3, POI2) – ActualRating(U1, POI2) | = | 5 – 8/3 | = 7/3 < ε

After applying Equation (26) to the entire M2 matrix (see Table 17), we obtain the matrix M3, as shown in Table 18 below, which contains the prediction accuracy deviations.

Table 18: Matrix M3, which contains the differences between actual and predicted ratings

<table>
<tr><th>M3</th><th colspan="2">POI1</th><th colspan="2">POI2</th><th colspan="2">POI3</th><th colspan="2">POI4</th><th colspan="2">POI5</th><th colspan="2">POI6</th></tr>
<tr><td rowspan="5">U1</td><td>U1</td><td>?</td><td>U1</td><td>|5-5|</td><td>U1</td><td>|2-2|</td><td>U1</td><td>?</td><td>U1</td><td>|3-3|</td><td>U1</td><td>?</td></tr>
<tr><td>U2</td><td>?</td><td>U2</td><td>|5-?|</td><td>U2</td><td>|2-?|</td><td>U2</td><td>?</td><td>U2</td><td>|3-?|</td><td>U2</td><td>?</td></tr>
<tr><td>U3</td><td>?</td><td>U3</td><td>|5-8/3|</td><td>U3</td><td>|2-11/3|</td><td>U3</td><td>?</td><td>U3</td><td>|3-?|</td><td>U3</td><td>?</td></tr>
<tr><td>U4</td><td>?</td><td>U4</td><td>|5-?|</td><td>U4</td><td>|2-?|</td><td>U4</td><td>?</td><td>U4</td><td>|3-2|</td><td>U4</td><td>?</td></tr>
<tr><td>U5</td><td>?</td><td>U5</td><td>|5-4|</td><td>U5</td><td>|2-4|</td><td>U5</td><td>?</td><td>U5</td><td>|3-?|</td><td>U5</td><td>?</td></tr>
<tr><td rowspan="5">U2</td><td>U1</td><td>|4-?|</td><td>U1</td><td>?</td><td>U1</td><td>?</td><td>U1</td><td>|3-?|</td><td>U1</td><td>?</td><td>U1</td><td>|4-?|</td></tr>
<tr><td>U2</td><td>|4-4|</td><td>U2</td><td>?</td><td>U2</td><td>?</td><td>U2</td><td>|3-3|</td><td>U2</td><td>?</td><td>U2</td><td>|4-4|</td></tr>
<tr><td>U3</td><td>|4-?|</td><td>U3</td><td>?</td><td>U3</td><td>?</td><td>U3</td><td>|3-?|</td><td>U3</td><td>?</td><td>U3</td><td>|4-4|</td></tr>
<tr><td>U4</td><td>|4-16/3|</td><td>U4</td><td>?</td><td>U4</td><td>?</td><td>U4</td><td>|3-10/3|</td><td>U4</td><td>?</td><td>U4</td><td>|4-?|</td></tr>
<tr><td>U5</td><td>|4-?|</td><td>U5</td><td>?</td><td>U5</td><td>?</td><td>U5</td><td>|3-?|</td><td>U5</td><td>?</td><td>U5</td><td>|4-7/3|</td></tr>
<tr><td rowspan="5">U3</td><td>U1</td><td>?</td><td>U1</td><td>|1-10/3|</td><td>U1</td><td>|2-1/3|</td><td>U1</td><td>?</td><td>U1</td><td>?</td><td>U1</td><td>|2-?|</td></tr>
<tr><td>U2</td><td>?</td><td>U2</td><td>|1-?|</td><td>U2</td><td>|2-?|</td><td>U2</td><td>?</td><td>U2</td><td>?</td><td>U2</td><td>|2-6/3|</td></tr>
<tr><td>U3</td><td>?</td><td>U3</td><td>|1-1|</td><td>U3</td><td>|2-2|</td><td>U3</td><td>?</td><td>U3</td><td>?</td><td>U3</td><td>|2-2|</td></tr>
<tr><td>U4</td><td>?</td><td>U4</td><td>|1-?|</td><td>U4</td><td>|2-?|</td><td>U4</td><td>?</td><td>U4</td><td>?</td><td>U4</td><td>|2-?|</td></tr>
<tr><td>U5</td><td>?</td><td>U5</td><td>|1-7/3|</td><td>U5</td><td>|2-7/3|</td><td>U5</td><td>?</td><td>U5</td><td>?</td><td>U5</td><td>|2-1/3|</td></tr>
<tr><td rowspan="5">U4</td><td>U1</td><td>|5-?|</td><td>U1</td><td>?</td><td>U1</td><td>?</td><td>U1</td><td>|3-?|</td><td>U1</td><td>|2-3|</td><td>U1</td><td>?</td></tr>
<tr><td>U2</td><td>|5-11/3|</td><td>U2</td><td>?</td><td>U2</td><td>?</td><td>U2</td><td>|3-8/3|</td><td>U2</td><td>|2-?|</td><td>U2</td><td>?</td></tr>
<tr><td>U3</td><td>|5-?|</td><td>U3</td><td>?</td><td>U3</td><td>?</td><td>U3</td><td>|3-?|</td><td>U3</td><td>|2-?|</td><td>U3</td><td>?</td></tr>
<tr><td>U4</td><td>|5-5|</td><td>U4</td><td>?</td><td>U4</td><td>?</td><td>U4</td><td>|3-3|</td><td>U4</td><td>|2-2|</td><td>U4</td><td>?</td></tr>
<tr><td>U5</td><td>|5-?|</td><td>U5</td><td>?</td><td>U5</td><td>?</td><td>U5</td><td>|3-?|</td><td>U5</td><td>|2-?|</td><td>U5</td><td>?</td></tr>
<tr><td rowspan="5">U5</td><td>U1</td><td>?</td><td>U1</td><td>|5-18/3|</td><td>U1</td><td>|5-3|</td><td>U1</td><td>?</td><td>U1</td><td>?</td><td>U1</td><td>|3-?|</td></tr>
<tr><td>U2</td><td>?</td><td>U2</td><td>|5-?|</td><td>U2</td><td>|5-?|</td><td>U2</td><td>?</td><td>U2</td><td>?</td><td>U2</td><td>|3-14/3|</td></tr>
<tr><td>U3</td><td>?</td><td>U3</td><td>|5-11/3|</td><td>U3</td><td>|5-14/3|</td><td>U3</td><td>?</td><td>U3</td><td>?</td><td>U3</td><td>|3-14/3|</td></tr>
<tr><td>U4</td><td>?</td><td>U4</td><td>|5-?|</td><td>U4</td><td>|5-?|</td><td>U4</td><td>?</td><td>U4</td><td>?</td><td>U4</td><td>|3-?|</td></tr>
<tr><td>U5</td><td>?</td><td>U5</td><td>|5-5|</td><td>U5</td><td>|5-5|</td><td>U5</td><td>?</td><td>U5</td><td>?</td><td>U5</td><td>|3-3|</td></tr>
</table>

Based on the matrix M3, we calculate a binary success or failure score, which estimates the closeness between the actual rating given by the current user (denoted as a) and the rating generated by a neighboring user (denoted as b) for a given POI (denoted as i), as indicated below.

$$Correct(i, b, a) \Leftrightarrow |PredictedRating(a, b, i) - ActualRating(a, i)| < \varepsilon$$

If $\left|P_{a,i}^{b} - r_{a,i}\right| < \varepsilon$ then $Correct(i, b, a) = 1$ Otherwise $Correct(i, b, a) = 0$.

If this score is less than ε (a precision parameter), the function $Correct(i, b, a)$ equals 1 (good precision). Otherwise, if this score is greater than or equal to ε, the function is equal to 0. In this way, we obtain matrix M4 as shown in Table 19 below, which contains the success and failure scores of the predictions.

Table 19: Matrix M4, which contains the success and failure scores of the predictions

M4	POI1		POI2		POI3		POI4		POI5		POI6	
	U1	?	U1	1	U1	1	U1	?	U1	1	U1	?
	U2	?	U2	?	U2	?	U2	?	U2	?	U2	?
U1	U3	?	U3	0	U3	0	U3	?	U3	?	U3	?
	U4	?	U4	?	U4	?	U4	?	U4	1	U4	?
	U5	?	U5	1	U5	0	U5	?	U5	?	U5	?
	U1	?	U1	?	U1	?	U1	?	U1	?	U1	?
	U2	1	U2	?	U2	?	U2	1	U2	?	U2	1
U2	U3	?	U3	?	U3	?	U3	?	U3	?	U3	1
	U4	0	U4	?	U4	?	U4	1	U4	?	U4	?
	U5	?	U5	?	U5	?	U5	?	U5	?	U5	0
	U1	?	U1	0	U1	0	U1	?	U1	?	U1	?
	U2	?	U2	?	U2	?	U2	?	U2	?	U2	1
U3	U3	?	U3	1	U3	1	U3	?	U3	?	U3	1
	U4	?	U4	?	U4	?	U4	?	U4	?	U4	?
	U5	?	U5	0	U5	1	U5	?	U5	?	U5	0
	U1	?	U1	?	U1	?	U1	?	U1	1	U1	?
	U2	0	U2	?	U2	?	U2	1	U2	?	U2	?
$u4$	U3	?	U3	?	U3	?	U3	?	U3	?	U3	?
	U4	1	U4	?	U4	?	U4	1	U4	1	U4	?
	U5	0	U5	?	U5	?	U5	?	U5	?	U5	?
	U1	?	U1	1	U1	0	U1	?	U1	?	U1	?
	U2	?	U2	?	U2	?	U2	?	U2	?	U2	0
$u5$	U3	?	U3	0	U3	1	U3	?	U3	?	U3	0
	U4	?	U4	?	U4	?	U4	?	U4	?	U4	?
	U5	?	U5	1	U5	1	U5	?	U5	?	U5	1

To be able to calculate the trust between users based on Equation (31), we use the previously mentioned Equations (27) and (28). These two equations allow us to identify the subsets $RecSet(user)$ and $CorrectSet(user)$.

For example, using Table 19 above: if user U5 was involved in a prediction concerning user U1, the row corresponding to U1 will be selected to calculate how many times U5 made a correct prediction for U1, out of the total number of times U5 was involved with this user (U1).

Using Equation (27), we can calculate $RecSet(u5)$ as follows:

$$RecSet(u5) = \left\{ \begin{array}{c} (P^{u5}_{u1,poi2}, r_{u1,poi2}), (P^{u5}_{u1,poi3}, r_{u1,poi3}), (P^{u5}_{u2,poi6}, r_{u2,poi6}), \\ (P^{u5}_{u3,poi2}, r_{u3,poi2}), (P^{u5}_{u3,poi3}, r_{u3,poi3}), (P^{u5}_{u3,poi6}, r_{u3,poi6}), \\ (P^{u5}_{u4,poi1}, r_{u4,poi1}), (P^{u5}_{u5,poi2}, r_{u5,poi2}), \\ (P^{u5}_{u5,poi3}, r_{u5,poi3}), (P^{u5}_{u5,poi6}, r_{u5,poi6}) \end{array} \right\}$$

Using Equation (28), we can calculate $CorrectSet(u5)$ as follows:

$$CorrectSet(u5) = \left\{ \begin{matrix} \left(P_{u1,poi2}^{u5}, r_{u1,poi2}\right), \left(P_{u3,poi3}^{u5}, r_{u3,poi3}\right), \\ \left(P_{u5,poi2}^{u5}, r_{u5,poi2}\right), \left(P_{u5,poi3}^{u5}, r_{u5,poi3}\right), \\ \left(P_{u5,poi6}^{u5}, r_{u5,poi6}\right) \end{matrix} \right\}$$

Using Equation (31), we can calculate $Trust^U(u1 \rightarrow u5)$ as follows:

- $card\left\{\left(P_{j,k}^{u5}, r_{j,k}\right) \in CorrectSet(u5) : j = u1\right\} = card\left\{\left(P_{u1,poi2}^{u5}, r_{u1,poi2}\right)\right\} = 1$
- $card\left\{\left(P_{j,k}^{u5}, r_{j,k}\right) \in RecSet(u5) : j = u1\right\} = card\left\{\left(P_{u1,poi2}^{u5}, r_{u1,poi2}\right), \left(P_{u1,poi3}^{u5}, r_{u1,poi3}\right)\right\} = 2$
- $Trust^U(u1 \rightarrow u5) = \dfrac{card\left\{\left(P_{j,k}^{u5}, r_{j,k}\right) \in CorrectSet(u5) : j = u1\right\}}{card\left\{\left(P_{j,k}^{u5}, r_{j,k}\right) \in RecSet(u5) : j = u1\right\}} = \dfrac{1}{2} = 0.5$

In the same way, we calculate all the values of the user-user trust matrix, and from this, we can derive the TDMR (Trust Derivation Matrix based on Ratings) shown below (see Table 20).

Table 20: The TDMR Matrix Representing Trust Between Users

TDMR	$u1$	$u2$	$u3$	$u4$	$u5$
$u1$	**1**		0	1	0,50
$u2$		**1**	1	0,50	0
$u3$	0	1	**1**		0,33
$u4$	1	0,50		**1**	
$u5$	0,50	0	0,33		**1**

- The value 1 represents full trust.
- The value 0 represents that there is no trust between these two users.
- An empty grey cell indicates that there are no common ratings between the two users. (Here, it is clear that our solution — in this work — does not actually attempt to solve the sparsity problem of the rating matrix.)

After deriving the trust value, we calculate the rating prediction using Equation (33) as follows:

$$P_{u1,poi1} = \overline{r_{u1}} + \frac{\left(r_{u2,poi1} - \overline{r_{u2}}\right) * Trust^U(u1 \rightarrow u2) + \left(r_{u4,poi1} - \overline{r_{u4}}\right) * Trust^U(u1 \rightarrow u4)}{Trust^U(u1 \rightarrow u2) + Trust^U(u1 \rightarrow u4)}$$

$$P_{u1,poi1} = \frac{10}{3} + \frac{(5 - 10/3) * 1}{1}$$

$$P_{u1,poi1} = 5$$

In the same way, we compute all the values of the rating prediction matrix, and we can then derive the RPMR (Rating Prediction Matrix based on Ratings) shown below (see Table 21).

Table 21: The RPMR Matrix Representing the Predicted Ratings of Unvisited POIs

RPMR	POI1	POI2	POI3	POI4	POI5	POI6
$u1$	**5**	5	2	**3**	3	**2**
$u2$	4	**1.33**	**4**	3	**2.33**	4
$u3$	**2**	1	2	**1**	**1.66**	2
$u4$	5	**5**	**2**	3	2	**3.66**
$u5$	**4.33**	5	5	**4.33**	**4**	3

IV.3.3. Example of Calculating the Trust Matrix from Check-ins

To clearly explain how to derive the user-user trust matrix, denoted TDMC (Trust Derivation Matrix based on Check-ins), based on users' check-in records at POIs.

We found it necessary to provide more details on the calculation steps through an example using the UPCM (User-POI Check-ins Matrix) which contains the check-in data of users at various POIs shown below (see Table 22).

Table 22: Example of UPCM Matrix

UPCM	POI1	POI2	POI3	$\overline{Check_u}$
U1		1	0	1/2
U2	1	1		2/2
U3	1		0	1/2
U4	0		1	1/2

- 0: the user visited the POI, but did not like it.
- 1: the user visited the POI and liked it.
- empty cell: The user did not visit the POI.

Step1:

Using Equation (34) mentioned earlier, we calculate the predicted check-in at POI *i* for the active user *a*, derived through their recommender *b*.

PredictedCheck-in(*a*,*b*,*i*) = meanCheck(*a*) + Check(*b*,*i*) – meanCheck(*b*)

For example,

PredictedCheck-in(U1, U3, POI3) = meanCheck(U1) + Check(U3, POI3) – meanCheck(U3)

= ½ + 0 – ½ = 0

After applying Equation (34) to the entire UPCM matrix, we obtain matrix Ma, which contains the predicted check-ins as shown below (see Table 23):

Table 23: Matrix Ma containing the predicted check-ins

Ma	POI1		POI2		POI3	
U1	U1	?	U1	1	U1	0
	U2	?	U2	½	U2	?
	U3	?	U3	?	U3	0
	U4	?	U4	?	U4	1
U2	U1	?	U1	1	U1	?
	U2	1	U2	1	U2	?
	U3	3/2	U3	?	U3	?
	U4	1/2	U4	?	U4	?
U3	U1	?	U1	?	U1	0
	U2	1/2	U2	?	U2	?
	U3	1	U3	?	U3	0
	U4	0	U4	?	U4	1
U4	U1	?	U1	?	U1	0
	U2	1/2	U2	?	U2	?
	U3	1	U3	?	U3	0
	U4	0	U4	?	U4	1

Step2:

As we observe that the values are real numbers, we need to convert them into binary values in order to proceed to the next step (error distance calculation using Equation (35)); we divide our real-valued scale into two parts as follows, (see Table 24):

IF $-1 < \text{PredictedCheck-in}(U_x, U_y, POI_j) < = ½$ **THEN**
$\text{PredictedCheck-in}(U_x, U_y, POI_j) = 0$

ELSEIF $½ < \text{PredictedCheck-in}(U_x, U_y, POI_j) < 2$ **THEN**
$\text{PredictedCheck-in}(U_x, U_y, POI_j) = 1$

Table 24: The Ma matrix containing the predicted check-ins

Ma	POI1		POI2		POI3	
U1	U1	?	U1	1 = 1	U1	0 = 0
	U2	?	U2	½ = 0	U2	?
	U3	?	U3	?	U3	0 = 0
	U4	?	U4	?	U4	1 = 1
U2	U1	?	U1	1 = 1	U1	?
	U2	1 = 1	U2	1 = 1	U2	?
	U3	3/2 = 1	U3	?	U3	?
	U4	½ = 0	U4	?	U4	?
U3	U1	?	U1	?	U1	0 = 0
	U2	½ = 0	U2	?	U2	?
	U3	1 = 1	U3	?	U3	0 = 0
	U4	0 = 0	U4	?	U4	1 = 1
U4	U1	?	U1	?	U1	0 = 0
	U2	½ = 0	U2	?	U2	?
	U3	1 = 1	U3	?	U3	0 = 0
	U4	0 = 0	U4	?	U4	1 = 0

Step3:

Using the previously mentioned Equation (35), we can calculate the difference between the actual check-in made by the active user *a* and the predicted check-in made by the recommender user *b* for this same active user *a*.

For example, to calculate the difference between the actual check-in made by the active user U1 at POI3 and the predicted check-in made by the recommender user U3 at the same POI3, we perform the following calculation:

| PredictedCheck-in(U1, U3, POI3) - ActualCheck-in(U1, POI3) | = | 0 - 0 | = 0

After applying Equation (35) to the entire Ma matrix, we obtain the Mb matrix below, which contains the precision deviations in the predictions (see Table 25):

Table 25: The Mb matrix containing the differences between actual and predicted check-ins

Mb	POI1		POI2		POI3	
U1	U1	?	U1	| 1 - 1 |	U1	| 0 - 0 |
	U2	?	U2	| 0 - 1 |	U2	?
	U3	?	U3	?	U3	| 0 – 0 |
	U4	?	U4	?	U4	| 1 – 0 |
U2	U1	?	U1	| 1 - 1 |	U1	?
	U2	| 1 - 1 |	U2	| 1 - 1 |	U2	?
	U3	| 1 - 1 |	U3	?	U3	?
	U4	| 0 - 1 |	U4	?	U4	?
U3	U1	?	U1	?	U1	| 0 - 0 |
	U2	| 0 - 1 |	U2	?	U2	?
	U3	| 1 - 1 |	U3	?	U3	| 0 - 0 |
	U4	| 0 - 1 |	U4	?	U4	| 1 - 0 |
U4	U1	?	U1	?	U1	| 0 – 1 |
	U2	| 0 – 0 |	U2	?	U2	?
	U3	| 1 – 0 |	U3	?	U3	| 0 - 1 |
	U4	| 0 – 0 |	U4	?	U4	| 1 – 1 |

Step4:

Based on the Mb matrix, we calculate a binary success or failure score that estimates the closeness between the actual check-in made by the current user (denoted as *a*) and the check-in predicted by a recommender user (denoted as *b*) for a given POI (denoted as *i*), as indicated below:

$Correct\_C(i, b, a) \Leftrightarrow$ | PredictedCheck-in(a, b, i) - ActualCheck-in(a, i) | = 0

IF $\left|C_{a,i}^{b} - c_{a,i}\right| = 0$
THEN $Correct\_C(i, b, a) = 1$
ELSE $Correct\_C(i, b, a) = 0.$

Example:

If |PredictedCheck-in(U1, U3, POI3) - ActualCheck-in(U1, POI3)| = | 0-0 | = 0 = = 0
THEN $Correct\_C(POI3, U3, U1) = 1.$

If the score $\left|C_{a,i}^{b} - c_{a,i}\right|$ is 0, the function $Correct\_C(i, b, a)$ is equal to 1 (indicating that the decisions or opinions are the same). Conversely, if the score is 1 (meaning the decisions or opinions differ), the same function is equal to 0. In this way, we obtain the Mc matrix below, which contains the success and failure scores of the predictions (see Table 26):

Table 26: The Mc matrix containing the success and failure scores of the predictions

Mc	POI1		POI2		POI3	
U1	U1	?	U1	1	U1	1
	U2	?	U2	0	U2	?
	U3	?	U3	?	U3	1 *correct*
	U4	?	U4	?	U4	0
U2	U1	?	U1	1 *correct*	U1	?
	U2	1	U2	1	U2	?
	U3	1 *correct*	U3	?	U3	?
	U4	0	U4	?	U4	?
U3	U1	?	U1	?	U1	1 *correct*
	U2	0	U2	?	U2	?
	U3	1	U3	?	U3	1
	U4	0	U4	?	U4	0
U4	U1	?	U1	?	U1	0
	U2	1 *correct*	U2	?	U2	?
	U3	0	U3	?	U3	0
	U4	1	U4	?	U4	1

Step5:

To be able to calculate the trust between users based on Equation (38), we use Equations (36) and (37) mentioned earlier. These two formulas allow us to determine the subsets $RecSet\_C(user)$ and $CorrectSet\_C(user)$.

For example, using Table 26 above: if user U3 participated in a prediction involving user U1, U1's row will be selected to calculate how many times U3 made a correct prediction for U1 out of the total number of times U3 was involved with this user (U1).

