Smart Travel Super-App for Personalized, Adaptive, and Collaborative Travel Planning

Madhusudhanan S.[1], Monic Auditya A.[2], Monish D.Y.[2], Kamalesh P.[2]

[1]Associate Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, India
[2]Student, Department of Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, India

madhusudhan.s@rajalakshmi.edu.in, 230701194@rajalakshmi.edu.in,
230701195@rajalakshmi.edu.in, 230701137@rajalakshmi.edu.in

***Abstract*—Traveling around the world can be a fun experience, but planning a trip can be tedious - selecting destinations, arranging transportation, finding food, and booking hotels are scattered across different apps. Most travel applications fail to provide adaptive recommendations in real-time, as they do not integrate with live data from sources such as Google Places or Foursquare. This forces customers to switch between several apps, leading to fragmented planning and outdated recommendations. ExploreEase is a travel super-app that utilizes artificial intelligence to integrate all the features of trip planning on one platform. It offers individual- and group-based recommendations by combining structured data (place attributes) with unstructured data (user reviews). Unsupervised learning groups users according to their preferences, neural matrix factorization (NMF) maps user-item interactions, and reinforcement learning supports adaptive planning. The core functionality includes dynamic trip packages themed to the season, three route suggestions optimized for cost savings, time savings, and scenic views, and live travel and accommodation bookings. Group functionality includes trip conversation through chat, polls to make shared decisions such as cancellations, shared expense accounting, and real-time support for emergencies. User reviews are stored in MongoDB to support scalability. In general, ExploreEase transforms challenging travel planning into a seamless and smart experience. challenging travel planning into a seamless and smart experience.**

***Index Terms*—Travel Planning, Recommendation Systems, Neural Matrix Factorization, Reinforcement Learning, Smart Tourism, Real-Time Personalization**

I. INTRODUCTION

The thrill of exploring new places, experiencing different cultures, and making memories that last is what continues to make travel a global activity that people love. With the advancement of technology and digital tools, the traveler has more power than ever before, yet planning a trip is still complicated, time-consuming, and frustrating. Usually, users rely on several disparate apps to choose places, arrange travel, reserve accommodations, monitor expenses, and organize with travel friends, leading to inefficient, fragmented workflows and lost optimization potential.

Classic travel recommendation platforms tend to be based on static data and do not provide real-time adaptive recommendations [1]. They are not efficient at merging structured data (e.g., attributes of a place) with unstructured data (e.g., user feedback) and are seldom refreshed by live information from sources like Google Places or Foursquare [11]. In addition, most current sites lack the intelligence to offer customized user suggestions and do not enable integrated group planning experiences—both of which are important determinants in contemporary digital tourism [2].

To address these constraints, we introduce **ExploreEase**, an intelligent, AI-driven travel super-app that transforms trip planning with smart, integrated, and user-oriented features. The framework makes use of unsupervised learning to cluster user interests, factorization of neural matrix to capture user-item interactions, and reinforcement learning to dynamically update recommendations [5] [12] [13].

ExploreEase comprises the following fundamental features: Individualized and group recommendations based on user profiles, social networks, and interest groups [2] [6] [8], Dynamic trip packages according to seasonal and monthly patterns, optimized with user interest [3], Hotel reservation and transportation booking (flights, trains, and car rentals) through one integrated interface [11], Three optimization modes for private travel (Cost-saving routes, Time-saving routes, Scenic view-enriched routes), Group chat feature for group trip planning, where users can exchange places, make reservations, and decide together, Expense tracker to record shared and personal expenses transparently, Segregation of photos and trip albums, automatically sorting memories by location, time, and group [14].

These functionalities are architected to bring the entire travel experience together under one umbrella, minimizing app-switching and maximizing planning efficiency.

The rest of the paper is organized as follows: Section II presents an overview of related work in recommendation systems and intelligent travel. Section III discusses the organization of the dataset, its sources, and preprocessing techniques. Section IV outlines the proposed methodology and system workflow. Section V describes the experimental configuration, evaluation metrics, and results. Section VI summarizes the project's conclusions. Section VII discusses future improvements and scalability possibilities.