Using Equation (36), we can calculate the $RecSet\_C(U3)$ as follows:

$$RecSet\_C(U3) = \begin{Bmatrix} \left(C_{u1,poi3}^{u3}, c_{u1,poi3}\right), \left(C_{u2,poi1}^{u3}, c_{u2,poi1}\right), \\ \left(C_{u4,poi1}^{u3}, c_{u4,poi1}\right), \left(C_{u4,poi3}^{u3}, c_{u4,poi3}\right) \end{Bmatrix}$$

Using Equation (37), we can calculate the $CorrectSet\_C(U3)$ as follows:

$$CorrectSet\_C(U3) = \left\{\left(C_{u1,poi3}^{u3}, c_{u1,poi3}\right), \left(C_{u2,poi1}^{u3}, c_{u2,poi1}\right)\right\}$$

Step6: ***(in case of O'donovan: profile-level; adapted to Check-in)***

Using Equation (29), adapted to the check-in case, we can calculate the profile-level as follows (see Table 27):

$$Trust^P(u3) = \frac{card\{CorrectSet(u3)\}}{card\{RecSet(u3)\}} = \frac{2}{4} = 0.5$$

Table 27: Adapted O'Donovan profile-level trust based on check-ins

User	$Trust^P$
U1	2/3 = 0.66
U2	1/3 = 0.33
U3	2/4 = 0.5
U4	0/4 = 0

The Table 27 can be expressed as follows (see Table 28):

Table 28: Adapted O'Donovan profile-level trust based on check-in (user-user trust)

	U1	U2	U3	U4
U1	**1**	0.33	0.5	0
U2	0.66	**1**	0.5	0
U3	0.66	0.33	**1**	0
U4	0.66	0.33	0.5	**1**

Step7: ***(our proposition: user-user Trust)***

Using Equation (38), we can calculate the $Trust\_C^U(u1 \to u3)$ as follows:

$$card\{(C_{j,k}^{u3}, c_{j,k}) \in CorrectSet\_C(u3) \ : \ j = u1\} = card\ \{(C_{u1,poi3}^{u3}, c_{u1,poi3})\} = 1$$

$$card\{(C_{j,k}^{u3}, c_{j,k}) \in RecSet\_C(u3) \ : \ j = u1\} = card\ \{(C_{u1,poi3}^{u3}, c_{u1,poi3})\} = 1$$

$$Trust\_C^U(u1 \to u3) = \frac{card\{(C_{j,k}^{u3}, c_{j,k}) \in CorrectSet\_C(u3) \ : \ j = u1\}}{card\{(C_{j,k}^{u3}, c_{j,k}) \in RecSet\_C(u3) \ : \ j = u1\}} = \frac{1}{1} = 1$$

In the same way, we calculate all of the trust matrix values between users, from which we can derive the TDMC (Trust Derivation Matrix based on Check-ins) shown below (see Table 29).

Table 29: The TDMC matrix representing trust between users based on check-ins

TDMC	U1	U2	U3	U4
U1	**1**	$\frac{0}{1} = 0$	$\frac{1}{1} = 1$	$\frac{0}{1} = 0$
U2	$\frac{1}{1} = 1$	**1**	$\frac{1}{1} = 1$	$\frac{0}{1} = 0$
U3	$\frac{1}{1} = 1$	$\frac{0}{1} = 0$	**1**	$\frac{0}{2} = 0$
U4	$\frac{0}{1} = 0$	$\frac{1}{1} = 1$	$\frac{0}{2} = 0$	**1**

IV.4. The proposed algorithms

After explaining how to calculate the trust between users, noted as TDM (Trust Derivation Matrix), based on their check-ins and ratings of POIs, we propose in this section two algorithms to implement these calculations.

Algorithm 1 below takes as input the rating matrix, noted UPRM (User-POI Rating Matrix), with dimensions m × n (m: number of users and n: number of POIs), to compute the TDMR (Trust Derivation Matrix based on Ratings), which has dimensions m × m (m: number of users).

Algorithm 1 : Trust between users based on POI's ratings

```
INPUT: UPRM: User-POI Rating Matrix
OUTPUT: TDMR: Trust Derivation Matrix based on Ratings
        RPMR: Rate Prediction Matrix based on Ratings
Var M2, M3, M4: Usr-User-POI Matrix of dimension m × m × n
    RecSet, CorrectSet: empty Lists
BEGIN
//Trust between users
FOR each user b DO
  FOR each user a < > b DO
    FOR each POI i DO
      //Compute predict rating M2(a,b,i) using Eq (25)
      M2(a,b,i) ← meanRate(a) + Rate(b,i) – meanRate(b)
      //Compute distance error M4(a,b,i)  using Eq (26)
      M3(a,b,i) ← | Rate(a, i) - M2(a,b,i) |
      IF (M3(a,b,i) < Ɛ) THEN
        M4(a,b,i) ← 1
      ELSE:
        M4(a,b,i) ← 0
      ENDIF
      //the set of user b's recommendations using Eq (27)
      RecSet(b) ← countsum.(M4(a,b,i));
      //the set of user b's correct recommendations using Eq (28)
      CorrectSet(b) ← count sum.(M4(a,b,i) | M4(a,b,i) = 1)
    END FOR
  END FOR
  //Compute user-user trust TDMR(a,b) using Eq (31)
  TDMR(a,b) ← CorrectSet(b) / RecSet(b)
END FOR
//Compute Rating Prediction based on rating trust (RPMR) using Eq (33)
FOR each user a DO
  FOR each POI x DO
    IF (UPRM(user a, POI x) = = empty) THEN
      //b ϵ N set of user a 's neighborhood
      FOR user b DO
        Numerator ← sum (Rate(user b, POI x) - meanRate(user b)) * TDMR(user a, user b)
        Denominator ← sum TDMR(user a, user b)
      END FOR
  END FOR
  RPMR(user a, POI x) = meanRate(user a) + (Numerator / Denominator)
END FOR
END
```

Algorithm 2 below takes as input the check-in matrix, noted UPCM (User-POI Check-in Matrix), with dimensions m × n (m: number of users and n: number of POIs), to compute the TDMC (Trust Derivation Matrix based on Check-ins), which has dimensions m × m.

Algorithm 2 : Trust between users based on users's check-ins

```
INPUT: UPCM: User-POI Check-in Matrix
       UPRM: User-POI Rating Matrix
OUTPUT: TDMC: Trust Derivation Matrix based on Check-ins
        RPMC: Rating Prediction Matrix based on Check-ins
Var Ma, Mb, Mc: User-User-POI Matrix of dimension m × m × n
    RecSet, CorrectSet: empty List
BEGIN
//Trust between users
FOR each user b DO
  FOR each user a < > b DO
    FOR each POI i DO
    //Compute predict check-in Ma(a, b, i) using Eq (34)
    Ma(a,b,i) ← meanCheck(a) + Check(b,i) – meanCheck(b)
    //Compute distance error Mb(a,b,i) using Eq (35)
    Mb(a,b,i) ← | Check(a, i) - Mb(a,b,i) |
      IF (Mb(a,b,i)  = = 0) THEN
          Mc(a,b,i) ← 1
      ELSE:
          Mc(a,b,i) ← 0
      ENDIF
    //the set of user b's recommendations using Eq (36)
    RecSet(b) ← sum.(Mc(a,b,i));
    //the set of user b's correct recommendations using Eq (37)
   CorrectSet(b) ← sum.(Mc(a,b,i) | Mc(a,b,i) = 1)
   END FOR
 END FOR
//Compute user-user trust TDMC(a,b) using Eq (38)
TDMC(a,b) ← CorrectSet(b) / RecSet(b)
END FOR
//Compute Rating Prediction based on check-in trust (RPMC) using Eq (39)
FOR each user a DO
  FOR each POI x DO
    IF (UPRM(user a, POI x) = = empty) THEN
      //b ϵ N set of user a 's neighborhood
      FOR user b DO
          Numerator ← sum (Rate(user b, POI x) - meanRate(user b)) * TDMC(user a, user b)
          Denominator ← sum TDMC(user a, user b)
      END FOR
   END FOR
    RPMC(user a, POI x) ← meanRate(user a) + (Numerator / Denominator)
END FOR
END
```

These two algorithms use Equations (33) and (39) to calculate their prediction matrices of dimension m × n (m: number of users and n: number of POIs), respectively denoted RPMR (Rating Prediction Matrix based on Ratings) and RPMC (Rating Prediction Matrix based on Check-ins).

IV.5. Proposed Model

This section presents in detail the POI recommendation approach proposed by the model named Hybrid Rating Check-in Trust (HRCT), which is based on POI ratings and user check-ins on the one hand, and on the two algorithms explained in Section IV.4 on the other hand. The first algorithm (Algorithm 1) uses the rating matrix, denoted UPRM, to first calculate the user-user trust matrix, denoted TDMR, and then the matrix RPMR, which contains the predicted POI ratings by the users. The second algorithm (Algorithm 2) uses the check-in matrix, denoted UPCM, to calculate the user-user trust matrix, denoted TDMC. This matrix is then used to compute the RPMC matrix, which contains the predicted POI ratings by the users. Next, the two trust matrices (TDMR and TDMC) obtained from Algorithm 1 and Algorithm 2 above can be combined using Algorithm 3 below to produce the H-Trust matrix of dimension m × m (m: number of users). This matrix can be used to calculate the predicted POI ratings in the RPMH matrix (Rating Prediction Matrix based on H-Trust) of dimension m × n (m: number of users and n: number of POIs), using Equation (42) below, as indicated in Algorithm 3:

Equation 42: Rating Prediction based on H-Trust combined trust between users

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N} (r_{b,x} - \overline{r_b}) * H\text{-}Trust^{U}\ (a \to b)}{\sum_{b=1}^{N} H\text{-}Trust^{U}\ (a \to b)}$$

Where:

- $P_{a,x}$: The predicted rating for user a on item x.
- $\overline{r_a}$: The average rating of user a for all items.
- $r_{b,x}$: The actual rating given to item x by user b.
- $H - Trust^{U}\ (a \to b)$: The trust based on ratings and check-ins.

Algorithm 3 : User-user trust based on ratings and check-ins fusion

```
INPUT: TDMR: Trust Derivation Matrix based on Ratings
       TDMC: Trust Derivation Matrix based on Check-ins
       UPRM: User-POI Rating Matrix
OUTPUT: H-Trust: User-user hybrid trust matrix
        RPMH: Rating Prediction Matrix based on Hybrid Trust
BEGIN
//Combine trust
FOR each user x DO
 FOR each user y < > x DO
  IF (TDMR(x,y) exist AND TDMC(x,y) exist) THEN
     H-Trust(x,y) ← 2 * TDMR(x,y) * TDMC(x,y) / (TDMR(x,y) + TDMC(x,y))
  ELSEIF (TDMR(x,y) exist and TDMC(x,y)! exist) then
     H-Trust(x,y) ← TDMR(x,y)
  ELSEIF (TDMR(x,y)! exist and TDMC(x,y) exist) then
     H-Trust(x,y) ← TDMC(x,y)
  ELSE:
     H-Trust(x,y) ← 0
  ENDIF
 END FOR
END FOR
//Compute Rating Prediction based on Hybrid trust (RPMH) using Eq (42)
FOR each user a DO
 FOR each POI x DO
   IF ( UPRM(user a, POI x) = = empty ) THEN
     //b ϵ N set of user a 's neighborhood
     FOR user b DO
       Numerator ← sum (Rate(user b, POI x) - meanRate(user b))  * H-Trust(a,b);
       Denominator ← sum H-Trust(a,b);
     END FOR
   ENDIF
 END FOR
 RPMH(user a, POI x) ← meanRate(user a) + (Numerator / Denominator);
END FOR
END
```

In Figure 4 below, the HRCT model consists of five main steps (from a to e in Figure 4), and each step of this model is described using the data it processes and the algorithms it uses. These steps can be summarized as follows:

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

(b) After loading the rating/check-in data (the UPRM and UPCM matrices), the HRCT model can infer Algorithm 1 and Algorithm 2 (see arrows I.1 and II.1 in Figure 4) to calculate the user-user trust matrices TDMR and TDMC. Next, these two matrices enable the execution of Algorithm 3 (see arrow III.1 in Figure 4), which can derive the H-Trust matrix. Finally, these three trust matrices: TDMR, TDMC, and H-Trust are used to compute the three rating prediction matrices respectively: RPMR, RPMC, and RPMH (see arrows I.2, II.2, and III.2 in Figure 4).

(c) These three prediction matrices make it possible to generate three lists of POIs, where each list contains K POIs ranked in descending order (from the most interesting/relevant POI to the least interesting/relevant one). These three lists can be merged and displayed on a map-based interface.

(d) After browsing the available POIs on the map, the user can select the POI that meets their expectations and then either assign a rating to it or perform a check-in at that location.

(e) The rating of this user and their check-in will be added to the UPRM and UPCM matrices to enrich the dataset to be used for the system's future recommendations.

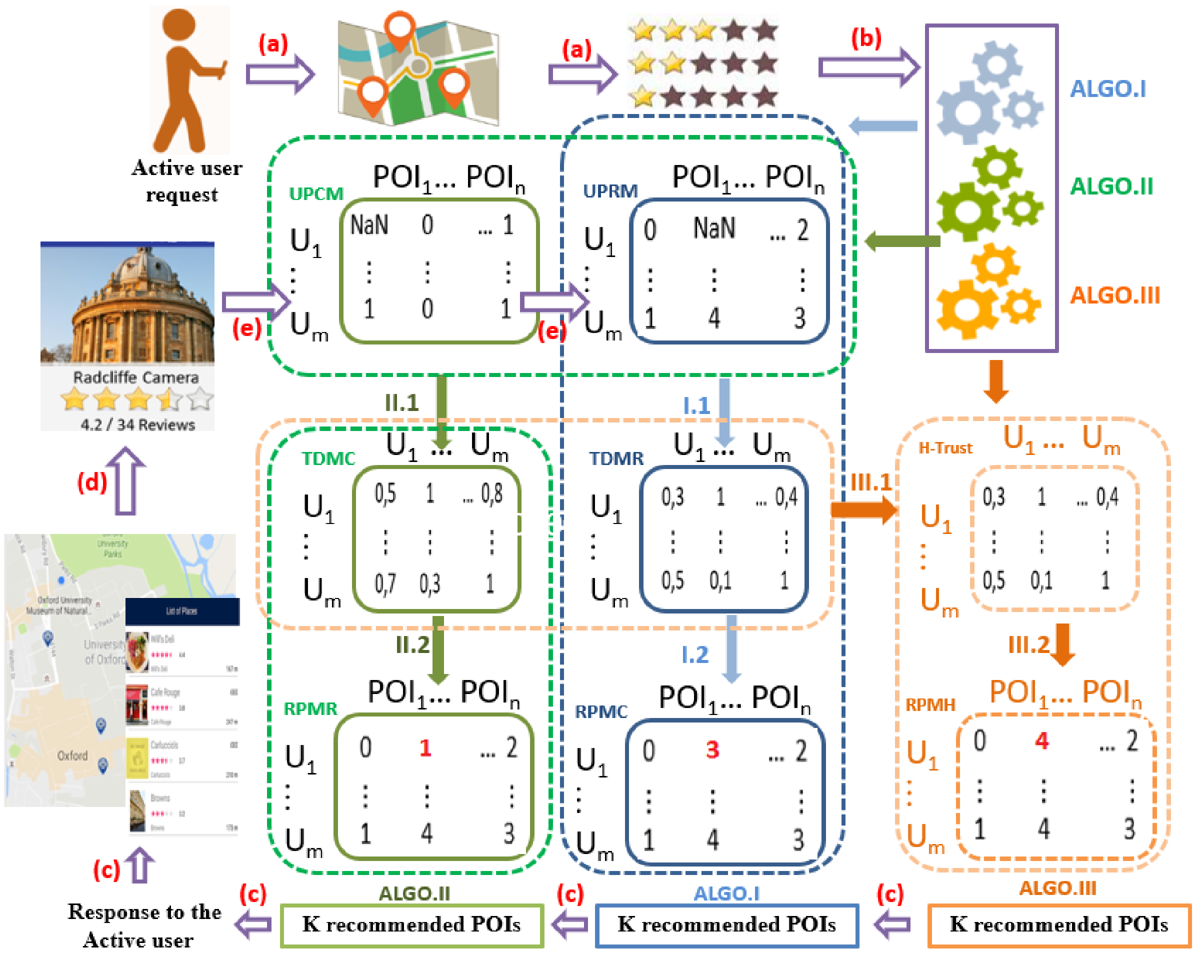

Figure 4: Functional Description of the HRCT Model

IV.6. Experimentation and Results

This section is dedicated to evaluating the performance of our HRCT model. To achieve this goal, parameters such as RMSE, PRECISION, and RECALL are calculated based on a dataset from the testing phase of an LBSN. These parameters are used to compare the three variants of the HRCT model corresponding to the three algorithms previously presented. Then, these three variants are compared with traditional collaborative filtering (CF) approaches that rely on similarity metrics, including Pearson Correlation (CF-PCC), Cosine Similarity (CF-Cosine), and Jaccard Similarity (CF-Jaccard). Next, several combinations between the HRCT model variants and algorithms using the aforementioned similarity measures are tested to improve the performance of our approach. Finally, a data sparsity analysis is conducted to demonstrate the contribution of our model in addressing this issue. It should be noted that, in what follows, the results of Algorithm 1, Algorithm 2, and Algorithm 3 are referred to as R-Trust, C-Trust, and H-Trust, respectively.

IV.6.1. Experimental Setup

In order to compute the evaluation parameters (RMSE, PRECISION and RECALL) for comparing the HRCT model variants (R-Trust, C-Trust, and H-Trust) with other approaches (PCC, Cosine and Jaccard) during the LBSN testing phase, a dataset (currently being collected) is used (see Table 30), and a set of hyperparameters is defined in Table 31. This dataset includes user interactions with POIs through ratings and check-ins, and the hyperparameters concern the settings to be adopted for all the comparisons presented in the next subsection.

Table 30: Description of Dataset Columns

Column name	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 31: List of the HRCT Hyperparameters

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

IV.6.2. Evaluation metrics

To evaluate the performance of the HRCT model, the dataset and hyperparameters described above are used along with the RMSE, PRECISION, and RECALL metrics.

- **The RMSE metric**

The RMSE parameter is used to evaluate the difference between the actual user rating (denoted as r_i) and the predicted rating (denoted as P_i) by the recommender system using the HRCT model (Hadjhenni et al., 2024). This parameter is calculated using Equation (43) below:

Equation 43: Root Mean Square Error (RMSE) for User Prediction Accuracy

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

Where:

- r_i : The rating *i* provided by this user.
- P_i: The predicted rating *i* inferred from a given model.
- n : The total number of ratings made by the user.

This metric is used to evaluate the accuracy of the predicted ratings generated by the different variants of the HRCT model, and to compare them with other existing POI recommendation approaches from the literature.

- **The PRECISION and RECALL metrics:**

PRECISION and RECALL are widely used to evaluate the quality of the POI list provided by a recommender system (RS). The PRECISION of an RS for a user *i* measures the proportion of truly relevant POI recommendations within the list of recommended POIs, as indicated by Equation (44) below:

Equation 44: PRECISION of the Recommendation System (RS)

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

Where:

- $POIs_{rec\ and\ pert}$: The set of recommended POIs which are pertinent for the user *i*.
- $POIs_{rec}$: The set of recommended POIs for user *i*.

RECALL measures the ratio of truly relevant POI recommendations among all relevant POIs for a user *i*, as indicated in Equation (45) below (Débora Nice Ferrari Barbosa et al., s. d.):

Equation 45: RECALL of the Recommendation System (RS)

$$Recall_{RS}(i) = \frac{Card\{POIs_{rec\ and\ pert}\}}{Card\ \{POIs\ pert\}}$$

Where:

- $POIs_{pert}$: the set of pertinent POIs for utilisateur *i*.

In this work, these two metrics are used to assess the quality of the recommendations provided by the three variants of our HRCT model on the one hand, and to compare these recommendations with those obtained from other state-of-the-art models on the other hand.

IV.6.3. Comparison between the Variants of the HRCT Model

To compare the three variants of the HRCT system (Algorithm 1 referred to as R-Trust, Algorithm 2 as C-Trust, and Algorithm 3 as H-Trust), the dataset (detailed in Table 30), which contains user ratings and check-ins of POIs, is divided into two parts based on the hyperparameters in Table 31: 80% for training and 20% for testing (see arrows I.1 and II.1 in Figure 5). The system then takes the first part (80% of the dataset) as input for Algorithms 1, 2 and 3, as explained in Subsections IV.4 and IV.5. 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 5. Finally, the trust matrices derived from the training portion of the dataset will be used to predict the ratings corresponding to the test portion of the same dataset (see arrows I.3 and II.2 in Figure 5). 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 used to evaluate the accuracy and quality of the POI recommendations (see arrow III in Figure 5).

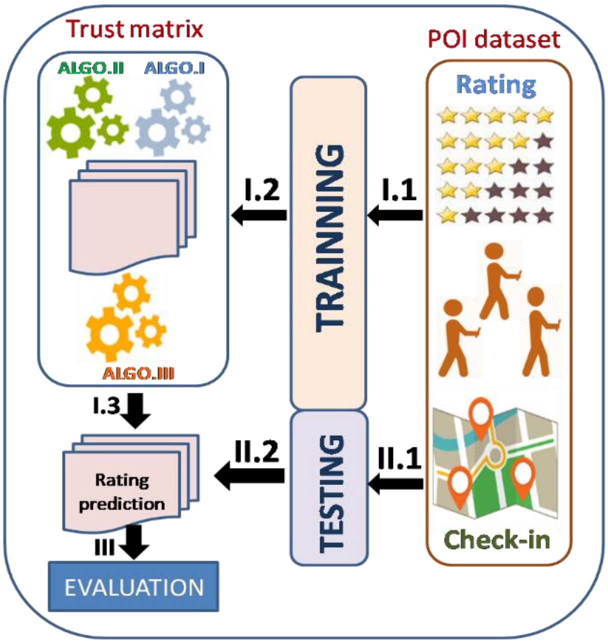

Figure 5: Evaluation Framework for the Three Variants of the HRCT Model

In the following, Figure 6, 7 and 8 illustrate a comparison of the three variants of the HRCT model regarding RMSE, PRECISION and RECALL.