II. Related Works

Several research works were conducted to create a user-friendly travel app that meets all user needs. The major contributions were as follows:

Parikh et al. [1] developed a mobile app called *Travigate* that employs K-means clustering along with CNN image recognition capabilities and provides interest-based and image-matching recommendations. Similar alternatives utilize unsupervised learning and personalized suggestions, but they lack group coordination and real-time data.

Kim et al. [2] proposed *GRec_Tr*, a group recommender that utilizes collaborative filtering and constraint satisfaction to accommodate a variety of group member preferences, contributing to the investigation of group-focused recommendations, polls for consensus that integrate trip packaging.

Kong et al. [3] offered *RPMTD*, a multi-agent reinforcement learning system for overtourism that distributes tourists over routes with dual-congestion reward definitions. This is indicative of an adaptive, sustainability-responsible reinforcement learning approach for dynamic and equitable trip recommendations.

Zhang et al. [4] proposed a tourist routing planning mechanism that utilizes a "comprehensive attractiveness index" taking into consideration cost, duration, and popularity, and is optimized with genetic algorithms. This work is similar in its aims of personalizing routing; however, it adds the ability for real-time flexibility with reinforcement learning.

Sharaff et al. [5] developed a hybrid LMF and popularity model to deal with personalization and cold-start issues, using standard neural matrix factorization (NMF) and a fallback of popularity-based recommendation to improve user-item engagement.

Alenezi and Hirtle [6] developed the Normalized Attraction Travel Personality (NATP) model that applies topic modeling on crowd-sourced reviews to produce attraction embeddings. This model not only accounted for implicit travel personalities in reviews through review semantics, but it also significantly outperformed traditional approaches to prediction and ranking ratings.

Stefanovič and Ramanauskaitė [7] recommended travel destinations by analyzing Instagram photos using the object detection capabilities of pixel-based clustering and SOMs. Other examples of organizing trip photos, and subsequently inferring preferences from media content, will use similar techniques.

Qin et al. [8] developed *DCSGR*, which divides large groups into subgroups based on interests, and uses collaborative filtering to aggregate individual recommendations. Subgroup dynamics and group decision-making features are aided by unsupervised learning.

Tang et al. [9] integrated travel survey data and Amap API data to improve travel mode prediction using XGBoost and SHAP, and worked with other sources of live feed data, including Google Places, to improve recommendation accuracy and responsiveness.

Jia-Xiang et al. [10] developed a multimodal travel route system to calculate trips using taxi, subway, and walking elements. This aligns with support for multiple mixed transport modes and group route customization preferences, such as trust of savings time, savings money, or scenic routes.

Chang et al. [11] improved hotel recommendation performance using heterogeneous social media data and leveraged signal information available to refine suggestions across social platforms.

Overall, these systems exemplify the higher-order sophistication developing in travel recommender systems. Yet, none accounted for an integrated mobile experience encompassing personalized, real-time, and group-based planning as a single challenge.

III. Dataset Description

A. Where the Data Comes From

To construct this dataset, we combined real-world data with some well-constructed synthetic data. For locations and events, we accessed the Google Places API, Foursquare API, and Eventbrite API. These provided us with rich, city-level information on Indian locations, including events, place categories, and descriptions. To simulate users, we went to Kaggle, the "Traveller" dataset containing base location, age, and sentiment data. We created interaction data with Python packages such as Faker and random to mimic how users can interact with one another and with various places, making the dataset more realistic and dynamic.

The entire step from data cleaning to feature engineering is clearly shown in Fig. 1 and are also discussed below.

All uses of API data are done under the terms of use and conditions of the given platforms. The Kaggle dataset is typically covered under a Creative Commons license, and the synthetic data that we generated ourselves is free from any restrictions.

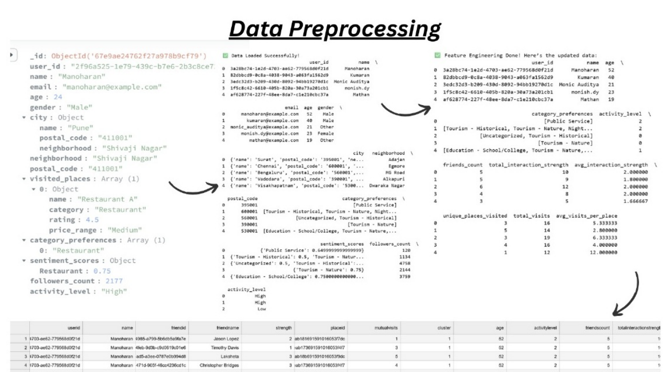

Fig. 1. Data Cleaning, Pre-processing and Feature Engineering

B. How We Collected the Data

Everything was scraped at once — a one-day scrape via APIs. We automated requests to retrieve data using Python scripts and stored it in JSON format.

For user interactions (since we didn't have actual behavioral data), we created custom scripts to mimic them. We randomly assigned the strengths of connections between users (such as how close two individuals are) and approximated the number of times users may visit a place. This made our dataset more comprehensive, networked, and viable for modeling and analysis.

C. What the Dataset Contains

Together, the dataset contains approximately 2,000 samples three main components: places, users, and interactions. Here is a brief overview:

Locations: We collected information from 64 cities in India, divided over various regions — North, South, East, West, Central, North-East, and Union Territories. We found the name and brief description for each location. After that, we categorized them into 15 predefined categories based on rule-based tagging. If a location did not belong to any of the given categories, we passed it through a BERT-based AI model to generate a fresh, smart label.

Users: Extracted from the Kaggle Traveler dataset, the user profiles contain the base city, age (grouped into teen, middle-aged and elderly) and sentiment data. This provided us with a means to examine user behavior along demographic lines.

Interactions: We modeled this aspect with Python. Each user could be linked to other users with a given strength level - low (0), medium (1), or high (2), and their travel to places was recorded as numeric values. This gave us a real-world network of travel and social behavior.

D. Cleaning and Processing the Data

Once we had all the information, we reduced the noise. We reduced fields such as postal codes, full addresses, and neighborhood information — things that weren't useful for our analysis. We then encoded the levels of interaction in numerical terms (low = 0, medium = 1, high = 2).

We also performed some feature engineering: Average visits per location, Levels of activity (based on the frequency of visits of a user to locations), Number of distinct locations visited.

In order to determine user patterns, we partitioned users into clusters based on a combination of features such as total visits, number of distinct locations visited, friend relationships, and strength of interaction. We employed the elbow method to determine the best number of clusters, which happened to be four (determined from the Fig. 2), and then applied PCA (Principal Component Analysis) [14] to map out those clusters in a significant manner.

E. Data Availability and Ethics Statement

The dataset contains publicly available data (Google Places, Foursquare, Eventbrite APIs) as well as synthetic data created with Python scripts. These resources can be shared on reasonable request. This experiment did not involve real human subjects or personal user data. All user-related data were synthetically generated and anonymized.

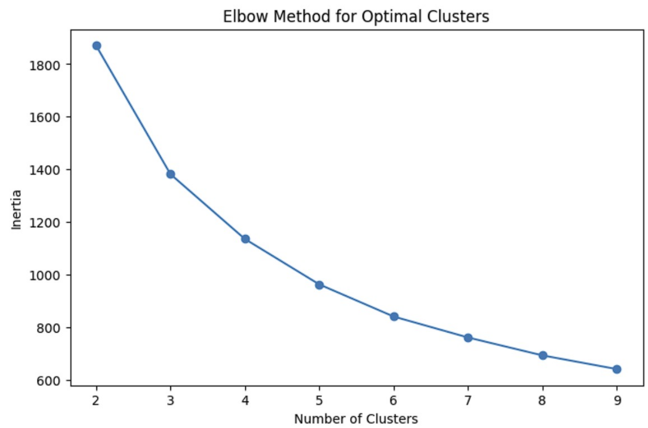

Fig. 2. Elbow method - graph to determine the best number of clusters

IV. Methodology

ExploreEase utilizes a modular microservices architecture featuring RESTful APIs and MongoDB for a scalable, real-time integration platform. The ExploreEase system is delineated into several modules, as seen in Fig. 4.