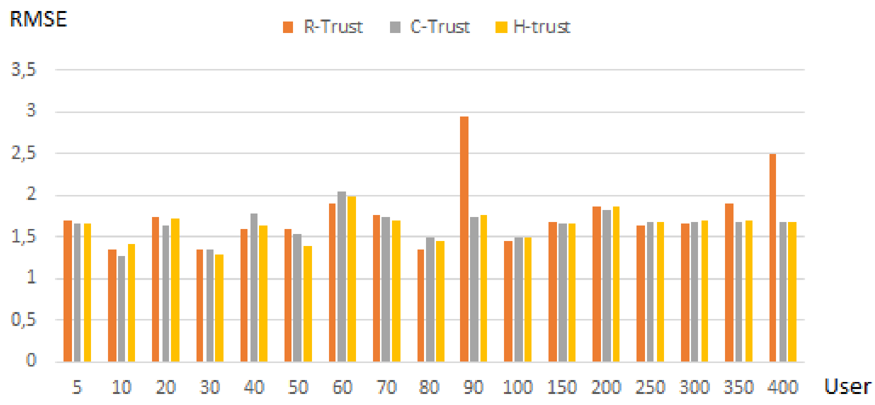

Figure 6: RMSE Metric Comparison of R-Trust, C-Trust, and H-Trust

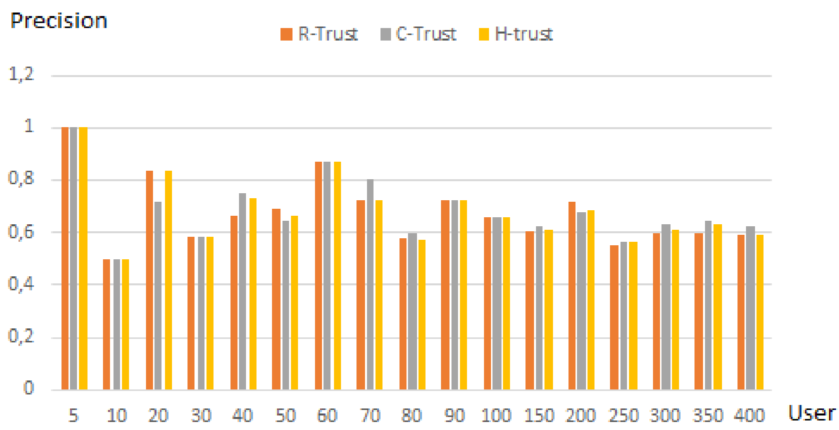

Figure 7: PRECISION Metric Comparison of R-Trust, C-Trust, and H-Trust

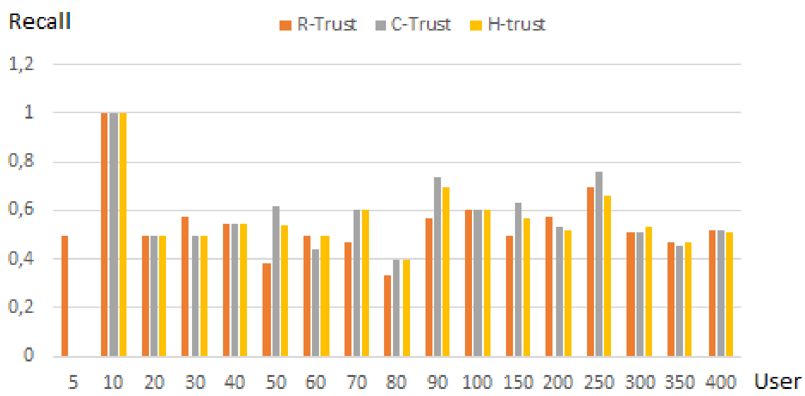

Figure 8: RECALL Metric Comparison of R-Trust, C-Trust, and H-Trust

In Table 32 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 32: 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	0,6753	**0,6828**	0,6794	0,1277	0,1206	0,1259
RECALL	0,5433	**0,5495**	0,5375	0,1427	0,2020	0,1905
RMSE	1,7605	1,6410	**1,6314**	0,4087	0,1818	0,1740

IV.6.4. Comparison 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) (Sheugh & Alizadeh, 2015) similarity, Cosine similarity (Inderprastha Engineering College, AKTU et al., 2020) and Jaccard similarity (Bag et al., 2019), as well as user check-ins, also employing Pearson Correlation Coefficient (PCC) similarity, Cosine similarity (Zeng et al., 2020), (L. Guo et al., 2017), (Song et al., 2019) and Jaccard similarity (Eravci et al., 2016). To achieve this objective, we divide the dataset described in Table 30 into two segments: 80% for training and 20% for testing (as depicted by arrow I and arrow II in Figure 9), employing the same hyperparameters outlined in Table 31. The initial portion (80% of the dataset) serves as input for computing the user-user trust matrices (R-Trust, C-Trust and H-Trust), which will be used to predict the POI ratings, as indicated by arrow a.1 in Figure 9 below. This same training portion (80% of the dataset) is also used 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 9 below.

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 9). 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 9).

These predictions derived from these two processes (trust and similarity) can be compared using the RMSE 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 used to compare the quality of POI recommendations from these similarity models compared to the HRCT model (refer to arrows a.4 and b.4 in Figure 9).

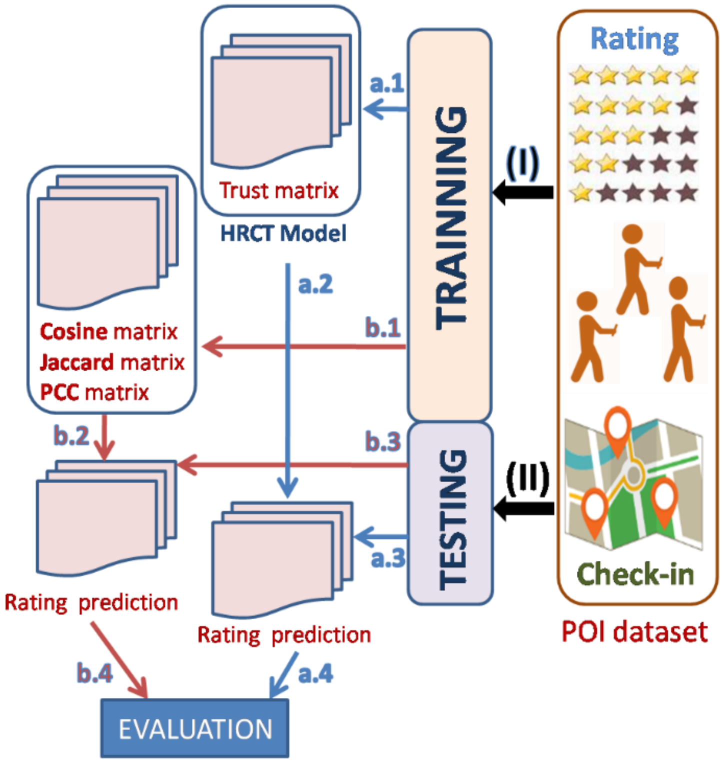

Figure 9: Method for comparing the HRCT model with other models

Figure 10, 11 and 12 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.

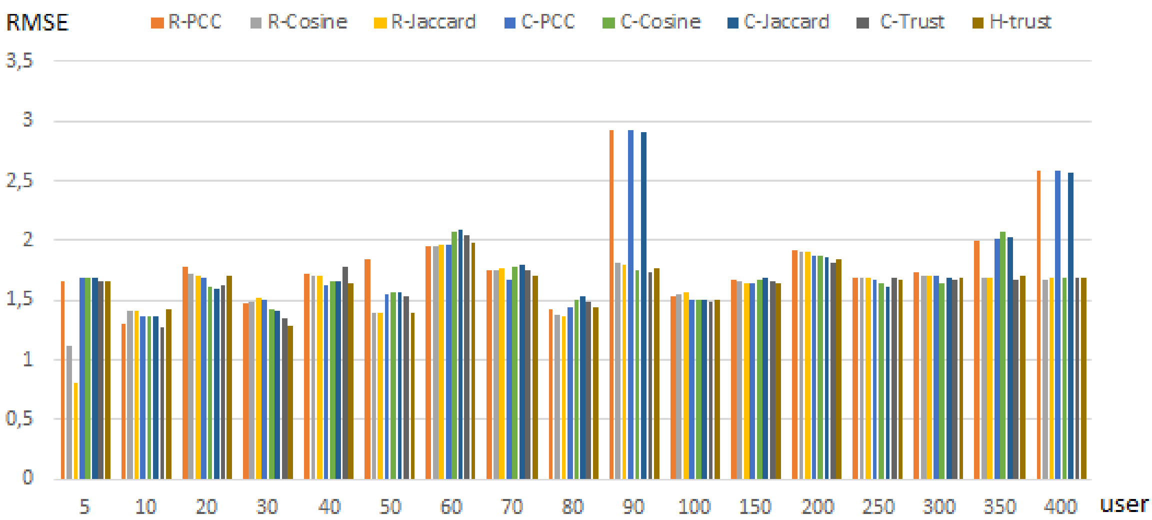

Figure 10: Comparison between the C-Trust and H-Trust variants of the HRCT model and the similarity-based approaches R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine, and C-Jaccard using the RMSE metric

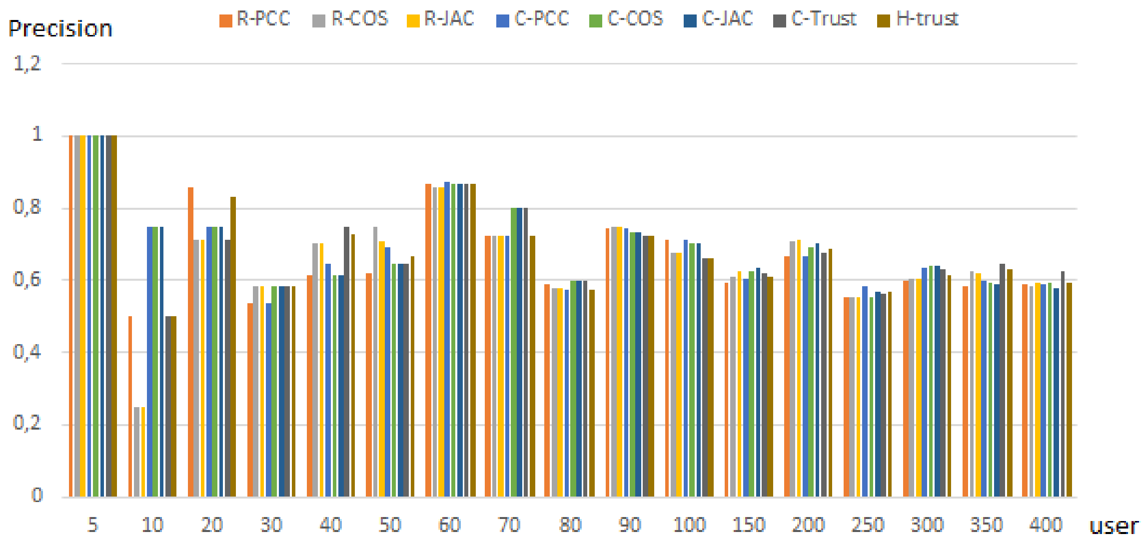

Figure 11: Comparison between the C-Trust and H-Trust variants of the HRCT model and the similarity-based approaches R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine, and C-Jaccard using the PRECISION metric

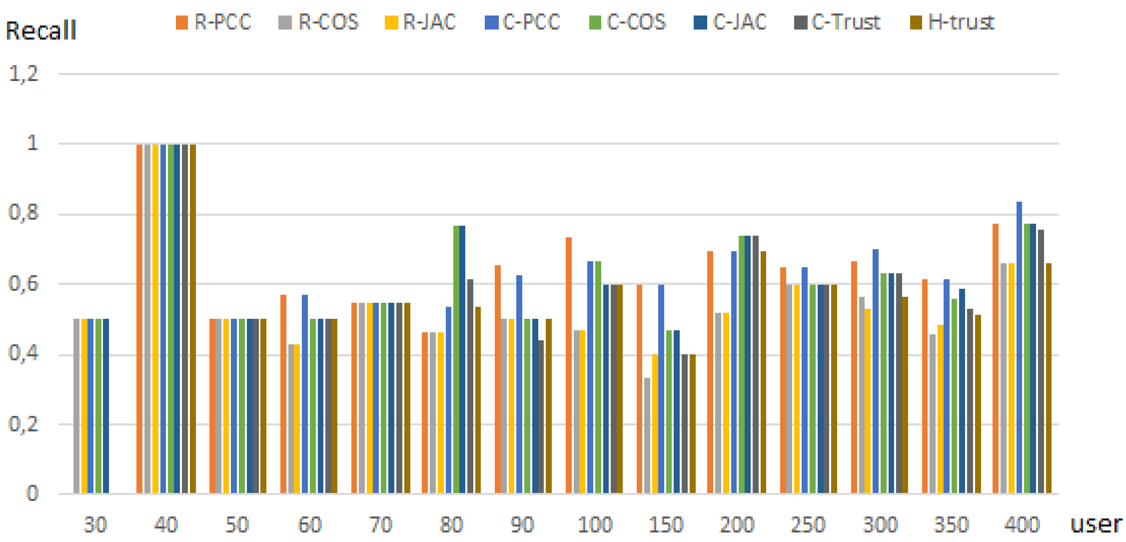

Figure 12: Comparison between the C-Trust and H-Trust variants of the HRCT model and the similarity-based approaches R-PCC, R-Cosine, R-Jaccard, C-PCC, C-Cosine, and C-Jaccard using the RECALL metric

In the following, Table 33 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 enable a comparison between the H-Trust and C-Trust variations 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-user similarities based on their ratings, while C-PCC, C-Cosine, and C-Jaccard are also user-user similarities, but based on their check-ins.

Table 33: 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 Table 33, 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.

IV.6.5. 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 13). These combinations will allow for two types of predictions. The first type of predictions concerns the combination of Algorithm1, denoted as R-Trust (trust based on POI’s Ratings), 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.

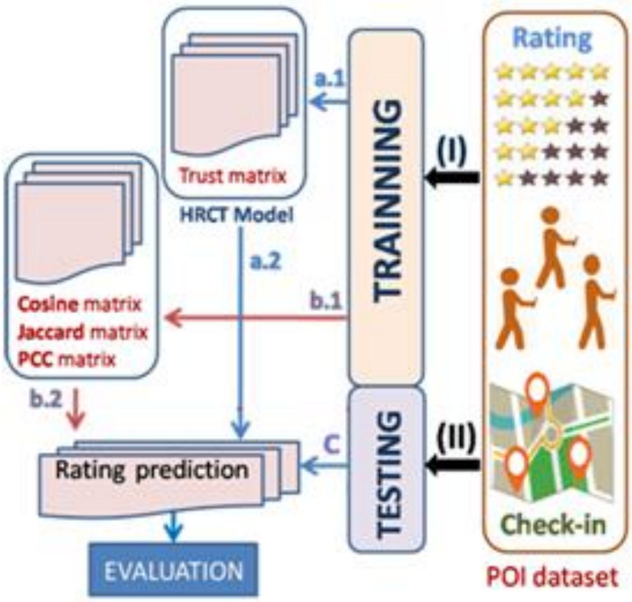

Figure 13: Combining the HRCT Model with PCC, Cosine, and Jaccard Similarities in the Rating Prediction Phase

In the following, Figure 14, 15 and 16 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-COS, and R-Trust-JAC, using RMSE, PRECISION and RECALL metrics.

These are hybrid approaches that combine similarity and trust, specifically trust derived from ratings (R-Trust), with different similarity measures:

a) **R-Trust-PCC**: a recommender system that combines R-Trust (trust based on ratings) with the Pearson Correlation Coefficient (PCC) similarity.
b) **R-Trust-COS**: a recommender system that combines R-Trust with Cosine similarity.
c) **R-Trust-JAC**: a recommender system that combines R-Trust with Jaccard similarity.

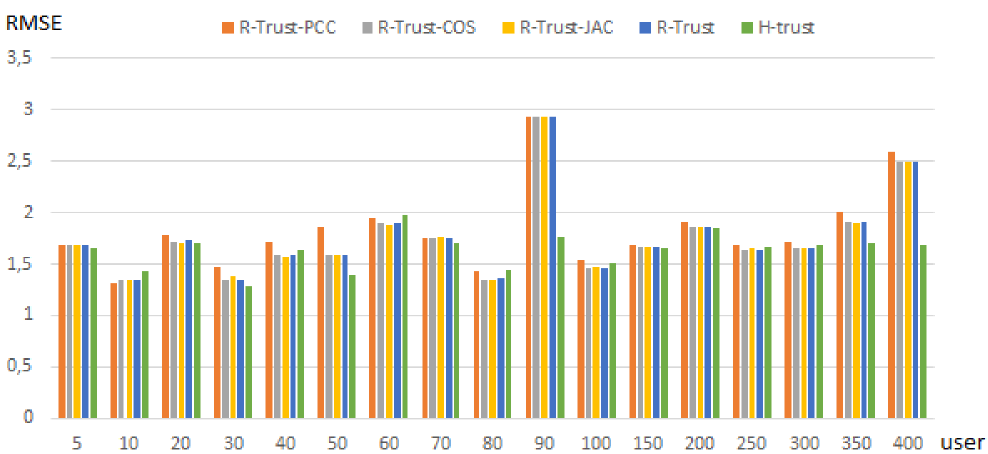

Figure 14: RMSE-based comparison of the R-Trust and H-Trust variants of the HRCT model against R-Trust combined with PCC, Cosine, and Jaccard measures

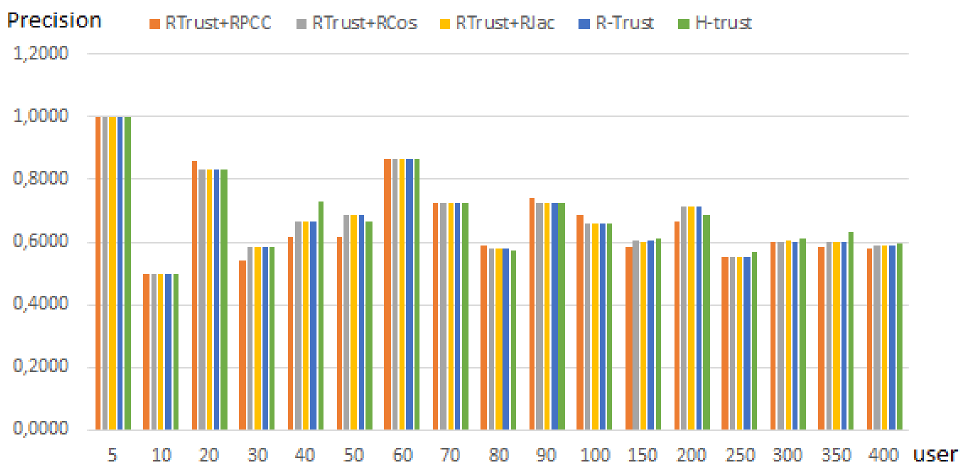

Figure 15: PRECISION-based comparison of the R-Trust and H-Trust variants of the HRCT model with R-Trust combined with PCC, Cosine, and Jaccard similarity measures

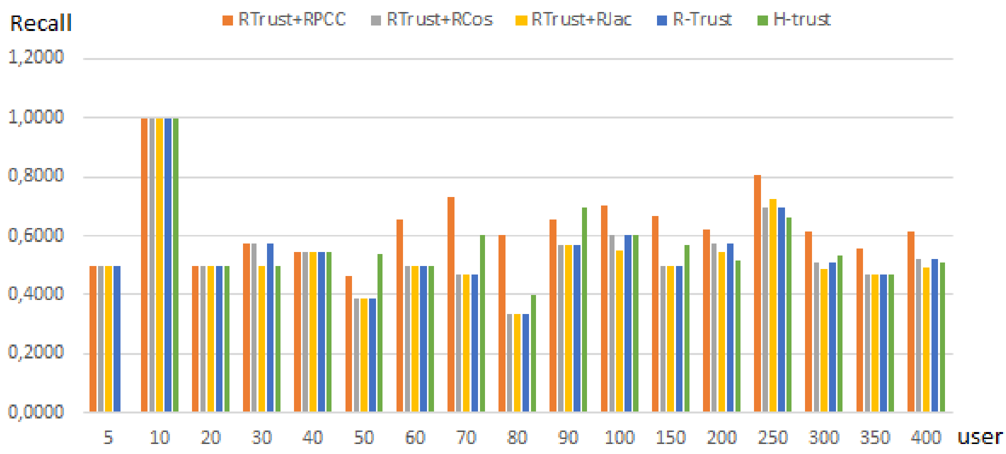

Figure 16: RECALL-based comparison of the R-Trust and H-Trust variants of the HRCT model against R-Trust combined with PCC, Cosine, and Jaccard similarity measures

Table 34 below compares the prediction performances of three combinations (R-Trust-PCC: R-Trust + PCC similarity, R-Trust-COS: R-Trust + Cosine similarity and R-Trust-JAC: R-Trust + Jaccard similarity) using two algorithms: Algorithm 1 (R-Trust) and Algorithm 3 (H-Trust).

Table 34: Comparison of the combinations of R-Trust with PCC, Cosine, and Jaccard using the RMSE, PRECISION, and RECALL metrics

	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

As shown in Table 34, the recommendation algorithm R-Trust-JAC, which results from a combination of trust based on POI ratings with the Jaccard similarity between these same POIs, provides better differences between predicted ratings and actual ratings (RMSE) when compared to the algorithms R-Trust-PCC, R-Trust-Cosine, and R-Trust. However, the R-Trust-JAC algorithm performs less effectively than Algorithm 3 (H-Trust) of the HRCT model.

In the following, Figure 17, 18 and 19 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-COS and C-Trust-JAC, evaluated using RMSE, PRECISION and RECALL parameters.

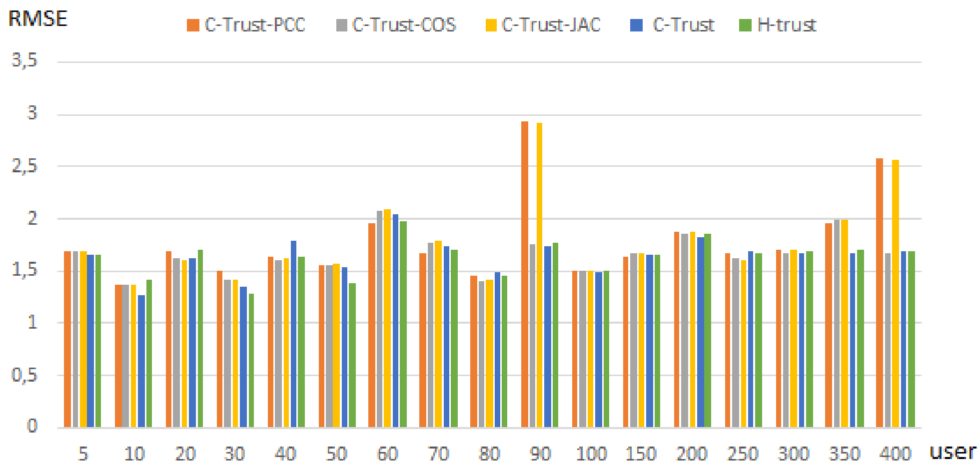

Figure 17: RMSE-based comparison of the C-Trust and H-Trust variants of the HRCT model with C-Trust combined with PCC, Cosine, and Jaccard similarity measures

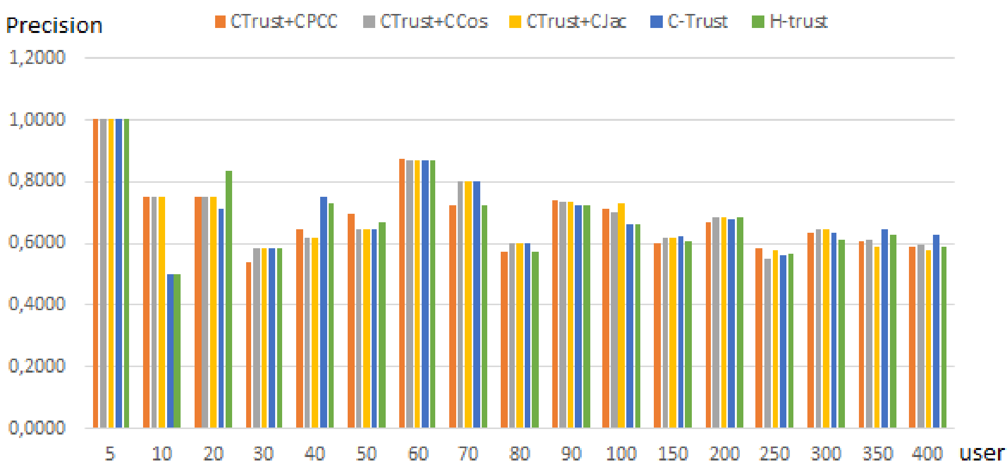

Figure 18: PRECISION-based comparison of the C-Trust and H-Trust variants of the HRCT model against C-Trust combined with PCC, Cosine, and Jaccard

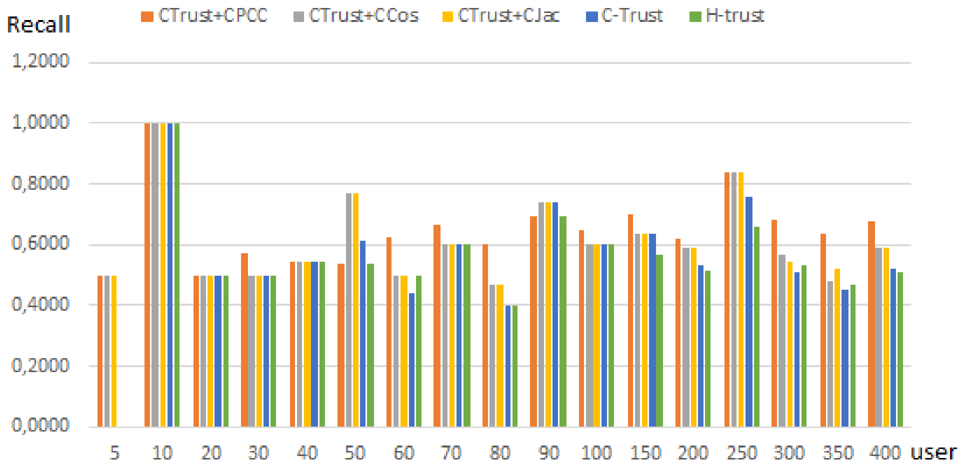

Figure 19: RECALL-based comparison of the C-Trust and H-Trust variants of the HRCT model against C-Trust combined with PCC, Cosine, and Jaccard

Similarly, a comparison of the prediction performance between the three combinations (C-Trust-PCC: C-Trust + PCC similarity, C-Trust-COS: C-Trust + Cosine similarity, and C-Trust-JAC: C-Trust + Jaccard similarity) with the two algorithms; Algorithm 2 (C-Trust) and Algorithm 3 (H-Trust) is performed in Table 35 below.

Table 35: Comparison of the combinations of C-Trust with PCC, Cosine, and Jaccard using the RMSE, PRECISION, and RECALL metrics

	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

As shown in Table 35, the recommendation algorithm C-Trust-COS, which results from the combination of trust based on POI check-ins with Cosine similarity performs better than the other two combination algorithms (C-Trust-Jaccard 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).

IV.6.6. 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 Equation (46) below (Singh, 2020):

Equation 46: Trust Matrix Sparsity

$$Sparsity = 1 - \ Density \quad (Density = \frac{total\ non{-}zero\ Trust\ values}{no.\ of\ users * no.\ of\ users})$$

Where:

- $total\ non - zero\ Trust\ values$: The trust values between users.
- $no.\ of\ users$: The number of users in the user-user trust matrix.

The Equation above quantifies the sparsity of data in a given matrix. A higher value indicates that the matrix contains more 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 4) are merged using Algorithm 3 to create the H-Trust which suppoe to be 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-COS and H-JAC 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-COS and H-JAC) is presented in Table 36 below.

Table 36: Comparison of the sparsity of the H-Trust matrix with the sparsity of the H-PCC, H-COS, and H-JAC 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**	57,00436	44,7805	48,6831

Table 36 indicates that the H-Trust matrix is denser than the H-PCC, H-COS and H-JAC matrices. Additionally, this trust matrix can utilize the principle of trust propagation to further reduce its sparsity percentage.

IV.6.7. Summary of Results and Discussion

At the initial stage of the experimentation phase, the three HRCT model variants were 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 accuracy 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 assessed uniformly, using the same dataset and identical hyperparameters. The results show that the H-Trust algorithm outperforms certain similarity-based models based on ratings, but is less effective than others.

Finally, combinations of the HRCT model with other similarity models are explored. The performance is compared using the same evaluation metrics. The results show that some combinations yield better performance than others, but the HRCT model remains competitive in most cases. Moreover, it is noted that the trust matrix of the HRCT model is denser than the other similarity matrices, which may contribute to improving the quality of the recommendations.

The results of the experimental tests demonstrate that the HRCT model can deliver better performance in terms of Root Mean Square Error (RMSE) and PRECISION/RECALL compared to state-of-the-art algorithms, such as user-user collaborative filtering based on Pearson similarity, Cosine similarity, and Jaccard similarity. Moreover, this model helps address the data sparsity issues found in user-user similarity matrices derived from LBSNs, as it offers a strong alternative by leveraging and combining data from POI ratings and user check-ins to generate denser user-user trust matrices. Finally, since the POI recommendation system using the HRCT model is still in the testing phase and our new LBSN is currently in the data collection phase, it would be beneficial to incorporate user feedback regarding the implicit trust suggestions.

Conclusion

In this chapter, we explored the inference of implicit trust between users of an LBSN based on two key sources: point-of-interest (POI) ratings and user check-ins. This trust is initially modeled through two separate matrices, each derived from one of these data sources. Subsequently, these matrices are merged to create a combined trust matrix, leveraging both the preferences expressed through ratings and the visit patterns revealed by check-ins. Building on this foundation, we developed the HRCT system, which relies on three trust matrices: (1) the TDMR matrix, derived from POI ratings, (2) the TDMC matrix, based on user check-ins, and (3) the H-Trust matrix, resulting from the combination of the first two. As a future perspective, we plan to integrate the propagation of implicit trust in LBSNs by leveraging the fused matrix of ratings and check-ins. This approach, which will be addressed in the next chapter, aims to further reduce the sparsity of user-user trust matrices and enhance the accuracy of recommendations.

Chapter V:

The PRCT Model Integrating Propagation For Predictions

V.1. Introduction

The propagation of implicit trust between users represents a promising approach to overcoming challenges related to cold-start and data sparsity in point-of-interest (POI) recommendation systems within location-based social networks (LBSNs). In this context, we have analyzed its impact on improving the accuracy of recommendations.

This chapter details the integration of user interaction analysis, through their ratings and check-ins, into an innovative recommendation model called PRCT (Propagation of Rating/Check-in for implicit Trust). We then compare the two complementary approaches within this model: Similarity Trust Rating (STR), which leverages user ratings, and Similarity Trust Check-in (STC), which relies on check-ins. Finally, we explain how the PRCT model can increase the density of its similarity matrices and enhance its predictive performance by incorporating trust propagation.

V.2. Problem Definition

Location-Based Social Networks (LBSNs) play a crucial role in various domains, particularly in smart tourism, by influencing user choices and behaviors through personalized recommendations. These recommendations, whether for hotels, restaurants, or historical sites are based on users' interactions with Points of Interest (POIs), particularly through their ratings and their check-ins.

However, the quality and relevance of these recommendations are limited by two major issues:

Data sparsity: A large number of users interact with only a small subset of POIs, making it difficult to establish meaningful similarity relationships between users.

Cold-start problem: New users and new POIs lack sufficient interaction history, making it challenging to generate accurate recommendations.

A key factor that can enhance the relevance of recommendations is implicit trust, which propagates among users based on their behaviors and interactions on LBSNs. Although not explicitly stated, this trust can be inferred from the collected data and used to improve POI recommendations.

The objective of this study is to evaluate the impact of implicit trust propagation on improving POI recommendations and to examine how it can help mitigate the problem of data sparsity. To this end, we explore methods for inferring implicit trust from user interactions with POIs and apply a trust propagation principle to enrich the density of similarity matrices.

As part of this research, we adopted an implicit trust propagation approach, which allows for inferring indirect trust relationships between users by considering multiple possible paths within a trust network. To this end, we propose a recommendation model, PRCT (Propagation of Rating/Check-in for implicit Trust), which relies on two main approaches:

• Similarity Trust Rating (STR): based on user ratings.
• Similarity Trust Check-in (STC): leveraging user check-ins.

The objective of the PRCT model is to improve the accuracy of POI recommendation predictions by leveraging these two types of similarities, while applying a trust propagation mechanism to enrich the similarity matrices between users.

Finally, to validate our approach, we developed and evaluated the algorithms to implement this model. We tested the PRCT model on the Yelp dataset, using standard metrics such as RMSE, PRECISION, and RECALL, and compared its performance with other approaches from the literature, including O'Donovan's trust-based models as well as those based on Pearson, Jaccard, and Cosine similarity measures.

V.3. Problem Formulation

This section first explains how to infer implicit trust between users based on their ratings and check-ins of POIs. Then, it explores the application of trust propagation principles to enhance the density of the user-to-user trust matrix.

V.3.1. Implicit Trust Calculation

O'Donovan and Smith define trust based on the reliability of a partner's profile in providing accurate recommendations in the past. For example, a profile that has consistently made accurate recommendations will be considered more trustworthy than one that has mostly made inaccurate predictions. This type of assessment can be determined using Equation (47) below (O’Donovan & Smyth, 2005; Resnick et al., 1994).

In what follows, note that **RC** refers to either the rating of a given POI or a user's check-in on a given POI.

Equation 47: Rating Prediction Adapted for LBSN Context, R (rating) & C (check-in)

$$P_{a,i} = \overline{RC_a} + \frac{\sum_{b=1}^{N}(RC_{b,i} - \overline{RC_b}) * sim(a,b)}{\sum_{b=1}^{N}|sim(a,b)|}$$

Where:

- $P_{a,i}$: The predicted rating that user *a* might give, or the check-in they might check in to item *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) given to POI *i* by user *b*.
- $sim(a, b)$: The similarity between user *a* and user *b*.
- N : The set of neighbors of user *a*.

However, to calculate the predicted rating (or check-in) of a user *a* for a given POI *i* based on a single user *b* considered as the recommender (O'Donovan & Smyth, 2005), Equation (48), derived from Equation (47), can be used (Hwang & Chen, 2007; Shambour & Lu, 2011):

Equation 48: Rating/Check-in Prediction Based on a Single Recommender's Influence Adapted for LBSN

$$P_{a,i}^{b} = \overline{RC_a} + (RC_{b,i} - \overline{RC_b})$$

Where:

- $P_{a,i}^{b}$: The predicted rating that user *a* might give, or the check-in they might check in to 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 (check-in) given 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 Equation (49).

Equation 49: Rating or Check-in based Correct Function Adapted for LBSN

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

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 Equation (50) below to define $RecSet(b)$ as the complete set of recommendations in which user *b* acted as a recommender:

Equation 50: The set of recommendations in case of Ratings or Check-ins

$$RecSet(b) = \{(P^b_{1,1}, RC_{1,1}), \ldots, (P^b_{m,n}, RC_{m,n})\}$$

Where:

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

From $RecSet(b)$, the subset of correct recommendations, denoted as $CorrectSet(b)$, is calculated using the Equation (51) below (O'Donovan & Smyth, 2005).

Equation 51: The set of correct recommendations in case of Ratings or Check-ins Data

$$CorrectSet(b) = \{(P^b_{j,k}, RC_{j,k}) \in RecSet(b) : Correct(k, b, P^b_{j,k})\}$$

Finally, the notion of trust at the profile-level, denoted as $Trust^P$ for a recommender b, can be defined by the percentage of correct recommendations out of all the recommendations in which this recommender has participated, using the Equation (52) below (O'Donovan & Smyth, 2005).

Equation 52: Profile-Level Trust Adapted for LBSN based on Ratings or Check-ins Data

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

Based on Equation (52), a more refined trust metric at the item-level, denoted as $Trust^I$, can be defined to measure only the percentage of correct recommendations made by a recommender b for a specific POI i, out of all their recommendations, as indicated in Equation (53) (O'Donovan & Smyth, 2005).

Equation 53: Item-Level Trust Adapted for LBSN based on Ratings or Check-ins Data

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

Equation (52) serves to represent a user's reputation by enabling the computation of their overall trust of a given user based on their common ratings (or check-ins) across all POIs (Hwang & Chen, 2007; Shambour & Lu, 2011). On the other hand, Equation (53) emphasizes the reputation of a specific user among all users based on their common ratings (or check-ins) for a particular POI.

In the same context, but inspired by the work of (Zahir et al., 2019), 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 Equation (54) (Medjroud et al., 2022) :

Equation 54: User-user Trust based on Ratings or Check-ins

$$Trust^{U}(a \to b) = \frac{card\left\{\left(P_{j,k}^{b}, RC_{j,k}\right) \in CorrectSet(b) : j=a\right\}}{card\left\{\left(P_{j,k}^{b}, RC_{j,k}\right) \in RecSet(b) : j=a\right\}}$$

Where $Trust^{U}(a \to b)$ represents the trust of user a towards a 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) across all POIs.

Based on Equation (54), the trust of user a towards recommender b for a particular POI i, denoted as $ust^{U}(a \to b, i)$, can be derived from the percentage of correct recommendations in which recommender b has participated with user a, based solely on that POI i, as indicated in Equation (55) below:

Equation 55: Item-Specific Trust Estimation Between Users based on Ratings or Check-ins

$$Trust^{U}(a \to b, i) = \frac{card\left\{\left(P_{j,k}^{b}, RC_{j,k}\right) \in CorrectSet(b) : j=a \,\&\, k=i\right\}}{card\left\{\left(P_{j,k}^{b}, RC_{j,k}\right) \in RecSet(b) : j=a \,\&\, k=i\right\}}$$

In the following, we used Equation (54) to deduce implicit trust between users based on their ratings (or check-ins) of the POIs.

This type of trust will be used to calculate the predicted rating using Equation (56).

Equation 56: Rating Prediction based on trust deduced from Ratings or Check-ins

$$P_{a,x} = \overline{r_a} + \frac{\sum_{b=1}^{N}(r_{b,x} - \overline{r_b}) * Trust^{U}(a \to b)}{\sum_{b=1}^{N} Trust^{U}(a \to b)}$$

Where:

- $P_{a,x}$: The predicted rating for user a on POI x.
- $\overline{r_a}$, $\overline{r_b}$: 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.
- $Trust^{U}(a \to b)$: The implicit trust derived from the ratings (or check-ins) made by user a towards user b.

V.3.2. Implicit Trust Propagation Computation

Sometimes, the rating and check-in matrices may contain several instances where two users have no common POIs, meaning that they have no co-rated POIs in the rating matrix and no shared check-ins in the check-in matrix. To address this issue, the calculation of trust propagation scores is necessary, as there are no direct trust relationships between users. Thus, from the direct trust network, it is possible to propagate trust and create new relationships between users who do not have a direct trust link between them (Hwang & Chen, 2007).

For instance, if user $a \in$ U (source user) places trust in user $b \in$ U (intermediate user) and user b in turn trusts user $c \in$ U (target user), we can infer, using the principle of propagation in the trust matrix, user a can trust user c by assigning a trust score, as described by Equation (57) below (Hwang & Chen, 2007):

Equation 57: Propagation of Implicit Trust in User trust Network

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

Where:

- $Ptrust_{a\rightarrow c}$: The indirect trust of user a in user c.
- a, b, and $c \in U$: The users from the set U (U: the set of users).
- $b \in$ adj(a) : a user from the set of neighbors of user a who trusts 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.

To reduce the sparsity problem in the TDMR matrix, which corresponds to the empty grey cells in Table 37, we used the trust propagation technique between users described in subsection II.6 and Equation (57) above, to infer new trust relationships, as shown in Figure 20 below.

Table 37: Non-Propagated Trust Matrix based on Rating

TDMR	$u1$	$u2$	$u3$	$u4$	$u5$
$u1$	**1**		0	1	0,50
$u2$		**1**	1	0,50	0
$u3$	0	1	**1**		0,33
$u4$	1	0,50		**1**	
$u5$	0,50	0	0,33		**1**

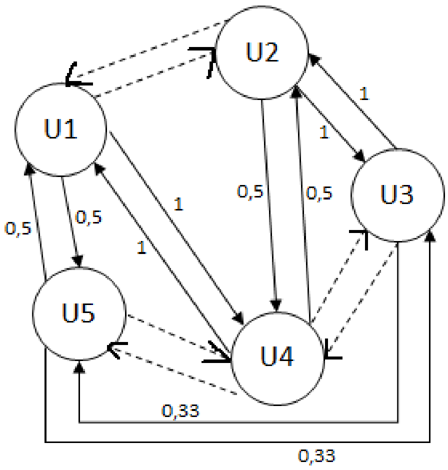

Figure 20: Trust graph derived from the TDMR matrix

To achieve this objective, we constructed the trust graph shown above (Figure 20), where the nodes represent users, the edges indicate trust relationships between these users, and the value above each edge represents an estimate of the trust level between the two connected users. Dashed edges correspond to inferred trust relationships, while solid edges represent trust relationships that have been directly computed.

Using Equation (57), we can calculate the trust score between users based on all possible paths connecting them.

In our case, we considered only paths that include a single intermediate. For example, to calculate $Trust^{U}(u1 \rightarrow u2)$, we consider only the path u1 → u4 → u2, which consists of two hops: u1 → u4 and u4 → u2. For this example, the trust score, denoted $Trust^{U}(u1 \rightarrow u2)$, is equal to 0.66 using Equation (57). In the same way, we infer the remaining trust scores and obtain the PTMR matrix (Propagated Trust Matrix based on Rating) shown in Table 38 below:

Table 38: PTMR (PropagatedTrustMatrix based on Rating)

PTMR	**U1**	**U2**	**U3**	**U4**	**U5**
U1	1	**2/3**	0/2	1/1	1/2
U2	**2/3**	1	1/1	1/2	0/1
U3	0/2	1/1	1	**2/3**	1/3
U4	1/1	1/2	**2/3**	1	**2/3**
U5	1/2	0/1	1/3	**2/3**	1

After inferring the trust scores in Table 37, we can calculate the rating prediction using Equation (56) as follows:

$$P_{u1,poi1} = \overline{r_{u1}} + \frac{(\mathrm{r}_{u2,poi1} - \overline{r_{u2}}) * Trust^{U}(u1 \rightarrow u2) + (\mathrm{r}_{u4,poi1} - \overline{r_{u4}}) * Trust^{U}(u1 \rightarrow u4)}{Trust^{U}(u1 \rightarrow u2) + Trust^{U}(u1 \rightarrow u4)}$$

$$P_{u1,poi1} = \frac{10}{3} + \frac{\left(4 - \frac{11}{3}\right) * \frac{2}{3} + \left(5 - \frac{10}{3}\right) * \frac{1}{1}}{\frac{2}{3} + \frac{1}{1}}$$

$$P_{u1,poi1} = 4.46$$

In the same way, we calculate all the values of the rating prediction matrix and can thus derive the RPMR matrix (Rating Prediction Matrix based on Rating) shown in Table 40 below (Table 39 represent the rating prediction User-POI matrix calculated based on trust before the propagation process).

Table 39: The RPMR matrix, representing the predicted ratings based on non-propagated trust Matrix for unvisited POIs

RPMR	POI1	POI2	POI3	POI4	POI5	POI6
U1	**5**	5	2	**3**	3	**2**
U2	4	**3**	**4**	3	**2.33**	4
U3	**2**	1	2	**1**	**$\overline{r_{u3}}$=5/3**	2
U4	5	**$\overline{r_{u4}}$=10/3**	**2**	3	2	**3.66**
U5	**$\overline{r_{u5}}$=13/3**	5	5	**$\overline{r_{u5}}$=13/3**	**4**	3

Table 40: The RPMR matrix, representing the predicted ratings based on propagated trust Matrix for unvisited POIs

RPMR	POI1	POI2	POI3	POI4	POI5	POI6
U1	**4.46**	5	2	**2.86**	3	**2.95**
U2	4	**4.11**	**3.33**	3	**2.90**	4
U3	**2.53**	1	2	**1.13**	**0.33**	2
U4	5	**4.04**	**3.04**	3	2	**3.06**
U5	**5**	5	5	**4**	**3.42**	3

V.4. The PRCT model

This section provides a detailed presentation of the POI recommendation approach proposed by the model named Propagation of Rating/Check-in for implicit Trust (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, denoted STR (Similarity Trust Rating), uses trust-based similarity derived from the ratings in the User-POI Rating Matrix (UPRM), while the second approach, denoted STC (Similarity Trust Check-in), uses trust-based similarity derived from the check-in matrix, denoted UPCM (User-POI Check-in Matrix). These two approaches are then used to calculate the user-user trust matrix, denoted TDM (Trust Derivation Matrix). The TDM matrix serves as the basis for applying the trust propagation principle to compute the RPM (Rating Prediction Matrix), which stores the predicted ratings that users would assign to POIs.

V.4.1. The Proposed Algorithms

After explaining how to calculate the trust between users, denoted TDM (Trust Derivation Matrix), based on their check-ins and POI ratings, and how to use propagation to enrich this matrix in order to avoid data sparsity issues, we propose in this subsection the algorithms to be used to implement these calculations.

The first algorithm, denoted UUTC (User-User Trust Computation), integrates the STR and STC approaches for calculating similarities based on ratings and check-ins. For this reason, the UUTC algorithm uses as input the rating matrix, denoted UPRM (User-POI Rating Matrix), of size m × n (where m indicates the number of users and n is the number of POIs) to calculate the TDMR (Trust Derivation Matrix based on Rating) of dimension m × m (where m indicates the number of users). This same algorithm also uses as input the check-in matrix, denoted UPCM (User-POI Check-in Matrix), of dimension m × n (where m is the number of users and n is the number of POIs) to calculate the TDMC (Trust Derivation Matrix based on Check-in) of dimension m × m.