A. Personalized and Group-Based Recommendations Module

This module builds off important systems such as K-Means Clustering, Neural Matrix Factorization (NMF), and Reinforcement Learning, into a single recommendation pipeline.

K-Means clustering groups users into four behavioral segments according to their travel patterns (Frequent Travellers, Loyal Locals, Moderate Explorers, Casual Travellers) based on visit frequency and interaction strength among social ties that are visualized using PCA [14] as shown in Fig. 3. Each user is assigned a `cluster_no` representing this behavior segment.

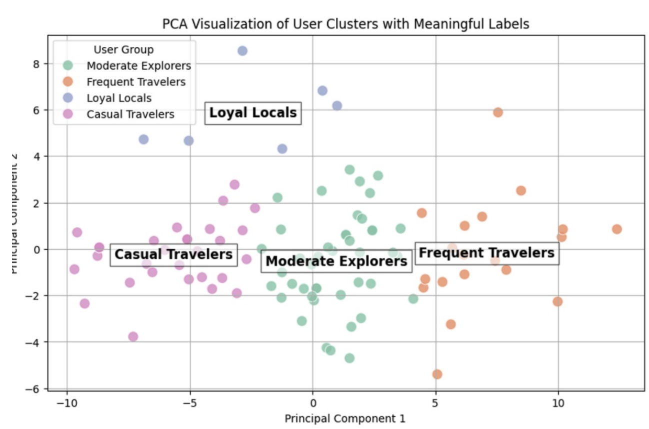

Fig. 3. PCA visualization of User Clusters

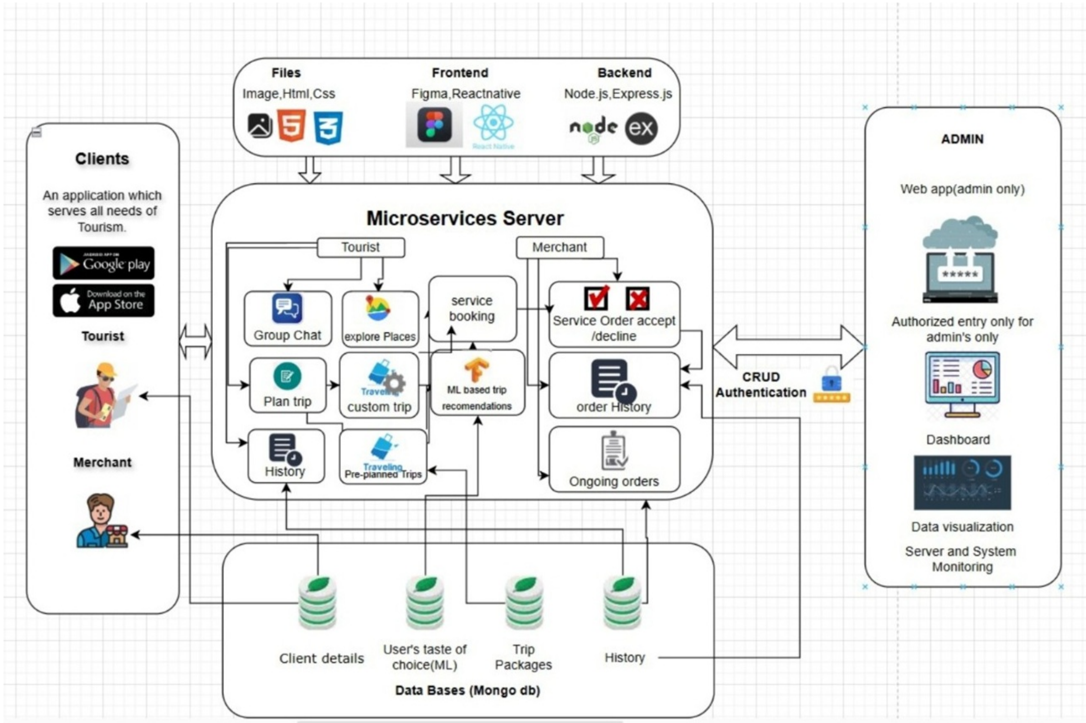

Fig. 4. Architecture Diagram of ExploreEase

NMF is then employed to learn the latent user-place interaction embeddings from the clustered dataset [12]. This is seen in Fig. 5.

It then predicts place preferences for new users and groups by analyzing their historical visits, friend network structure, and interaction strengths together, as well as when users interact with places outside their typical cluster or preference boundaries.

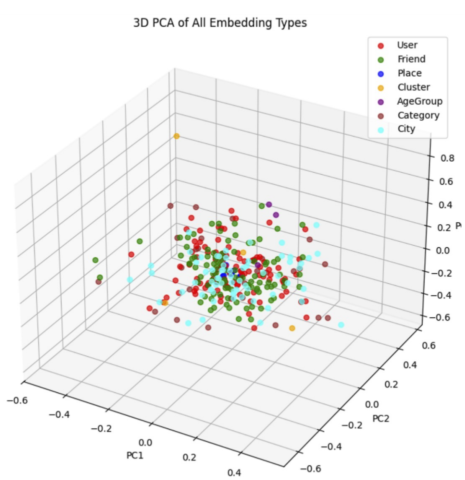

Fig. 5. 3D PCA of All Embedding Types

Reinforcement Learning [13] updates recommendations over time based on the user's feedback.

To integrate the impact of clustering, matrix factorization, and reinforcement updates, a Unified Recommendation Score (URS) is formulated for suggesting a location p to a user u:

$$\mathrm{URS}(u,p) = \alpha \cdot C_u(p) + \beta \cdot N_u(p) + \gamma \cdot R_u(p) \quad (1)$$

where $C_u(p)$ is the Cluster Consistency Score (measuring how well the location is consistent with the user's cluster), $N_u(p)$ is the NMF-based preference score and $R_u(p)$ is the reinforcement-based update from recent action.

The weights $\alpha, \beta, \gamma \in [0, 1]$ are tunable to regulate each component's relative contribution.

Model performance is measured in terms of Silhouette Score for cluster effectiveness and RMSE, MAE, and Hit Rate for recommendation accuracy.

B. Route Optimization Module

This module provides three different types of route suggestions: time-optimized, cost-optimized, and scenic [4]. The time-optimized route will rely on the Google Maps application programming interface (API) and use real-time distance and traffic data [7] [15]. The cost-optimized route will consider the estimated fuel usage, mode of transportation prices (for example, public transit or shared vehicles), and the likelihood of ride-sharing [10].

The scenic route will be produced using a modified Dijkstra's or A* algorithm and must assign meaningful weights to each edge of the graph (travel time T and scenic value S). The scenic value is calculated using the photo metadata, elevation data, and user reviews.

The edge weight is defined as:

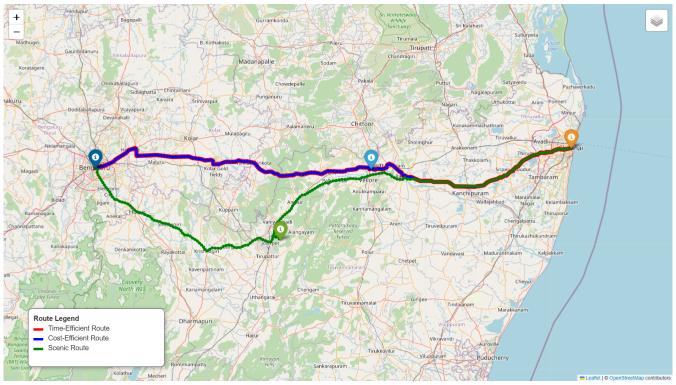

Fig. 6. Map showing three route suggestions: time-efficient (red), cost-efficient (blue), and scenic route (green)