After filling the two derived matrices TDMR and TDMC, this algorithm can calculate the prediction matrices of size m × n (where m is the number of users and n is the number of POIs), denoted respectively RPMR (Rating Prediction Matrix based on Rating) and RPMC (Rating Prediction Matrix based on Check-in), as indicated below (Algorithm 4):

Algorithm 4 : Trust between users based on ratings or check-ins

Rating Based	**Check-in Based**
INPUT:	**INPUT:**
UPRM: User-POI Rating Matrix	UPCM: User-POI Check-in Matrix
OUTPUT:	**OUTPUT:**
TDMR: Trust Derivation Matrix based on Rating	TDMC: Trust Derivation Matrix based on Check-in
RPMR: Rating Prediction Matrix based on Rating	RPMC: Rating Prediction Matrix based on Check-in
Var:	**Var:**
PR, distance, Correct ← empty lists	PC, distance, Correct ← empty lists
BEGIN	**BEGIN**
For each user *b* in UPRM Do	For each user *b* in UPCM Do
For each user *a* in UPRM AND *a* ≠ *b* Do	For each user *a* in UPCM AND *a* ≠ *b* Do
For each POI *i* in UPRM Do	For each POI *i* in UPCM Do
//Compute predict rating PR(*a*,*b*,*i*) using Eq(48)	//Compute predict check-in PC(*a*,*b*,*i*) using Eq(48)
PR(*a*,*b*,*i*) ← meanRate(*a*) + Rate(*b*,*i*) – meanRate(*b*)	PC(*a*,*b*,*i*) ← meanCheck(*a*) + Check(*b*,*i*) – meanCheck(*b*)
//Compute Correct(*a*,*b*,*i*) function using Eq(49)	//Compute Correct(*a*,*b*,*i*) function using Eq(49)
distance(*a*,*b*,*i*) ← | Rate(*a*, *i*) - PR(*a*,*b*,*i*) |	distance(*a*,*b*,*i*) ← | Check(*a*, *i*) - PC(*a*,*b*,*i*) |
IF (distance(*a*,*b*,*i*) < Ɛ) THEN	IF (distance(*a*,*b*,*i*) = = 0) THEN
Correct(*a*,*b*,*i*) ←1	Correct(*a*,*b*,*i*) ← 1
ELSE	ELSE
Correct(*a*,*b*,*i*) ←0	Correct(*a*,*b*,*i*) ← 0
END IF	END IF
//the set of user *b*'s recommendations using Eq(50)	//the set of user *b*'s recommendations using Eq(50)

RecSet(b)← ∑(Correct(a, b, i)) //the set of user b's correct recommendations using Eq(51)	RecSet(b) ← ∑(Correct(a, b, i)) //the set of user b's correct recommendations using Eq(51)
CorrectSet(b)←∑(Correct(a, b, i)) |Correct(a,b,i) = =1)	CorrectSet(b)← ∑(Correct(a, b, i)) |Correct(a,b,i) = =1)
END FOR	END FOR
END FOR	END FOR
//Compute user-user trust TDMR(a,b) using Eq(54)	//Compute user-user trust TDMC(a,b) using Eq(54)
TDMR(a,b) ← CorrectSet(b) / RecSet(b);	TDMC(a,b) ← CorrectSet(b) / RecSet(b);
END FOR	END FOR
//Compute Rating Prediction (RPMR) based on rating trust (TDMR) using Eq(56)	//Compute Rating Prediction (RPMR) based on check-in trust (TDMC) using Eq(56)
For each user a in UPRM Do	For each user a in UPRM Do
For each POI x in UPRM Do	For each POI x in UPRM Do
IF (UPRM(a, x) = = empty) THEN	IF (UPRM(a, x) = = empty) THEN
numerator ← 0	numerator ← 0
denominator ← 0	denominator ← 0
For user b in UPRM Do	For user b in UPRM Do
IF $b \neq a$ AND isNeighbor(a, b) THEN //b is a neighbor of a	IF $b \neq a$ AND isNeighbor(a, b) THEN //b is a neighbor of a
numerator ← numerator + (Rate(b, x) - meanRate(b)) * TDMR(a, b)	numerator = numerator + (Rate(b, x) - meanRate(b))* TDMC(a, b);
denominator ← denominator + TDMR(a, b)	denominator = denominator + TDMC(a, b)
END IF	END IF
END FOR	END FOR
IF denominator ≠ 0 THEN	IF denominator ≠ 0 THEN
RPMR(a,x) ← meanRate(a) + (numerator / denominator)	RPMC(a,x) ← meanRate(a) + $\frac{\text{numerator}}{\text{denominator}}$
ELSE	ELSE
RPMR(a,x) ← meanRate(a)	RPMC(a,x) ← meanRate(a)
END IF	END IF
END IF	END IF
END FOR	END FOR
END FOR	END FOR
END	**END**

To mitigate the issues related to data sparsity, the PRCT model can use the UUTP (User-User Trust Propagation) algorithm to enrich the TDMR and TDMC matrices using the propagation principle. This algorithm allows the densification of the TDMR and TDMC matrices by leveraging indirect trust links between users to obtain the PTMR (Propagated Trust Matrix based on Rating) of dimension m × m (where m is the number of users) and PTMC (Propagated Trust Matrix based on Check-in) of dimension m × m (where m is the number of users), as indicated below (Algorithm 5):

V.4.2. Operation of the PRCT model

Figure 21 below illustrates the operation of the PRCT model, which is based on four main steps, summarized as follows:

Algorithm 5 : Propagation of user-user trust

	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 Eq(57)**	
	For each user a in TDMR Do	For each user a in TDMC Do
	For each user c in TDMR Do	For each user c in TDMC Do
	IF (TDMR(a, c) = = empty) THEN	IF (TDMC(a, c) = = empty) THEN
	//Search for all possible intermediate users *b*	
	For each user b in TDMR Do	For each user b in TDMC Do
	IF (TDMR(a,b) AND TDMR(b,c)) THEN	IF (TDMC(a,b) AND TDMC(b,c)) THEN
	$PTMR(a,c) \leftarrow (commonRate(a,b) * TDMR(a,b) + commonRate(b,c) * TDMR(b,c))/(commonRate(a,b) + commonRate(b,c))$	$PTMC(a,c) \leftarrow (commonCheck(a,b) * TDMC(a,b) + commonCheck(b,c) * TDMC(b,c))/(commonCheck(a,b) + commonCheck(b,c))$
	END IF	END IF
	END FOR	END FOR
	//If there are multiple intermediate users *b*, perform aggregation	
	IF (length(PTMR(a,c)) > 1) THEN	IF (length(PTMC(a,c)) > 1) THEN
	PTMR(a,c) ← mean(PTMR(a,c))	PTMC(a,c) ← mean(PTMC(a,c))
	END IF	END IF
	END FOR	END FOR
	END FOR	END FOR
END		

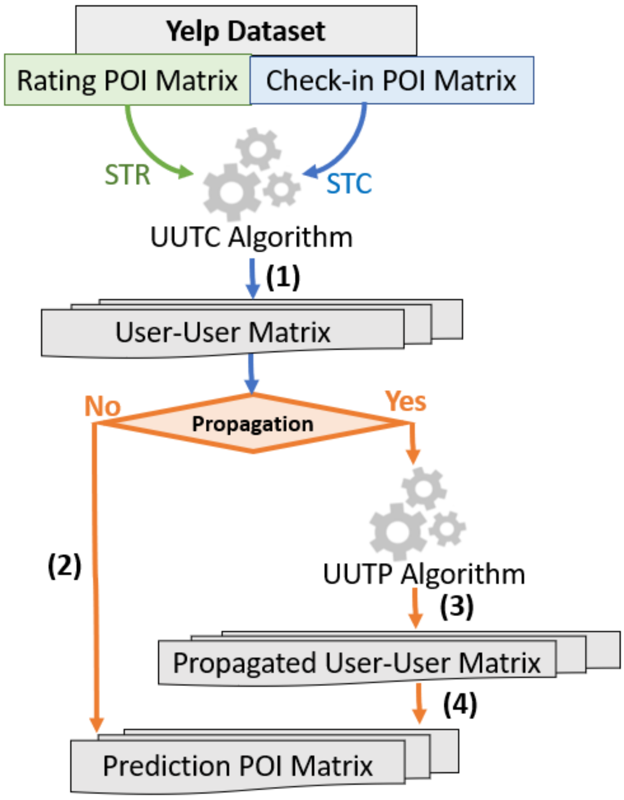

Figure 21: Functional Description of the PRCT Model

1. After filtering the review.js file from the Yelp dataset[1] (X. Wang et al., 2021) to extract only the ratings and check-ins, the UUTC algorithm calculates the trust between users based on (1) the ratings, referred to as STR (Similarity Trust Rating), and (2) the trust derived from the check-ins, denoted STC (Similarity Trust Check-in).

2. The PRCT model can ignore the propagation principle and proceed directly to the prediction calculations. This process is performed without using the UUTP algorithm.

3. The PRCT model can adopt the propagation principle (two hops) to further enrich the content of the similarity (trust) matrices between users. These matrices are calculated using the UUTP algorithm.

4. Calculate predictions from the similarity (trust) matrices between users derived using the propagation principle (two hops).

V.4.3. Evaluation du modèle PRCT

In Figure 22 below, we explain how to evaluate the PRCT model by comparing its two approaches (STR and STC) with other types of approaches from the literature, such as: (1) Pearson similarity, (2) Jaccard similarity, (3) Cosine similarity, and (4) trust-based similarity defined by O'Donovan (O'Donovan & Smyth, 2005).

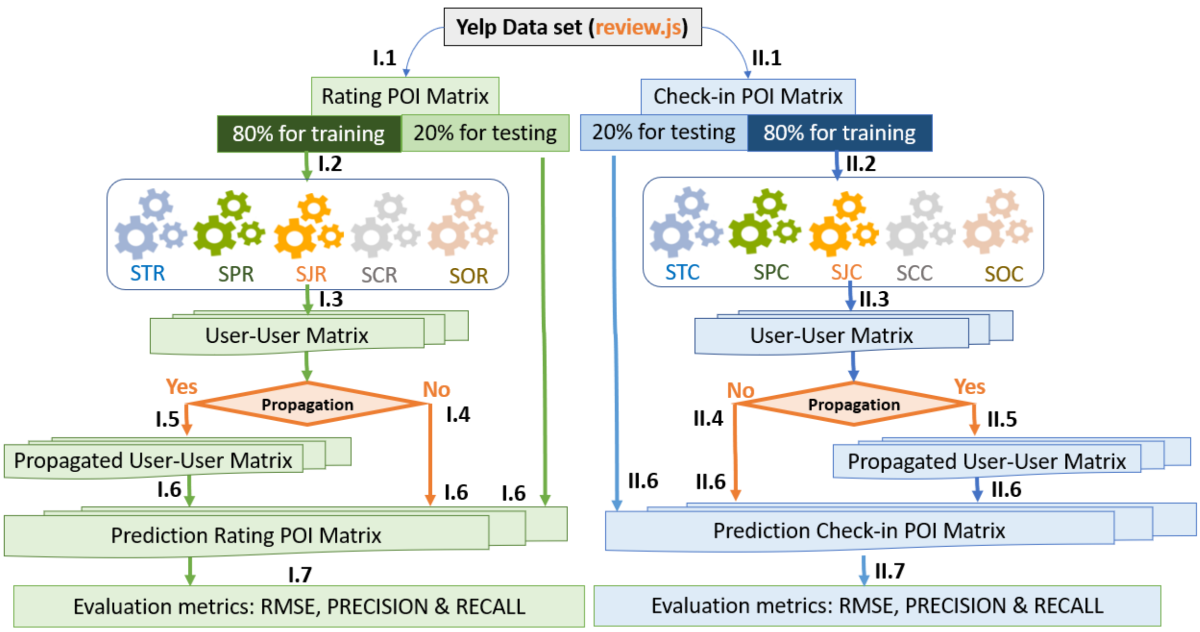

Figure 22: Description of the PRCT Model Evaluation Process

[1] https://www.yelp.com/dataset

Figure 22 above consists of two parts: the first part deals with ratings and is composed of 7 main steps (from I.1 to I.7 in Figure 22), and the second part focuses on check-ins data, also consisting of 7 main steps (from II.1 to II.7 in Figure 22). Each step in this model is described based on the data it processes and the algorithms it uses. 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 SPR (Similarity Pearson Rating), (3) Jaccard similarity from ratings, noted SJR (Similarity Jaccard Rating), (4) Cosine similarity from ratings, noted SCR (Similarity Cosine Rating), and (5) trust-based similarity defined by O'Donovan from ratings, noted SOR (Similarity O'Donovan Rating).

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.

Partie 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 Points of Interest (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 consisting of user check-ins on 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 SPC (Similarity Pearson Check-in), (3) Jaccard similarity from check-ins, denoted as SJC (Similarity Jaccard Check-in), (4) Cosine similarity from check-ins, denoted as SCC (Similarity Cosine Check-in), and (5) the trust-based similarity defined by O'Donovan, which we have adapted to the context of check-ins, denoted as SOC (Similarity O'Donovan Check-in).

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.

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 41 and a series of hyperparameters defined in Table 42. 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 the next subsections.

Table 41: Description of dataset 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; ***0**: visited but did **not** like it;* ***1**: visited and **liked***

Table 42: List of the HRCT 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

Finally, to evaluate the performance of the PRCT model, the dataset and hyperparameters described above are used with the metrics RMSE, PRECISION, and RECALL. Initially, the RMSE metric allows for comparing the accuracy of the ratings predicted by the two variants of the PRCT model (STR and STC) with other existing POI recommendation approaches in the literature. In the second step, the PRECISION and RECALL metrics are used to gain insight into the quality of the recommendations provided by these two variants of our PRCT model. Additionally, the F1-score can be calculated using Equation (58) to provide a balanced measure of PRECISION and RECALL.

Equation 58: F1 Score, Harmonic Mean of Precision and Recall

$$\mathrm{F1} = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}$$

V.5. Results and Discussions

In this section, a comparison between the variants of the PRCT model is conducted using the Yelp dataset, which is divided into two parts: 80% for training and 20% for testing. The STR, STC, SOR, and SOC algorithms use the training portion to build the trust matrices of the PRCT model and predict POI ratings in the test portion. The performance of this model is evaluated using RMSE and PRECISION/RECALL metrics to assess the quality of POI recommendations. The PRCT model is then compared with other POI recommendation models that use different similarity measures such as Jaccard, Cosine, and Pearson. Finally, a study on the impact of trust propagation on sparsity and recommendation quality is carried out on the trust matrices of the PRCT model and the prediction matrices.

V.5.1. Comparison between 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 (see Table 43).

Table 43: 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

In Table 43, we observe that the rating-based prediction algorithm, called STR, outperforms the STC, STR with propagation, and STC with propagation algorithms in terms of PRECISION and RECALL. Moreover, it is noteworthy that the STC and STR approaches (with or without propagation) of the PRCT model offer better performance than traditional O'Donovan approach, which also uses ratings and check-ins, in terms of PRECISION and RECALL.

V.5.2. The PRCT model without propagation

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

V.5.2.1. Comparison between STR and the approaches SPR, SCR, SJR, and SOR

In Table 44 below, we observe that the SPR (Similarity Pearson Rating) approach outperforms all rating-based approaches in terms of RMSE and RECALL. However, the STR approach of the PRCT model appears to perform in terms of PRECISION.

Table 44: 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

V.5.2.2. Comparison between STC and the approaches SPC, SCC, SJC, and SOC

In Table 45 below, we observe that the SPC (Similarity Pearson Check-in) approach outperforms all other check-in-based approaches in terms of RMSE and RECALL. However, the SCC (Similarity Cosine Check-in) approach appears to perform better in terms of PRECISION.

Table 45: 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

V.5.2.3. Summary of the Comparisons

In Table 46 below, 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.

Table 46: 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

V.5.3. The PRCT Model with Propagation

In this subsection, we studied the impact of propagation on the PRCT model and on the other approaches: SPR, SCR, SJR, SOR, SPC, SCC, SJC, and SOC. Then, we conducted a comparison between these approaches in terms of RMSE, PRECISION, and RECALL. Finally, we summarized the results obtained from these comparisons.

V.5.3.1. Study of the impact of propagation on sparsity

In Table 47 below, we observe that the approaches "SCR with propagation" and "SJR with propagation" show the highest density rates for both the similarity matrices and the prediction matrices.

Table 47: Effect of propagation on the sparsity of similarity and prediction matrices

Type of Similarity	AVG Similarity 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

V.5.3.2. Comparison of Approaches with or without Propagation

In Table 48 below, we observe that, in general, approaches involving propagation lead to a decrease in the performance of the recommendation systems, particularly 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, and the SJC approach remains stable in terms of RECALL.

Table 48: 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**

V.5.3.3. Summary of comparisons

Table 49: Comparison between Rating-based approaches and Check-in-based approaches

Type of Similarity	**AVG RMSE**	**AVG PRECISION**	**AVG RECALL**	**AVG Similarity 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**

In Table 49, we observe that propagation in check-in-based approaches provides better performance than rating-based ones in terms of RMSE, PRECISION, and RECALL. However, propagation in rating-based approaches yields better results than check-in-based ones when it comes to the sparsity of similarity and prediction matrices.

V.5.4. Overall Summary and Discussion of the Results

The results obtained from the Yelp dataset show that the rating-based approach (STR) of the PRCT model generally outperforms other variants in terms of PRECISION and RECALL. However, the PRCT model approaches based on check-ins (STC) prove to be particularly effective in terms of RMSE, offering greater robustness in the POI recommendation process. Furthermore, although trust propagation improves the density of prediction matrices, it does not always lead to significant improvements in prediction performance, except in specific cases where it is well controlled, such as with the SCC and SJC approaches.

In summary, check-in-based approaches offer superior performance in terms of RMSE, PRECISION, and RECALL compared to rating-based approaches. However, while propagation improves matrix sparsity, it does not always lead to notable enhancements in PRECISION and RECALL. Approaches such as SCC and SJC stand out for their robustness and stability in the face of propagation, making them particularly valuable in certain usage contexts. Still, it is important to properly adjust these propagation methods and choose the best-suited approaches to improve the quality of recommendations.

Conclusion

In this chapter, we proposed and studied the PRCT model to improve POI recommendation by leveraging implicit trust between users, inferred from their interactions with POIs through ratings and check-ins. The main objective was to explore the impact of trust propagation on the quality of recommendations, particularly in cases where similarity (trust) matrices are sparse. Overall, this chapter demonstrates the effectiveness of trust propagation in enriching trust matrices, especially in contexts where data is insufficient or incomplete.

Chapter VI:

The ITCRC Model Integrating Trust in the Prediction Calculation

VI.1. Introduction

In this chapter, we introduce a hybrid model named ITCRC (Implicit Trust based on Combining point of interest Ratings and user Check-ins), designed to mitigate the cold-start problems that affect trust-based collaborative filtering techniques. This model leverages both POI ratings and user check-ins to estimate implicit trust, which is a key factor in improving location recommendations in a Location-Based Social Network (LBSN).

VI.2. Problem Definition

Recommender systems (RSs) are designed to analyze users' past preferences and generate reliable predictions about their future choices. Among the most widely used methods, collaborative filtering (CF) relies on analyzing interactions between users to compute similarities and recommend points of interest (POIs). However, this approach faces several major challenges: data sparsity, the cold-start problem, and scalability issues.

To address these limitations, integrating the notion of trust into POI recommendations has proven to be a promising alternative. Trust can be either explicit, when users directly declare their level of trust in others, or implicit, when it is inferred from usage behaviors such as POI ratings and check-ins. Implicit trust, derived from recorded interactions, represents a relevant approach to alleviating issues related to data sparsity and the cold-start problem.

In the context of POI recommendations, LBSNs play a crucial role by capturing spatiotemporal data from users' shared experiences in the form of check-ins and ratings. This data, rich in information, helps refine the understanding of user preferences and mitigate the challenges faced by traditional recommender systems.

In response to these challenges, we propose a hybrid solution that combines ratings and check-ins to accurately estimate implicit trust. This approach aims to optimize the quality of recommendations by fully leveraging users' social relationships and behavioral interactions, thereby enhancing the user experience and the relevance of the suggestions.

VI.3. Problem Formulation

This section presents an approach for predicting users' future ratings of points of interest (POIs) by leveraging implicit trust calculation equations derived from their past interactions (ratings and check-ins). Although the equations were detailed in Sections IV and V, we introduce here new equations specifically designed to estimate POI ratings, relies solely on trust scores between users, without considering their similarities. To calculate trust from POI ratings and user check-ins, we used the same principle mentioned in section IV and section V.

VI.3.1. Trust based on POI Rating

In this approach, trust between two users is estimated based on the accuracy of a rating prediction made by one user for another. Specifically, user *b* predicts a rating on behalf of user *a*, and the trust from user *a* towards user *b* is inferred by evaluating the accuracy of this prediction, as detailed in the following steps.

1. To compute trust based on ratings between two users, we rely on the accuracy of a prediction made by a single user. Specifically, we calculate the rating prediction from user *b* to user *a*, as shown in Equation (59) below. This approach allows us to assess how closely the prediction aligns with the actual rating, thereby reflecting the level of trust in user *b*'s ability to predict user *a*'s preferences.

Equation 59: Rating Prediction Based on a Single User in a Rating Context

$$PR_{a,i}^{b} = \overline{Rate_a} + (Rate_{b,i} - \overline{Rate_b})$$

Where :

- $PR_{a,i}^{b}$: The predicted rating for user a on POI i based on user b.
- $\overline{Rat_a}$: The average ratings of user a for all POIs.
- $Rat_{b,i}$: The actual rating given to POI i by user b.

2. After computing the predicted rating from user *b* on behalf of user *a*, the next step is to evaluate its accuracy by comparing it with the actual rating provided by user *a*. This is done by calculating the distance between the predicted and real rating values, as described in Equation (60). The resulting difference is then compared to a predefined threshold to determine whether the prediction is sufficiently accurate to establish trust.

Equation 60: Determining Prediction Correctness between User b and User a for POI i, in a Rating Context

$$Correct(i, b, a) \Leftrightarrow |PR_{a,i}^{b} - Rate_{a,i}| < \varepsilon$$

3. Identifying the set of recommendations generated by user *b* when acting as a recommender, as shown in Equation (61) below.

Equation 61: Recommendation Set of User b in a Rating Context

$$RecSet(b) = \{(PR^b_{1,1}\ , Rate_{1,1}), \dots, (PR^b_{m,n}\ , Rate_{m,n})\}$$

Where:

- $P^b_{j,k}$: represents the prediction made by recommender b for the rating that a user j (where j ranges from 1 to m) will give to an POI k (where k ranges from 1 to n).
- $RC_{j,k}$: represents the actual rating of POI k (where k ranges from 1 to n) given by user j (where j ranges from 1 to m).