$$W_{\text{edge}} = \alpha \cdot T - \beta \cdot S \tag{2}$$

with α and β being tunable parameters, where every parameter is in the range from 0 to 1, with it indicating the user's preferences of balancing the two. To ensure the scenic route is not inefficient, the total travel time will be capped at no more than 30% additional time on the shortest path.

The implementation relies on the Google Maps API for time and cost routes and uses Python's NetworkX package for customized graph traversals. Scenic information comes from OpenStreetMap, Google Places, and crowdsourced tags.

Consider a small trip from Chennai to Bangalore. The route optimization module suggests three routes as shown in Fig. 6.

C. Other Supporting Modules

ExploreEase includes other modules like Dynamic Trip Packages (created using seasonal patterns and historical user interest patterns), Hotel & Transport Booking (using third-party APIs like Skyscanner, Redbus, and IRCTC), Real-time communication, Expense Tracker for group expenses,and Trip Album Manager (uses EXIF metadata, time-based sorting, and location clustering). These modules together complete all the features needed for a super-app.

D. Implementation and Integration Details

The front-end was developed as a React Native mobile app providing fluid cross-platform UI and native performance. The back-end is built on Node.js, allowing API endpoint creation, business logic, database functionality, and easily scalable request management. Machine learning components are built on Python and integrated with the back-end using RESTful APIs. This enables model training, inference, and data pre-processing without impacting site/user performance, and easy portability on separate environments. The system is built with session and secure authentication, and services with specified REST API endpoints are used for inter-module communication, which frees each environment to run on separate maintainable processes.

V. Experimental Results

A. Trip Recommendation module

The recommendation engine was assessed against conventional metrics to rate predicted user-place interaction performance. The performance across a variety of trials was consistent and is outlined below in Table I

TABLE I
RECOMMENDATION SYSTEM PERFORMANCE METRICS

Metric	**Value**
Test Loss	0.0275
RMSE	0.0008
MAE	0.1344
R^2 Score	0.9711

This results in an overall prediction performance accuracy of 97.11%. This in turn substantiates our model combining K-Means Clustering, Neural Matrix Factorization and Reinforcement Learning worked effectively in our integrations.

B. Route Optimization module

We assessed the ExploreEase route phase on a small trip between Chennai and Bangalore. ExploreEase generated three diverse routes, each with differing priorities, Priority 1-3 as shown in the visualisation. Table II.

TABLE II
CHENNAI TO BANGALORE: ROUTE OPTIMIZATION OUTPUTS

Route Type	**Time & Distance**	**Key Feature**
Time-Efficient	5 hr 30 min (347 km)	Fastest via NH 48 with live traffic
Cost-Efficient	6 hr 15 min (325 km)	Public transport (train + bus) for low cost
Scenic Route	6 hr 45 min (360 km)	Through Yelagiri Hills and forest regions

The scenic-aware routing algorithm balanced travel time against the coherent visual score using a modified Dijkstra/A* with a restricted travel time (maximum of 30% travel time over the shortest route).

C. Baseline Comparison

For validation purposes, we evaluated ExploreEase against two common recommendation approaches shown in Table III.

TABLE III
BASELINE COMPARISON OF RECOMMENDATION MODELS

Method	**RMSE**	**MAE**	**R^2 Score**
ExploreEase (Ours)	0.0008	0.1344	0.9711
Naive Average Recommender	0.1510	0.2329	0.6124
Traditional CF (KNN-based)	0.0923	0.1871	0.7432

ExploreEase consistently exceeds these approaches on all evaluation metrics, confirming its ability to generate highly personal, contextual recommendations.