4. Defines the set of correct recommendations made by recommender b, as shown in Equation (62) below:

Equation 62: Set of Correct Recommendations by Recommender b in a Rating Context

$$CorrectSet(b) = \{(PR^b_{j,k}\ , Rate_{j,k}) \epsilon RecSet(b) : Correct(k, b, PR^b_{j,k})\}$$

5. User-user trust: the set of correct recommendations derived from those made by recommender b when user b was involved in generating recommendations specifically for user a, as shown in Equation (63) below:

Equation 63: Trust between users based on Rating

$$Trust^U(a \to b) = \frac{card\{(PR^b_{j,k}\ , Rate_{j,k}) \in CorrectSet(b) : j = a\}}{card\{(PR^b_{j,k}\ , Rate_{j,k}) \in RecSet(b) : j = a\}}$$

VI.3.2. Trust based on user Check-in

1. User b predicts a check-in at POI i on behalf of user a, as shown in Equation (64) below:

Equation 64: Check-in Prediction Based on a Single User in a Check-in Context

$$Ch^b_{a,i} = \overline{Ch_a} + (Ch_{b,i} - \overline{Ch_b})$$

Where :

- $Ch^b_{a,i}$: The predicted check-in of POI i for user a based on user b.
- $Ch_{b,i} \in \{0,1\}$: The check-in of POI i by user b.
- $\overline{Ch_a}$: The average check-ins of user a.
- $\overline{Ch_b}$: The average check-ins of user b.

2. Compare the predicted check-in by user b with the actual check-in of user a. If they are equal, that is, if user b's prediction matches user a's actual check-in, then the prediction is considered correct, as shown in Equation (65) below:

Equation 65: Determining Prediction Correctness between User b and User a for POI i, in a Check-in Context

$$Correct\_Ch(i, b, a) \Leftrightarrow \left|Ch_{a,i}^{b} - Ch_{a,i}\right| = 0$$

3. The set of all recommendations in which user b is involved, as shown in Equation (66) below:

Equation 66: Recommendation Set of User b in a Check-in Context

$$RecSet\_Ch(b) = \left\{\left(Ch_{1,1}^{b}, Ch_{1,1}\right), \dots, \left(Ch_{m,n}^{b}, Ch_{m,n}\right)\right\}$$

Where:

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

4. The set of all correct recommendations in which user b was involved and made accurate predictions, as shown in Equation (67) below:

Equation 67: Set of Correct Recommendations by Recommender b in a Check-in Context

$$CorrectSet\_Ch(b) = \left\{\left(Ch_{j,k}^{b}, Ch_{j,k}\right) \epsilon\ RecSet\_Ch(b) : Correct\_Ch(k, b, Ch_{j,k}^{b})\right\}$$

5. User-user trust, as shown in Equation (68) below: the set of correct recommendations derived from those made by recommender b when user b was involved in generating recommendations specifically for user a, based on check-in data.

Equation 68: Trust between users based on Check-in

$$Trust\_Ch^{U}(a \rightarrow b) = \frac{card\left\{\left(Ch_{j,k}^{b}, Ch_{j,k}\right) \in CorrectSet\_Ch(b) : j = a\right\}}{card\left\{\left(Ch_{j,k}^{b}, Ch_{j,k}\right) \in RecSet\_Ch(b) : j = a\right\}}$$

After we calculate the trust between users, based on ratings and check-ins, we can calculate the rating prediction, as shown in Equation (69) below:

Equation 69: Rating Prediction based on Rating-trust or Check-in-trust

$$PR_{a,x} = \overline{Rat_a} + \frac{\sum_{b=1}^{N}(Rat_{b,x} - \overline{Rat_b}) * w(a,b)}{\sum_{b=1}^{N} w(a,b)}$$

$$\text{Where: } w(a,b) = \begin{cases} Trust\_R^U(a \rightarrow b) \\ Trust\_Ch^U\ (a \rightarrow b) \end{cases}$$

VI.4. The ITCRC Model

This section explains the algorithms necessary to predict users' future ratings using our recommendation model named ITCRC.

VI.4.1. The Proposed Algorithms

After explaining how to calculate trust between users, we can derive the values of the TDMR matrix (Trust Derivation Matrix based on Rating) and the TDMC matrix (Trust Derivation Matrix based on Check-in) from the UPRM (User-POI Rating Matrix) and UPCM (User-POI Check-in Matrix). Then, we can use the TDMR and TDMC matrices to compute the prediction matrices, respectively denoted RPM1 (Rating Prediction Matrix) based on TDMR and RPM2 (Rating Prediction Matrix) based on TDMC. These computations are carried out using Algorithm 6 and Algorithm 7 below.

Algorithm 6 : Trust Derivation Matrix based on Rating "TDMR"

INPUT: UPRM: User-POI Rating Matrix
OUTPUT: TDMR: Trust Derivation Matrix based on Rating
RPM1: Rating Prediction Matrix based on TDMR
Var RatePre, DisError, Correct, RecSet, CorrectSet: empty Lists
BEGIN
//Compute TDMR
For each user "*a*" and user "*b*" and POI "*i*"
Step 1: Compute rating prediction RatePre(*a*,*b*,*i*) using Eq(59)
Step 2: Compute distance error DisError(*a*,*b*,*i*) and binary success/fail score Correct(*a*,*b*,*i*) using Eq(60)
IF DisError(*a*,*b*,*i*) < ε THEN Correct(*a*,*b*,*i*) ← 1 (success)
ELSE Correct(*a*,*b*,*i*) ← 0 (Fail)
Step 3: RecSet(*b*) the set of recommendations of user *b* using Eq(61)
Step 4: CorrectSet(*b*) the set of correct recommendations of user *b* using Eq(62)
Step 5: Compute user-user trust TDMR(*a*,*b*) using Eq(63)
//Compute RPM1 using TDMR and Eq(69)
For each user "*a*" and POI "*x*"
Step 6: Compute Rating Prediction RPM1 based on rating trust (TDMR)
END

Algorithm 7 : Trust Derivation Matrix based on Check-in "TDMC"

INPUT: UPCM: User-POI Check-in Matrix
UPRM: User-POI Rating Matrix
OUTPUT: TDMC: Trust Derivation Matrix based on Check-in
RPM2: Rating Prediction Matrix based on TDMC
Var CheckPre, DisError, CorrectCH, RecSetCH, CorrectSetCH: empty Lists
BEGIN
//Compute TDMC
For each user "*a*" and user "*b*" and POI "*i*"
Step 1: Compute check-in prediction CheckPre(*a*,*b*,*i*) using Eq(64)
Step 2: Compute distance error DisError(*a*,*b*,*i*) and binary success/fail score CorrectCH(*a*,*b*,*i*) using Eq(65)
IF DisError(*a*,*b*,*i*) = = 0 THEN CorrectCH(*a*,*b*,*i*) ← 1 (Success)
ELSE CorrectCH(*a*,*b*,*i*) ← 0 (Fail)
Step 3: RecSetCH(*b*) the set of recommendations of user *b* using Eq(66)
Step 4: CorrectSetCH(*b*) the set of correct recommendations of user *b* using Eq(67)
Step 5: Compute user-user trust TDMC(*a*,*b*) using Eq(68)
//Compute RPM2 using TDMC and Eq(69)
For each user "*a*" and POI "*x*"
Step 6: Compute Rating Prediction RPM2 based on check-in trust (TDMC)
END

VI.4.2. Functioning of the ITCRC Model

This section presents the POI recommendation approach of our model named Implicit Trust based on Combining point of interest Ratings and user Check-ins (ITCRC). It is based, on the one hand, on the inclusion of POI ratings and user check-ins extracted from the Yelp dataset (see arrows a.0 and b.0 in Figure 23), and on the other hand, on Algorithm 6 and Algorithm 7 proposed in subsection VI.4.1.

These two algorithms use the trust matrices TDMR and TDMC (see arrows a.1 and b.1 in Figure 23) to compute the matrices RPM1 and RPM2, which contain the predicted ratings of POIs (see arrows a.2 and b.2 in Figure 23).

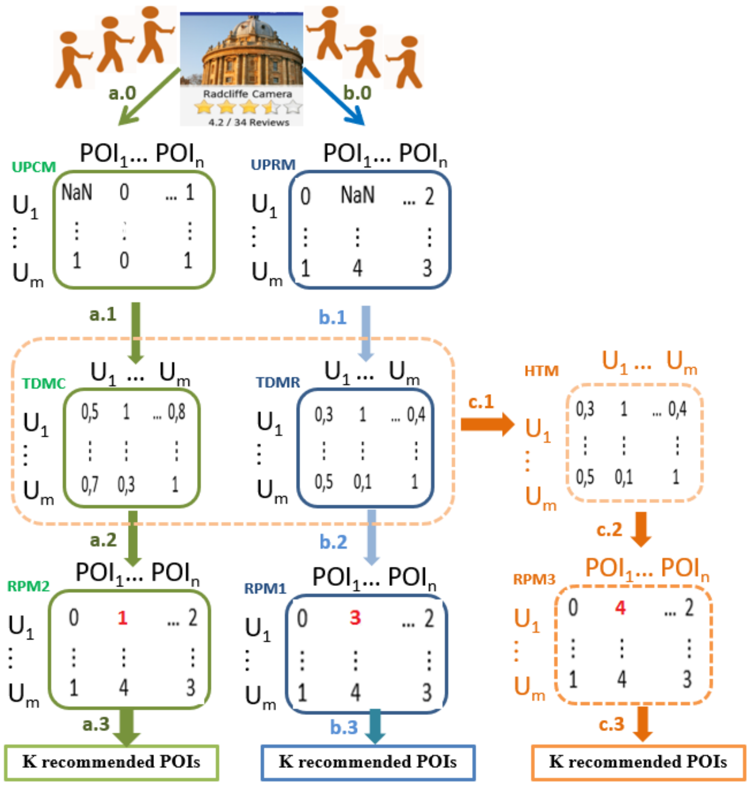

Figure 23: Functional Description of the ITCRC Model

Then, the two trust matrices TDMR and TDMC, obtained from Algorithm 6 and Algorithm 7, can be combined (see arrow c.1 in Figure 23) using Algorithm 8 below to generate the HTM (Hybrid Trust Matrix) of dimension m × m (m: number of users). This matrix can then be used to calculate the predicted POI ratings (see arrow c.2 in Figure 23) in the RPM3 (Rating Prediction Matrix) based on HTM of dimension m × n (m: number of users, n: number of POIs) using Equation (70) and Algorithm 8 below.

Equation 70: Rating prediction based on Hybrid Trust Matrix (HTM)

$$PR_{a,x} = \overline{Rat_a} + \frac{\sum_{b=1}^{N}(Rat_{b,x} - \overline{Rat_b}) * Trust\_H^U(a \to b)}{\sum_{b=1}^{N} Trust\_H^U(a \to b)}$$

Where:

- $PR_{a,x}$: The predicted rating for user *a* on POI *x*.
- $\overline{Rat_a}$: The average rating of user *a* for all POIs.
- $Rat_{b,x}$: The actual rating given to POI *x* by user *b*.
- $Trust\_H^U(a \to b)$: The trust based on the combination of ratings and check-ins is used to predict user preferences.

Algorithm 8 : Hybrid Trust Matrix "HTM"

INPUT: TDMR: Trust Derivation Matrix based on Rating
TDMC: Trust Derivation Matrix based on Check-in
UPRM: User-POI Rating Matrix
OUTPUT: HTM: Hybrid Trust Matrix
RPM3: Rating Prediction Matrix based on HTM
BEGIN
//Compute HTM
For each user "a" and user "b" and POI "i"
IF TDMR(a,b) exist and TDMC(a,b) exist THEN
$HTM(a,b) \leftarrow \frac{2*TDMR(a,b)*TDMC(a,b)}{TDMR(a,b)+TDMC(a,b)}$
ELSE IF TDMR(a,b) ! exist and TDMC(a,b) exist THEN
$HTM(a,b) \leftarrow TDMC(a,b)$
ELSE IF TDMR(a,b) exist and TDMC(a,b) ! exist THEN
$HTM(a,b) \leftarrow TDMR(a,b)$
ELSE
$HTM(a,b) \leftarrow 0$
//Compute RPM3 using HTM and Eq(70)
For each user "a" and POI "x"
$RPM3(a,x) \leftarrow \overline{Rat_a} + \frac{\sum_{b=1}^{N}(Rat_{b,x} - \overline{Rat_b}) * HTM(a,b)}{\sum_{b=1}^{N} HTM(a,b)}$
END

Finally, these three rating prediction matrices: RPM1, RPM2, and RPM3 indicated in Figure 23 can be compared to other approaches in the literature using evaluation metrics such as RMSE, PRECISION, and RECALL (see arrows a.3, b.3, and c.3 in Figure 23).

VI.4.3. Evaluation of the ITCRC model

To evaluate the performance of the ITCRC model, we used the Yelp dataset, as it contains both POI ratings and user check-ins. This dataset was then divided into two parts: (1) 80% for model training and (2) 20% for testing. Next, we defined ε = 0.9 as the trust threshold. Finally, we adopted standard evaluation metrics such as RMSE, PRECISION, and RECALL to assess the quality of the generated recommendations, along with additional metrics to estimate data sparsity.

VI.5. Results and Disscusions

VI.5.1. Comparison of ITCRC Model Variants

In this subsection, we compare the three variants of the ITCRC model: (1) the approach based on trust derived from ratings, denoted as TR, (2) the approach based on trust derived from check-ins, denoted as TC, and (3) the approach based on trust derived from the combination (hybridization) of ratings and check-ins, denoted as TH. This comparison is based on the RMSE and F1 score values, taking into account the variation in the number of users (see Figure 24 and Figure 25).

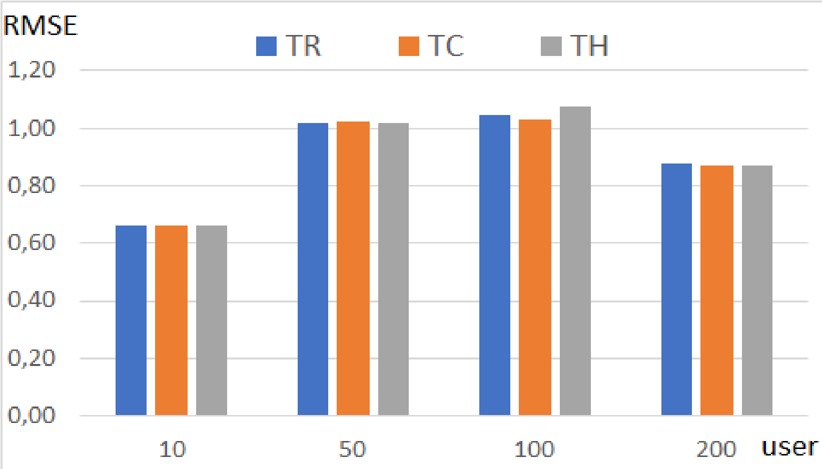

Figure 24: Comparison of TR, TC, and TH Approaches in Terms of RMSE

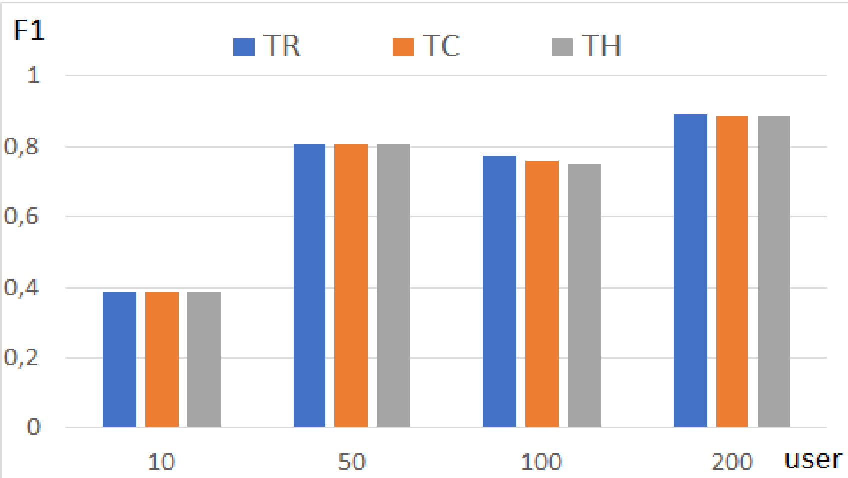

Figure 25: Comparison of TR, TC, and TH Approaches in Terms of F1

Moreover, in Table 50 below, we observe that the TR approach outperforms the TC and TH approaches in terms of average RMSE, PRECISION, and RECALL, denoted respectively as AVG RMSE, AVG PRECISION, and AVG RECALL.

Table 50: Comparison of the TR, TC, and TH approaches using AVG RMSE, AVG PRECISION, and AVG RECALL

	TR	**TC**	**TH**
AVG RMSE	**0,9440**	0,9465	0,9484
AVG PRECISION	**0,9154**	0,9133	0,9119
AVG RECALL	**0,6537**	0,6504	0,6497

VI.5.2. The ITCRC Model and Sparsity

To address the sparsity issue that can arise during POI recommendations, we selected the Yelp dataset because it contains both user check-ins information and ratings for visited POIs. Moreover, we maintained the same density level for the trust matrices of the "TR," "TC," and "TH" approaches in order to focus solely on the sparsity rates of the prediction matrices.

For these reasons, we were able to demonstrate that the hybrid approach (TH) reduces the sparsity of these prediction matrices by 36.08% compared to the other approaches, TR and TC, as shown in Figure 26 and Table 51 below.

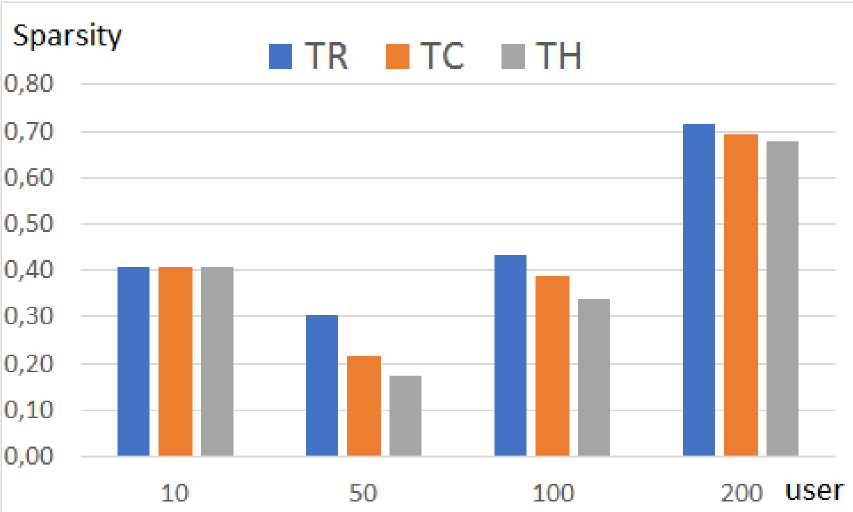

Figure 26: Comparison of the sparsity of the prediction matrices of the TR, TC, and TH approaches based on the number of users

Table 51: Comparison of the density of the prediction and trust matrices of the TR, TC, and TH approaches

	TR	TC	TH
AVG Trust Matrix Sparsity	0,388	0,388	0,388
AVG Prediction Matrix Sparsity	0,361	0,304	**0,265**

VI.5.3. Comparison of the variants of the O'Donovan model

In this subsection, we compare the different variants of the model based on the O'Donovan formula for calculating trust. This model consists of three types of approaches. The first approach, denoted OR (O'Donovan trust based on Ratings) see Algorithm 9, is based on ratings to calculate the matrix containing the trust value of each user (profile-level) (O'Donovan & Smyth, 2005). The second approach, denoted OC (O'Donovan trust based on Check-ins), is an adaptation of the OR approach for the check-in context, as outlined in Algorithm 10 below.

Algorithm 9 : Profile-level Trust O'donovan's approach adapted to LBSN (Rating case)

INPUT: UPRM: User-POI Rating Matrix
OUTPUT: R-TProfile: Trust Profile Matrix based on Rating
RPM4: Rating Prediction Matrix based on R-TProfile
Var RatePre, DistError, Correct, RecSet, CorrectSet: empty Lists
BEGIN
//Compute R-TProfile
For each user "*a*" and user "*b*" and POI "*i*"
Step 1: Compute rating prediction RatePre(*a*,*b*,*i*) using Eq(59)
Step 2: Compute distance error DistError(*a*,*b*,*i*) and binary success/fail score Correct(*a*,*b*,*i*) using Eq(60)
IF DistError(*a*,*b*,*i*) < ε THEN Correct(*a*,*b*,*i*) ← 1 (Success)
ELSE Correct(*a*,*b*,*i*) ← 0 (Fail)
Step 3: RecSet(*b*) the set of recommendations for user *b* using Eq(61)
Step 4: CorrectSet(*b*) the set of correct recommendations for user *b* using Eq(62)
Step 5: Compute Profile-Level Trust R-TProfile(*b*) using Eq(52)
//Compute RPM4 using R-TProfile and Eq(69)
For each user "*a*" and POI "*x*"
Step 6: Compute Rating Prediction RPM4 based on R-TProfile
END

Algorithm 10 : Profile-Level Trust O'Donovan's adapted to LBSN (Check-in case)

INPUT: UPCM: User-POI Check-in Matrix
UPRM: User-POI Rating Matrix
OUTPUT: C-TProfile: Trust Profile Matrix based on Check-in
RPM5: Rating Prediction Matrix based on C-TProfile
Var CheckPre, DistError, RecSet, CorrectSet: empty Lists
BEGIN
//Compute C-TProfile
For each user "*a*" and user "*b*" and POI "*i*"
Step 1: Compute check-in prediction CheckPre(*a*,*b*,*i*) using Eq(64)
Step 2: Compute distance error DistError(*a*,*b*,*i*) and binary success/fail score Correct(*a*,*b*,*i*) using Eq(65)
IF DistError(*a*,*b*,*i*) = = 0 THEN Correct(*a*,*b*,*i*) ← 1 (Success)
ELSE Correct(*a*,*b*,*i*) ← 0 (Fail)
Step 3: RecSet(*b*) the set of recommendations for user *b* using Eq(66)
Step 4: CorrectSet(*b*) the set of correct recommendations for user *b* using Eq(67)
Step 5: Compute Profile-Level Trust C-TProfile(*b*) using Eq(52) adapted for check-in
//Compute RPM5 using C-TProfile and Eq(69)
For each user "*a*" and POI "*x*"
Step 6: Compute Rating Prediction RPM5 based on C-TProfile
END

The third approach, denoted OH (O'Donovan trust based on Hybridization of ratings and check-ins), is a hybrid approach that combines the first (OR) and the second (OC) approaches. To compare these three approaches, we analyzed the values of their RMSE, PRECISION, and RECALL based on the evolution of the number of users (see Figure 27).

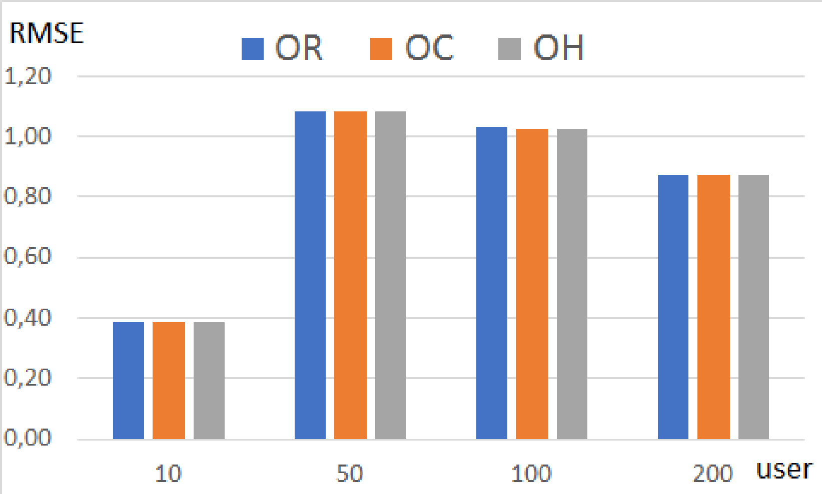

Figure 27: Comparison of the OR, OC, and OH approaches using RMSE

From Table 52 below and Figure 27, we observe that the classic O'Donovan approach based on ratings adapted to the context of LBSNs (OR) outperforms the other approaches OC and OH in terms of RMSE.

Table 52: Comparison of approaches OR, OC, and OH using AVG RMSE, AVG PRECISION, and AVG RECALL

	OR	OC	OH
AVG RMSE	**0,9420**	0,9428	0,9423
AVG PRECISION	0,9083	0,9083	0,9083
AVG RECALL	0,6391	0,6391	0,6391

VI.5.4. Comparison between the ITCRC model and the O'Donovan model

In this subsection, we compare the different variants of our ITCRC model with those of the O'Donovan model using the same dataset (Yelp), as shown in Figure 28 and Table 53 below.

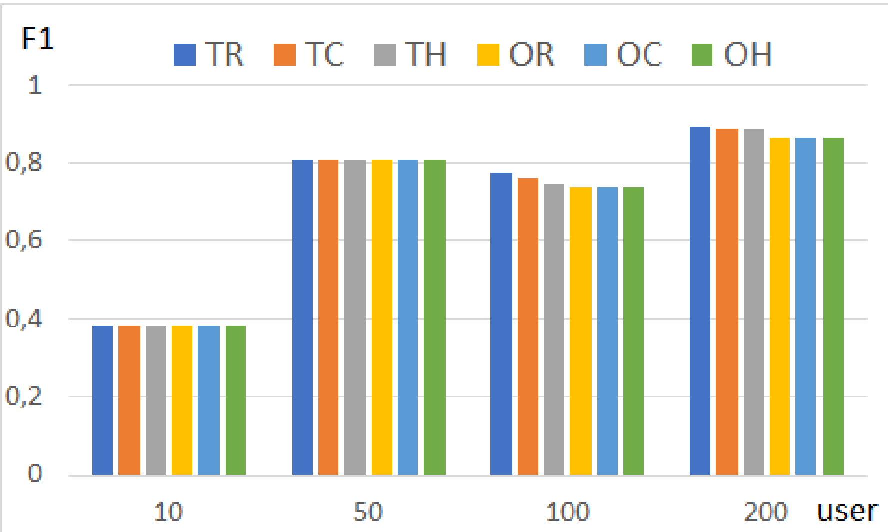

Figure 28: Comparison of TR, TC, TH, OR, OC, and OH approaches using F1

Table 53: Comparison of the TR, TC, and TH approaches with the OR, OC, and OH approaches using AVG RMSE, AVG PRECISION, and AVG RECALL

	TR	TC	TH	OR	OC	OH
AVG RMSE	0,944	0,946	0,948	**0,9420**	0,9428	0,9423
AVG PRECISION	**0,915**	0,913	0,911	0,9083	0,9083	0,9083
AVG RECALL	**0,653**	0,650	0,649	0,6391	0,6391	0,6391

In the Table 53 above, we show that the OR, OC, and OH approaches inspired by the adaptation of O'Donovan's formula to the context of LBSNs outperform the TR, TC, and TH approaches of our ITCRC model by 0.2% in terms of RMSE. However, the latter approaches outperform all the approaches of the O'Donovan model by 0.77% in terms of PRECISION and 0.99% in terms of RECALL.

VI.5.5. Summary of Results and Discussion

In this section, a comparative analysis of the three variants of the ITCRC model was conducted using the Yelp dataset and the metrics of sparsity, RMSE, PRECISION, and RECALL to recommend POIs based on trust derived from ratings, check-ins, or a combination of both. Although these three approaches TR, TC, and TH have a similar density for the trust matrices, the sparsity of the prediction matrices is lower in the hybrid approach (TH), making it an interesting option for fragmented datasets. The approaches OR, OC, and OH, inspired by the adaptation of O'Donovan's formula to the context of LBSNs, outperform the TR, TC, and TH approaches of the ITCRC model. This means that the OR, OC, and OH approaches are more accurate in predicting the actual user rating values. Furthermore, the TR, TC, and TH approaches of the ITCRC model outperform the OR, OC, and OH approaches in terms of PRECISION and RECALL. This shows that the ITCRC model is more effective at identifying relevant POIs and reducing false positives. Finally, these results show that the choice of model and approach depends on the main goal: prediction accuracy (O'Donovan) or recommendation quality (ITCRC).

Conclusion

The ITCRC model, thanks to its hybrid approach (TH), proves to be particularly effective in contexts where data coverage is low, by efficiently reducing the sparsity of the prediction matrices. This characteristic makes it a preferred choice for improving the quality of recommendations, by identifying relevant POIs while minimizing false positives. On the other hand, the OR, OC, and OH approaches, inspired by the adaptation of O'Donovan's formula to the LBSN context, demonstrate better accuracy for predicting POI visits, by minimizing the RMSE parameter. This makes them a suitable solution when the main objective is to provide accurate ratings to users. Thus, the choice of an approach strongly depends on the priority given to the target objective: the accuracy of explicit predictions (O'Donovan) or the quality and relevance of recommendations for fragmented databases (ITCRC model).

General Conclusion

This research work focused on the improvement of recommendation systems in the context of Location-Based Social Networks (LBSNs), emphasizing major issues such as data sparsity, cold-start, and the limitations of traditional approaches like collaborative filtering. Through three main models – HRCT, PRCT, and ITCRC – we explored different strategies to integrate the notion of implicit trust into the recommendation of points of interest (POIs). These models combine various sources of information, such as user ratings and check-ins, in order to deduce implicit trust relationships and improve the quality of predictions.

We can summarize the main contributions of this thesis as follows:

- **The HRCT Model**: This model introduced a hybrid approach combining trust matrices based on ratings and check-ins to build a denser H-Trust matrix. This densification allows for better handling of data sparsity while improving the accuracy of recommendations.
- **The PRCT Model**: By leveraging the principle of implicit trust propagation, this model demonstrated its effectiveness in enriching similarity matrices and mitigating issues related to cold-start. Although the propagation did not always improve performance in terms of RMSE, PRECISION, and RECALL, it significantly reduced matrix sparsity.
- **The ITCRC Model**: This hybrid model combined ratings and check-ins to estimate implicit trust with high precision. It proved to be particularly effective in contexts where data coverage is low, thanks to its ability to reduce the sparsity of prediction matrices and identify relevant POIs.

Among the notable results of this thesis, we can mention the following points:

- Approaches based on check-ins generally outperformed those based on ratings in terms of RMSE, PRECISION, and RECALL, confirming the importance of observable behaviors (such as check-ins) in modeling user preferences.
- The propagation of trust had a positive impact on the density of similarity and prediction matrices, but its effect on predictive performance varies depending on the approach. Some methods, such as SCC (Similarity Cosine Check-in) and SJC (Similarity Jaccard Check-in), proved to be robust in the face of propagation.
- The ITCRC model demonstrated a significant reduction in the sparsity of prediction matrices thanks to its hybrid approach, while offering better recommendation quality in terms of PRECISION and RECALL compared to models inspired by O'Donovan's formula.

In conclusion, this research has demonstrated the importance of integrating implicit trust into recommender systems to mitigate classical challenges such as data sparsity and the cold-start problem. The proposed models – HRCT, PRCT, and ITCRC – offer innovative solutions to improve the quality of recommendations in LBSNs. However, the choice of model and approach should be guided by the main objective: prioritizing the accuracy of predictions (models based on O'Donovan) or the quality and relevance of recommendations (hybrid models like ITCRC).

As short-term perspectives, a deeper exploration of propagation mechanisms could improve predictive performance while preserving matrix density. Moreover, the integration of additional contextual data, such as temporal and geographical information, would increase the relevance of recommendations. In the longer term, the focus will be on optimizing model performance, whose effectiveness largely depends on the quality and density of the initial data. Although approaches based on implicit trust show strong potential, they require precise adjustments to ensure their applicability in real-world environments.

Despite the progress achieved, several challenges remain. One of the main limitations of our models lies in their computational complexity, which may hinder their scalability when applied to large-scale location-based social networks (LBSNs) experiencing rapid growth in user and POI data. Future work should consider exploring optimization techniques or parallel processing architectures to enhance both the scalability and computational efficiency of these models. Furthermore, relying on user check-ins and ratings to infer implicit trust may introduce biases, particularly in datasets with uneven user participation. Additional research could investigate the integration of alternative data sources, such as social media interactions or geospatial information, to mitigate these biases and strengthen the robustness of trust inference mechanisms.

Finally, our models are 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.

References

Alahmadi, D. H., & Zeng, X.-J. (2015). ISTS : Implicit social trust and sentiment based approach to recommender systems. *Expert Systems with Applications*, *42*(22), 8840-8849. https://doi.org/10.1016/j.eswa.2015.07.036

An, J., Jiang, W., & Li, G. (2023). Bidirectional Trust-Enhanced Collaborative Filtering for Point-of-Interest Recommendation. *Sensors*, *23*(8), 4140. https://doi.org/10.3390/s23084140

Anwar, T., Uma, V., & Srivastava, G. (2021). Rec-CFSVD++ : Implementing Recommendation System Using Collaborative Filtering and Singular Value Decomposition (SVD)++. *International Journal of Information Technology & Decision Making*, *20*(04), 1075-1093. https://doi.org/10.1142/S0219622021500310

Aouali, I., Benhalloum, A., Bompaire, M., Heymann, B., Jeunen, O., Rohde, D., Sakhi, O., & Vasile, F. (2022). *Offline Evaluation of Reward-Optimizing Recommender Systems : The Case of Simulation* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2209.08642

Bag, S., Kumar, S. K., & Tiwari, M. K. (2019). An efficient recommendation generation using relevant Jaccard similarity. *Information Sciences*, *483*, 53-64. https://doi.org/10.1016/j.ins.2019.01.023

Bao, J., Zheng, Y., & Mokbel, M. F. (2012). Location-based and preference-aware recommendation using sparse geo-social networking data. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 199-208. https://doi.org/10.1145/2424321.2424348

Bao, J., Zheng, Y., Wilkie, D., & Mokbel, M. (2015). Recommendations in location-based social networks : A survey. *GeoInformatica*, *19*(3), 525-565. https://doi.org/10.1007/s10707-014-0220-8

Baral, R., Li, T., & Zhu, X. (2018). *CAPS : Context Aware Personalized POI Sequence Recommender System* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1803.01245

Bedi, P., & Sharma, R. (2012). Trust based recommender system using ant colony for trust computation. *Expert Systems with Applications*, *39*(1), 1183-1190. https://doi.org/10.1016/j.eswa.2011.07.124

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, *23*(6), 520-528. https://doi.org/10.1016/j.knosys.2010.03.009

Braunhofer, M., Elahi, M., Ricci, F., & Schievenin, T. (2013). Context-Aware Points of Interest Suggestion with Dynamic Weather Data Management. In Z. Xiang & I. Tussyadiah (Éds.), *Information and Communication Technologies in Tourism 2014* (p. 87-100). Springer International Publishing. https://doi.org/10.1007/978-3-319-03973-2_7

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. *Proceedings of the 22nd International Conference on Machine Learning - ICML '05*, 89-96. https://doi.org/10.1145/1102351.1102363

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews : The state of the art. *User Modeling and User-Adapted Interaction*, *25*(2), 99-154. https://doi.org/10.1007/s11257-015-9155-5

Cheng, C., Yang, H., King, I., & Lyu, M. (2021). Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *26*(1), 17-23. https://doi.org/10.1609/aaai.v26i1.8100

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility : User movement in location-based social networks. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1082-1090. https://doi.org/10.1145/2020408.2020579

Cui, Q., Wu, S., Liu, Q., Zhong, W., & Wang, L. (2020). MV-RNN : A Multi-View Recurrent Neural Network for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *32*(2), 317-331. https://doi.org/10.1109/TKDE.2018.2881260

Davtalab, M., & Alesheikh, A. A. (2021). A POI recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization. *Knowledge and Information Systems*, *63*(1), 65-85. https://doi.org/10.1007/s10115-020-01509-5

Débora Nice Ferrari Barbosa, Jorge Luis Victória Barbosa, Sandro José Rigo, & Tiago Wiedmann. (s. d.). *RecSim : A Model for Learning Objects Recommendation using Similarity of Sessions*. Verlag der Technischen Universität Graz. https://doi.org/10.3217/JUCS-022-08-1175

Demirci, U., & Karagoz, P. (2022). Explicit and Implicit Trust Modeling for Recommendation. *Digital*, *2*(4), 444-462. https://doi.org/10.3390/digital2040024

Devooght, R., & Bersini, H. (2017). Long and Short-Term Recommendations with Recurrent Neural Networks. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 13-21. https://doi.org/10.1145/3079628.3079670

Ekaterina, G., Ivan, D., & Oksana, S. (2020). A trust and relevance-based Point-Of-Interest recommendations method with inaccessible user location. *Procedia Computer Science*, *178*, 153-161. https://doi.org/10.1016/j.procs.2020.11.017

Eravci, B., Bulut, N., Etemoglu, C., & Ferhatosmanoglu, H. (2016). Location Recommendations for New Businesses Using Check-in Data. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 1110-1117. https://doi.org/10.1109/ICDMW.2016.0160

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019). Graph Neural Networks for Social Recommendation. *The World Wide Web Conference*, 417-426. https://doi.org/10.1145/3308558.3313488

Gao, C., Wang, X., He, X., & Li, Y. (2022). Graph Neural Networks for Recommender System. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1623-1625. https://doi.org/10.1145/3488560.3501396

Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., & Li, Y. (2023). A Survey of Graph Neural Networks for Recommender Systems : Challenges, Methods, and

Directions. *ACM Transactions on Recommender Systems*, *1*(1), 1-51. https://doi.org/10.1145/3568022

Gao, H., Tang, J., Hu, X., & Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. *Proceedings of the 7th ACM Conference on Recommender Systems*, 93-100. https://doi.org/10.1145/2507157.2507182

Gao, H., Tang, J., Hu, X., & Liu, H. (2015). Content-Aware Point of Interest Recommendation on Location-Based Social Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *29*(1). https://doi.org/10.1609/aaai.v29i1.9462

Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., & Alameda-Pineda, X. (2020). *Dynamical Variational Autoencoders : A Comprehensive Review*. https://doi.org/10.48550/ARXIV.2008.12595

Golbeck, J. (2006). Generating Predictive Movie Recommendations from Trust in Social Networks. In K. Stølen, W. H. Winsborough, F. Martinelli, & F. Massacci (Éds.), *Trust Management* (Vol. 3986, p. 93-104). Springer Berlin Heidelberg. https://doi.org/10.1007/11755593_8

Golbeck, J., & Hendler, J. (2006). FilmTrust : Movie recommendations using trust in web-based social networks. *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, *1*, 282-286. https://doi.org/10.1109/CCNC.2006.1593032

Guo, G., Zhang, J., & Thalmann, D. (2012). A Simple But Effective Method to Incorporate Trusted Neighbors in Recommender Systems. In J. Masthoff, B. Mobasher, M. C. Desmarais, & R. Nkambou (Éds.), *User Modeling, Adaptation, and Personalization* (Vol. 7379, p. 114-125). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31454-4_10

Guo, G., Zhang, J., & Yorke-Smith, N. (2015). TrustSVD : Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. *Proceedings of the AAAI Conference on Artificial Intelligence*, *29*(1). https://doi.org/10.1609/aaai.v29i1.9153

Guo, L., Jiang, H., Wang, X., & Liu, F. (2017). Learning to Recommend Point-of-Interest with the Weighted Bayesian Personalized Ranking Method in LBSNs. *Information*, *8*(1), 20. https://doi.org/10.3390/info8010020

Gupta, M., Thakkar, A., Aashish, Gupta, V., & Rathore, D. P. S. (2020). Movie Recommender System Using Collaborative Filtering. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 415-420. https://doi.org/10.1109/ICESC48915.2020.9155879

Hadjhenni, M., Dennouni, N., & Slama, Z. (2024). Toward a Systematic Evaluation Approach of Point Of InterestRecommendation Algorithms of a Novel Smart Tourism Tool. *International Journal of Computing and Digital Systems*, *15*(1), 653-670. https://doi.org/10.12785/ijcds/150148

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5-53. https://doi.org/10.1145/963770.963772

Hidasi, B., & Czapp, Á. T. (2023). Widespread Flaws in Offline Evaluation of Recommender Systems. *Proceedings of the 17th ACM Conference on Recommender Systems*, 848-855. https://doi.org/10.1145/3604915.3608839

Huang, Z., Ma, J., Dong, Y., Foutz, N. Z., & Li, J. (2022). Empowering Next POI Recommendation with Multi-Relational Modeling. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2034-2038. https://doi.org/10.1145/3477495.3531801

Hwang, C.-S., & Chen, Y.-P. (2007). Using Trust in Collaborative Filtering Recommendation. In H. G. Okuno & M. Ali (Éds.), *New Trends in Applied Artificial Intelligence* (Vol. 4570, p. 1052-1060). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73325-6_105

Inderprastha Engineering College, AKTU, Singh, R. H., Maurya, S., Inderprastha Engineering College, AKTU, Tripathi, T., Inderprastha Engineering College, AKTU, Narula, T., Inderprastha Engineering College, AKTU, Srivastav, G., & Inderprastha Engineering College, AKTU. (2020). Movie Recommendation System using Cosine Similarity and KNN. *International Journal of Engineering and Advanced Technology*, *9*(5), 556-559. https://doi.org/10.35940/ijeat.E9666.069520

Islam, Md. A., Mohammad, M. M., Sarathi Das, S. S., & Ali, M. E. (2022). A survey on deep learning based Point-of-Interest (POI) recommendations. *Neurocomputing*, *472*, 306‑325. https://doi.org/10.1016/j.neucom.2021.05.114

Jamali, M., & Ester, M. (2009). *TrustWalker* : A random walk model for combining trust-based and item-based recommendation. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 397‑406. https://doi.org/10.1145/1557019.1557067

Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the Fourth ACM Conference on Recommender Systems*, 135‑142. https://doi.org/10.1145/1864708.1864736

Jeunen, O., Potapov, I., & Ustimenko, A. (2024). On (Normalised) Discounted Cumulative Gain as an Off-Policy Evaluation Metric for Top- *n* Recommendation. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1222‑1233. https://doi.org/10.1145/3637528.3671687

Kasalický, P., Alves, R., & Kordík, P. (2023). *Bridging Offline-Online Evaluation with a Time-dependent and Popularity Bias-free Offline Metric for Recommenders* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2308.06885

Kim, J. K., Kim, H. K., Oh, H. Y., & Ryu, Y. U. (2010). A group recommendation system for online communities. *International Journal of Information Management*, *30*(3), 212‑219. https://doi.org/10.1016/j.ijinfomgt.2009.09.006

Krumm, J. (2009). A survey of computational location privacy. *Personal and Ubiquitous Computing*, *13*(6), 391‑399. https://doi.org/10.1007/s00779-008-0212-5

Lathia, N., Hailes, S., & Capra, L. (2008). Trust-Based Collaborative Filtering. In Y. Karabulut, J. Mitchell, P. Herrmann, & C. D. Jensen (Éds.), *Trust Management II* (Vol. 263, p. 119‑134). Springer US. https://doi.org/10.1007/978-0-387-09428-1_8

Lemire, D., & Maclachlan, A. (2005). Slope One Predictors for Online Rating-Based Collaborative Filtering. *Proceedings of the 2005 SIAM International Conference on Data Mining*, 471-475. https://doi.org/10.1137/1.9781611972757.43

Leng, Y., Yu, L., & Niu, X. (2022). Dynamically aggregating individuals' social influence and interest evolution for group recommendations. *Information Sciences*, *614*, 223-239. https://doi.org/10.1016/j.ins.2022.09.058

Li, D., Jin, R., Gao, J., & Liu, Z. (2020). On Sampling Top-K Recommendation Evaluation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2114-2124. https://doi.org/10.1145/3394486.3403262

Li, H., Li, K., An, J., Zheng, W., & Li, K. (2019). An efficient manifold regularized sparse non-negative matrix factorization model for large-scale recommender systems on GPUs. *Information Sciences*, *496*, 464-484. https://doi.org/10.1016/j.ins.2018.07.060

Li, M., Li, J., Yang, L., & Ding, Q. (2024). Self-Supervised Hypergraph Learning for Knowledge-Aware Social Recommendation. *Electronics*, *13*(7), 1306. https://doi.org/10.3390/electronics13071306

Li, R. Z., Urbano, J., & Hanjalic, A. (2021). New Insights into Metric Optimization for Ranking-based Recommendation. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 932-941. https://doi.org/10.1145/3404835.3462973

Li, Y., Luo, Y., Zhang, Z., Sadiq, S., & Cui, P. (2019). Context-Aware Attention-Based Data Augmentation for POI Recommendation. *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 177-184. https://doi.org/10.1109/ICDEW.2019.00-14

Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., & Rui, Y. (2014). GeoMF : Joint geographical modeling and matrix factorization for point-of-interest recommendation. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 831-840. https://doi.org/10.1145/2623330.2623638

Lifen, L. (2008). Trust Derivation and Transitivity in a Recommendation Trust Model. *2008 International Conference on Computer Science and Software Engineering*, 770-773. https://doi.org/10.1109/CSSE.2008.484

Liu, B., Fu, Y., Yao, Z., & Xiong, H. (2013). Learning geographical preferences for point-of-interest recommendation. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1043-1051. https://doi.org/10.1145/2487575.2487673

Liu, B., & Xiong, H. (2013). Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness. *Proceedings of the 2013 SIAM International Conference on Data Mining*, 396-404. https://doi.org/10.1137/1.9781611972832.44

Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, *56*, 156-166. https://doi.org/10.1016/j.knosys.2013.11.006

Logesh, R., & Subramaniyaswamy, V. (2017). A Reliable Point of Interest Recommendation based on Trust Relevancy between Users. *Wireless Personal Communications*, *97*(2), 2751-2780. https://doi.org/10.1007/s11277-017-4633-1

Ma, H., King, I., & Lyu, M. R. (2009). Learning to recommend with social trust ensemble. *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 203-210. https://doi.org/10.1145/1571941.1571978

Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). Recommender systems with social regularization. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 287-296. https://doi.org/10.1145/1935826.1935877

Macedo, A. Q., Marinho, L. B., & Santos, R. L. T. (2015). Context-Aware Event Recommendation in Event-based Social Networks. *Proceedings of the 9th ACM Conference on Recommender Systems*, 123-130. https://doi.org/10.1145/2792838.2800187

Massa, P., & Avesani, P. (2004). Trust-Aware Collaborative Filtering for Recommender Systems. In R. Meersman & Z. Tari (Éds.), *On the Move to Meaningful Internet Systems 2004 : CoopIS, DOA, and ODBASE* (Vol. 3290, p. 492-508). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30468-5_31

Massa, P., & Avesani, P. (2007). Trust-aware recommender systems. *Proceedings of the 2007 ACM Conference on Recommender Systems*, 17-24. https://doi.org/10.1145/1297231.1297235

Medjroud, S., Dennouni, N., Hadj Henni, M., & Bettache, D. (2022). Towards a new POI Recommendation Approach based on Implicit Trust between users. *2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA)*, 19-24. https://doi.org/10.1109/BIWA57631.2022.10038190

Medjroud, S., Dennouni, N., & Loukam, M. (2025a). Point of Interest Recommendation using Implicit Trust based on Combining Ratings and Check-ins of Smartphone Users. *Engineering, Technology & Applied Science Research*, *15*(2), 21249-21256. https://doi.org/10.48084/etasr.9965

Medjroud, S., Dennouni, N., & Loukam, M. (2025b). Towards a systematic point-of-interest recommendations based on trust between users deduced from their ratings and check-ins in a LBSN. *International Journal of Computing and Digital Systems*, *17*(1), 1-15. https://doi.org/10.12785/ijcds/1571107232

Migliorini, S., Carra, D., & Belussi, A. (2021). Distributing Tourists among POIs with an Adaptive Trip Recommendation System. *IEEE Transactions on Emerging Topics in Computing*, *9*(4), 1765-1779. https://doi.org/10.1109/TETC.2019.2920484

Moriya, Y., & Jones, Gareth. J. F. (2023). Improving Noise Robustness for Spoken Content Retrieval Using Semi-Supervised ASR and N-Best Transcripts for BERT-Based Ranking Models. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 398-405. https://doi.org/10.1109/SLT54892.2023.10023197

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2021). What Yelp Fake Review Filter Might Be Doing? *Proceedings of the International AAAI Conference on Web and Social Media*, *7*(1), 409-418. https://doi.org/10.1609/icwsm.v7i1.14389

Nilla, A., & Setiawan, E. B. (2024). Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, *10*(1), 17. https://doi.org/10.26555/jiteki.v9i4.28113

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 167-174. https://doi.org/10.1145/1040830.1040870

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). *Finding Deceptive Opinion Spam by Any Stretch of the Imagination*. https://doi.org/10.48550/ARXIV.1107.4557

Ouyang, Y., Zhang, J., Xie, W., Rong, W., & Xiong, Z. (2016). Implicit and Explicit Trust in Collaborative Filtering. In F. Lehner & N. Fteimi (Éds.), *Knowledge Science, Engineering and Management* (Vol. 9983, p. 489-500). Springer International Publishing. https://doi.org/10.1007/978-3-319-47650-6_39

Pal, B., & Jenamani, M. (2019). Trust inference using implicit influence and projected user network for item recommendation. *Journal of Intelligent Information Systems*, *52*(2), 425-450. https://doi.org/10.1007/s10844-018-0537-0

Papagelis, M., Plexousakis, D., & Kutsuras, T. (2005). Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. In P. Herrmann, V. Issarny, & S. Shiu (Éds.), *Trust Management* (Vol. 3477, p. 224-239). Springer Berlin Heidelberg. https://doi.org/10.1007/11429760_16

Pitsilis, G., & Marshall, L. F. (2008). Modeling Trust for Recommender Systems using Similarity Metrics. In Y. Karabulut, J. Mitchell, P. Herrmann, & C. D. Jensen (Éds.), *Trust Management II* (Vol. 263, p. 103-118). Springer US. https://doi.org/10.1007/978-0-387-09428-1_7

Pu, P., Chen, L., & Hu, R. (2011). A user-centric evaluation framework for recommender systems. *Proceedings of the Fifth ACM Conference on Recommender Systems*, 157-164. https://doi.org/10.1145/2043932.2043962

Qian, Y., Fan, Y., Hu, W., & Soong, F. K. (2014). On the training aspects of Deep Neural Network (DNN) for parametric TTS synthesis. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3829-3833. https://doi.org/10.1109/ICASSP.2014.6854318

Rahmani, H. A., Naghiaei, M., Tourani, A., & Deldjoo, Y. (2022). Exploring the Impact of Temporal Bias in Point-of-Interest Recommendation. *Proceedings of the 16th ACM Conference on Recommender Systems*, 598-603. https://doi.org/10.1145/3523227.3551481

Rampisela, T. V., Maistro, M., Ruotsalo, T., & Lioma, C. (2023). *Evaluation Measures of Individual Item Fairness for Recommender Systems : A Critical Study*. https://doi.org/10.48550/ARXIV.2311.01013

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens : An open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work - CSCW '94*, 175-186. https://doi.org/10.1145/192844.192905

Roy, F., Sarwar, S. M., & Hasan, M. (2015). User Similarity Computation for Collaborative Filtering Using Dynamic Implicit Trust. In M. Yu. Khachay, N. Konstantinova, A. Panchenko, D. Ignatov, & V. G. Labunets (Éds.), *Analysis of Images, Social Networks and Texts* (Vol. 542, p. 224-235). Springer International Publishing. https://doi.org/10.1007/978-3-319-26123-2_22

Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, 880-887. https://doi.org/10.1145/1390156.1390267

Sánchez, P., & Bellogín, A. (2022). Point-of-Interest Recommender Systems Based on Location-Based Social Networks : A Survey from an Experimental Perspective. *ACM Computing Surveys*, *54*(11s), 1-37. https://doi.org/10.1145/3510409

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web*, 285-295. https://doi.org/10.1145/371920.372071

Scellato, S., Noulas, A., & Mascolo, C. (2011). Exploiting place features in link prediction on location-based social networks. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1046-1054. https://doi.org/10.1145/2020408.2020575

Schwartz, I. (2021). *Ensemble of MRR and NDCG models for Visual Dialog* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2104.07511

Seyedhoseinzadeh, K., Rahmani, H. A., Afsharchi, M., & Aliannejadi, M. (2022). Leveraging social influence based on users activity centers for point-of-interest recommendation. *Information Processing & Management*, *59*(2), 102858. https://doi.org/10.1016/j.ipm.2021.102858

Shambour, Q., & Lu, J. (2011). A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services. *International Journal of Intelligent Systems*, *26*(9), 814-843. https://doi.org/10.1002/int.20495

Shambour, Q., & Lu, J. (2012). A trust-semantic fusion-based recommendation approach for e-business applications. *Decision Support Systems*, *54*(1), 768-780. https://doi.org/10.1016/j.dss.2012.09.005

Shambour, Q., & Lu, J. (2015). An effective recommender system by unifying user and item trust information for B2B applications. *Journal of Computer and System Sciences*, *81*(7), 1110-1126. https://doi.org/10.1016/j.jcss.2014.12.029

Shareef, M. A., Kapoor, K. K., Mukerji, B., Dwivedi, R., & Dwivedi, Y. K. (2020). Group behavior in social media : Antecedents of initial trust formation. *Computers in Human Behavior*, *105*, 106225. https://doi.org/10.1016/j.chb.2019.106225

Sheugh, L., & Alizadeh, S. H. (2015). A note on pearson correlation coefficient as a metric of similarity in recommender system. *2015 AI & Robotics (IRANOPEN)*, 1-6. https://doi.org/10.1109/RIOS.2015.7270736

Shokri, R., Theodorakopoulos, G., Le Boudec, J.-Y., & Hubaux, J.-P. (2011). Quantifying Location Privacy. *2011 IEEE Symposium on Security and Privacy*, 247-262. https://doi.org/10.1109/SP.2011.18

Singh, M. (2020). Scalability and sparsity issues in recommender datasets : A survey. *Knowledge and Information Systems*, *62*(1), 1-43. https://doi.org/10.1007/s10115-018-1254-2

Song, C., Wen, J., & Li, S. (2019). Personalized POI recommendation based on check-in data and geographical-regional influence. *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, 128-133. https://doi.org/10.1145/3310986.3311034

Sun, A. (2023). Take a Fresh Look at Recommender Systems from an Evaluation Standpoint. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2629-2638. https://doi.org/10.1145/3539618.3591931

Sun, Z., Han, L., Huang, W., Wang, X., Zeng, X., Wang, M., & Yan, H. (2015). Recommender systems based on social networks. *Journal of Systems and Software*, *99*, 109-119. https://doi.org/10.1016/j.jss.2014.09.019

Takács, G., & Tikk, D. (2012). Alternating least squares for personalized ranking. *Proceedings of the Sixth ACM Conference on Recommender Systems*, 83-90. https://doi.org/10.1145/2365952.2365972

Tamm, Y.-M., Damdinov, R., & Vasilev, A. (2021). Quality Metrics in Recommender Systems : Do We Calculate Metrics Consistently? *Fifteenth ACM Conference on Recommender Systems*, 708-713. https://doi.org/10.1145/3460231.3478848

Tang, J., Hu, X., & Liu, H. (2013). Social recommendation : A review. *Social Network Analysis and Mining*, *3*(4), 1113-1133. https://doi.org/10.1007/s13278-013-0141-9

Tong, X., Zhang, W., Long, Y., & Huang, H. (2013). Subjectivity and Objectivity of Trust. In L. Cao, Y. Zeng, A. L. Symeonidis, V. I. Gorodetsky, P. S. Yu, & M. P. Singh (Éds.), *Agents and Data Mining Interaction* (Vol. 7607, p. 105-114). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36288-0_10

Trattner, C., Oberegger, A., Eberhard, L., Parra, D., & Marinho, L. (2016). *Understanding the Impact of Weather for POI Recommendations*. https://doi.org/10.13140/RG.2.2.24392.11527

Trattner, C., Oberegger, A., Marinho, L., & Parra, D. (2018). Investigating the utility of the weather context for point of interest recommendations. *Information Technology & Tourism*, *19*(1-4), 117-150. https://doi.org/10.1007/s40558-017-0100-9

Wan, L., Wang, H., Hong, Y., Li, R., Chen, W., & Huang, Z. (2022). iTourSPOT : A context-aware framework for next POI recommendation in location-based social networks. *International Journal of Digital Earth*, *15*(1), 1614-1636. https://doi.org/10.1080/17538947.2022.2122611

Wang, H., Terrovitis, M., & Mamoulis, N. (2013). Location recommendation in location-based social networks using user check-in data. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 374-383. https://doi.org/10.1145/2525314.2525357

Wang, J., De Vries, A. P., & Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 501-508. https://doi.org/10.1145/1148170.1148257

Wang, K.-Y., Ting, I.-H., & Wu, H.-J. (2013). Discovering interest groups for marketing in virtual communities : An integrated approach. *Journal of Business Research*, *66*(9), 1360-1366. https://doi.org/10.1016/j.jbusres.2012.02.037

Wang, W., Chen, J., Wang, J., Chen, J., Liu, J., & Gong, Z. (2020). Trust-Enhanced Collaborative Filtering for Personalized Point of Interests Recommendation. *IEEE Transactions on Industrial Informatics*, *16*(9), 6124-6132. https://doi.org/10.1109/TII.2019.2958696

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T.-S. (2019). KGAT : Knowledge Graph Attention Network for Recommendation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 950-958. https://doi.org/10.1145/3292500.3330989

Wang, X., Ounis, I., & Macdonald, C. (2021). *Leveraging Review Properties for Effective Recommendation* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2102.03089

Wang, Y., & Ba, S. (2023). *Producer-Side Experiments Based on Counterfactual Interleaving Designs for Online Recommender Systems* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2310.16294

Wang, Z., Liao, J., Cao, Q., Qi, H., & Wang, Z. (2015). Friendbook : A Semantic-Based Friend Recommendation System for Social Networks. *IEEE Transactions on Mobile Computing*, *14*(3), 538-551. https://doi.org/10.1109/TMC.2014.2322373

Werneck, H., Silva, N., Viana, M., Pereira, A. C. M., Mourão, F., & Rocha, L. (2021). Points of Interest recommendations : Methods, evaluation, and future directions. *Information Systems*, *101*, 101789. https://doi.org/10.1016/j.is.2021.101789

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2023). Graph Neural Networks in Recommender Systems : A Survey. *ACM Computing Surveys*, *55*(5), 1-37. https://doi.org/10.1145/3535101

Xin Luo, Mengchu Zhou, Yunni Xia, & Qingsheng Zhu. (2014). An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. *IEEE Transactions on Industrial Informatics*, *10*(2), 1273-1284. https://doi.org/10.1109/TII.2014.2308433

Xu, C., Ding, A. S., & Zhao, K. (2021). A novel POI recommendation method based on trust relationship and spatial–temporal factors. *Electronic Commerce Research and Applications*, *48*, 101060. https://doi.org/10.1016/j.elerap.2021.101060

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2021). Towards the D-Optimal Online Experiment Design for Recommender Selection. *Proceedings of the 27th ACM SIGKDD*

Conference on Knowledge Discovery & Data Mining, 3817-3825. https://doi.org/10.1145/3447548.3467192

Yang, B., Lei, Y., Liu, J., & Li, W. (2017). Social Collaborative Filtering by Trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(8), 1633-1647. https://doi.org/10.1109/TPAMI.2016.2605085

Yang, D., Zhang, D., Yu, Z., & Wang, Z. (2013). A sentiment-enhanced personalized location recommendation system. *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, 119-128. https://doi.org/10.1145/2481492.2481505

Ye, M., Yin, P., & Lee, W.-C. (2010). Location recommendation for location-based social networks. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 458-461. https://doi.org/10.1145/1869790.1869861

Ye, M., Yin, P., Lee, W.-C., & Lee, D.-L. (2011). Exploiting geographical influence for collaborative point-of-interest recommendation. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 325-334. https://doi.org/10.1145/2009916.2009962

Yin, H., Sun, Y., Cui, B., Hu, Z., & Chen, L. (2013). LCARS : A location-content-aware recommender system. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 221-229. https://doi.org/10.1145/2487575.2487608

Yu, J., Guo, L., Zhang, J., & Wang, G. (2024). A survey on graph neural network-based next POI recommendation for smart cities. *Journal of Reliable Intelligent Environments*, *10*(3), 299-318. https://doi.org/10.1007/s40860-024-00233-z

Yuan, Q., Cong, G., Ma, Z., Sun, A., & Thalmann, N. M.-. (2013). Time-aware point-of-interest recommendation. *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 363-372. https://doi.org/10.1145/2484028.2484030

Yuan, W., Shu, L., Chao, H.-C., Guan, D., Lee, Y.-K., & Lee, S. (2010). iTARS : Trust-aware recommender system using implicit trust networks. *IET Communications*, *4*(14), 1709-1721. https://doi.org/10.1049/iet-com.2009.0733

Zahir, A., Yuan, Y., & Moniz, K. (2019). AgreeRelTrust—A Simple Implicit Trust Inference Model for Memory-Based Collaborative Filtering Recommendation Systems. *Electronics*, *8*(4), 427. https://doi.org/10.3390/electronics8040427

Zangerle, E., & Bauer, C. (2023). Evaluating Recommender Systems : Survey and Framework. *ACM Computing Surveys*, *55*(8), 1-38. https://doi.org/10.1145/3556536

Zeng, J., He, X., Li, F., & Wu, Y. (2020). A recommendation algorithm for point of interest using time-based collaborative filtering. *International Journal of Information Technology and Management*, *19*(4), 347. https://doi.org/10.1504/IJITM.2020.110242

Zhang, F., Wang, H., & Yi, H. (2014). An Adaptive Recommendation Method Based on Small-World Implicit Trust Network. *Journal of Computers*, *9*(3), 618-625. https://doi.org/10.4304/jcp.9.3.618-625

Zhang, H., Gan, M., & Sun, X. (2021). Incorporating Memory-Based Preferences and Point-of-Interest Stickiness into Recommendations in Location-Based Social Networks. *ISPRS International Journal of Geo-Information*, *10*(1), 36. https://doi.org/10.3390/ijgi10010036

Zhang, J.-D., & Chow, C.-Y. (2013). iGSLR : Personalized geo-social location recommendation: a kernel density estimation approach. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 334-343. https://doi.org/10.1145/2525314.2525339

Zhang, J.-D., & Chow, C.-Y. (2015). GeoSoCa : Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 443-452. https://doi.org/10.1145/2766462.2767711

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep Learning Based Recommender System : A Survey and New Perspectives. *ACM Computing Surveys*, *52*(1), 1-38. https://doi.org/10.1145/3285029

Zhang, Y., & Pennacchiotti, M. (2013). Predicting purchase behaviors from social media. *Proceedings of the 22nd International Conference on World Wide Web*, 1521-1532. https://doi.org/10.1145/2488388.2488521

Zhang, Z., & Liu, H. (2015). Social recommendation model combining trust propagation and sequential behaviors. *Applied Intelligence*, *43*(3), 695-706. https://doi.org/10.1007/s10489-015-0681-y

Zhao, T., McAuley, J., & King, I. (2014). Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 261-270. https://doi.org/10.1145/2661829.2661998

Zheng, Y., Liu, Y., Yuan, J., & Xie, X. (2011). Urban computing with taxicabs. *Proceedings of the 13th International Conference on Ubiquitous Computing*, 89-98. https://doi.org/10.1145/2030112.2030126

Zheng, Y., Zhang, L., Xie, X., & Ma, W.-Y. (2009). Mining interesting locations and travel sequences from GPS trajectories. *Proceedings of the 18th International Conference on World Wide Web*, 791-800. https://doi.org/10.1145/1526709.1526816

Zhi-Dan Zhao & Ming-Sheng Shang. (2010). User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. *2010 Third International Conference on Knowledge Discovery and Data Mining*, 478-481. https://doi.org/10.1109/WKDD.2010.54

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., & Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, *107*(10), 4511-4515. https://doi.org/10.1073/pnas.1000488107

Zhu, J., Ming, Q., & Liu, Y. (2018). Trust-Distrust-Aware Point-of-Interest Recommendation in Location-Based Social Network. In S. Chellappan, W. Cheng, & W. Li (Éds.), *Wireless Algorithms, Systems, and Applications* (Vol. 10874, p. 709-719). Springer International Publishing. https://doi.org/10.1007/978-3-319-94268-1_58

Zhu, J., Wang, C., Guo, X., Ming, Q., Li, J., & Liu, Y. (2019). Friend and POI recommendation based on social trust cluster in location-based social networks. *EURASIP Journal on Wireless Communications and Networking*, *2019*(1), 89. https://doi.org/10.1186/s13638-019-1388-2

MY SCIENTIFIC CONTRIBUTIONS

International Scientific Journals

1. Medjroud, S., Dennouni, N. and Loukam, M. 2025. Point of Interest Recommendation using Implicit Trust based on Combining Ratings and Check-ins of Smartphone Users. *Engineering, Technology & Applied Science Research*. 15, 2 (Apr. 2025), 21249–21256. DOI:https://doi.org/10.48084/etasr.9965.

2. Medjroud, S., Dennouni, N., & Loukam, M. (2024). Towards a systematic point-of-interest recommendations based on trust between users deduced from their ratings and check-ins in a LBSN. *International Journal of Computing and Digital Systems*, *16*(1), 189-203.

3. S. Medjroud, N. Dennouni et M. Loukam, POI Recommendations Using Propagated Implicit User Trust Deduced from Ratings and Check-ins. *Indonesian Journal of Electrical Engineering and Computer Science*. "*in press*".

International Conferences

4. S. Medjroud, N. Dennouni, M. Hadj Henni and D. Bettache, "Towards a new POI Recommendation Approach based on Implicit Trust between users," *2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA)*, Sidi Bel Abbes, Algeria, 2022, pp. 19-24, doi: 10.1109/BIWA57631.2022.10038190.

5. M. H. Henni, N. Dennouni, Z. Slama, S. Medjroud and D. Bettache, "Towards an approach for online evaluation of new variants of content-based POI recommender systems by mobile tourists," *2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA)*, Sidi Bel Abbes, Algeria, 2022, pp. 89-94, doi: 10.1109/BIWA57631.2022.10037820.

6. Bettache, D., Dennouni, N., Hadj Henni, M. H., & MEdjroud, S. (2024, November). Exploring the Impact of Similarity Measures on Implicit Collaborative Filtering in Point of Interest Recommender Systems. 2024 International Conference of the African Federation of Operational Research Societies (AFROS) (AFROS'24, Algeria, nov, 2024. IEEE.

7. Sara Medjroud & Nassim Dennouni, "Towards a New Approach for Point Of Interest Recommendation Based on Implicit Trust deduced from user Check-ins," the First International Conference On Artificial Intelligence, Smart Technologies And Communications AISTC'25. University of Chlef, Algeria, 14-15 April.

8. Sara Medjroud & Nassim Dennouni, “Implicit Trust-Based POI Recommendations Using User Check-Ins,” The 7th International Conference on Pattern Analysis and Intelligent Systems, (PAIS’25). University of Laghouat, Algeria, 23-24 April.

National Conferences

9. Sarah Medjroud, Nassim Dennouni, Mourad Loukam, “Point-Of-Interest recommendation using the trust between tourists in LBSNs,” the First National Conference on Artificial Intelligence, Smart Technologies, and Communications (AISTC'23), University of Chlef, Algeria, 8-9 November 2023.

10. Sarah Medjroud, Nassim Dennouni, Mourad Loukam, "Recommendation System: a review of trust techniques, " National Conference on Artificial Intelligence and its Applications (NCAIA'2023), Tlemcen, Algeria, 2023, pp. 84, doi: 10.6084/m9.figshare.24902382.v3