D. Discussions

The model's excellent $R^2(0.9711)$, and RMSE(0.0008) indicate the user preferences were predicted based on social data using reinforcers learning accurately, in addition to finding the best travel scenario for group travel. The practicality of the Route Optimization Module implemented in Chennai–Bangalore, demonstrated the ability to create the most efficient, economical and scenic routes that enhanced user travel satisfaction. Lastly, while cold-start and subjective scenic ratings present challenges for new users, ExploreEase offers a one-stop, intelligent, and thoughtful travel option that provides individualized and/or group travel experiences.

VI. Conclusion

This paper presents ExploreEase, a smart AI-powered super-app that changes the way people plan trips. This unique method utilizes K-means clustering, neural matrix factorization, and reinforcement learning algorithms. The different functionalities in the architecture of ExploreEase offer real-time personalized and group recommendations while optimizing the route and budget. The experiment with users provided a clear distinction in the level of accuracy of place recommendations. ExploreEase is an ideal app for trips and is focused on the places that other apps often fail to fulfill, being a solid foundation for the future development of innovations within smart tourism systems.

The authors declare that there are no conflicts of interest concerning this work. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

VII. Future Works

ExploreEase can be enhanced in the future by improving the accuracy of destination identification through the extension of its world place database and the use of more advanced calculation power. This will also allow it to include many more destinations with greater accuracy. Also, by including blockchain technology for traveling records and user identification can allow users to safe and dependable experience. Adding a voice -based virtual assistant, is another important direction to develop the project. This would allow users to engage and communicate with the app in natural language, therefore allowing users to easier and more convenientlyplan and manage trips while travelling.

References

[1] V. Parikh, M. Keskar, D. Dharia, and P. Gotmare, "A Tourist Place Recommendation and Recognition System," in *Proc. 2018 Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Coimbatore, India, 2018, pp. 218–222, doi: 10.1109/ICICCT.2018.8473077.

[2] J. K. Kim, W. C. Kwon, I. Y. Choi, H. Heo, and H. S. Moon, "A Group Travel Recommender System Based on Group Approximate Constraint Satisfaction," *IEEE Access*, vol. 12, pp. 96113–96125, 2024, doi: 10.1109/ACCESS.2024.3427122.

[3] Y. Kong, K. Yi, L. Wang, C. Peng, L.-M. Nguyen, and Q. Ma, "RPMTD: A Route Planning Model With Consideration of Tourists' Distribution," *IEEE Access*, vol. 12, pp. 69488–69504, 2024, doi: 10.1109/ACCESS.2024.3400373.

[4] Y. Zhang, L. Jiao, Z. Yu, Z. Lin, and M. Gan, "A Tourism Route-Planning Approach Based on Comprehensive Attractiveness," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2967060.

[5] A. Sharaff, S. Khurana, K. Cheepurupalli, and T. Sahu, "Personalized Recommendation System with User Interaction based on LMF and Popularity Model," in *Proc. 2020 Int. Conf. Syst. Comput. Autom. Netw. (ICSCAN)*, Pondicherry, India, 2020, pp. 1–6, doi: 10.1109/ICSCAN49426.2020.9262442.

[6] T. Alenezi and S. Hirtle, "Normalized Attraction Travel Personality Representation for Improving Travel Recommender Systems," *IEEE Access*, vol.10, pp.56493–56503, 2022, doi: 10.1109/ACCESS.2022.3178439.

[7] P. Stefanovič and S. Ramanauskaitė, "Travel Direction Recommendation Model Based on Photos of User Social Network Profile," *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3260103.

[8] D. Qin, X. Zhou, L. Chen, G. Huang, and Y. Zhang, "Dynamic Connection-Based Social Group Recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 453–467, Mar. 2020, doi: 10.1109/TKDE.2018.2879658.

[9] L. Tang, X. Lin, J. Yu, and C. Tang, "Integrating Travel Survey and Amap API Data Into Travel Mode Choice Analysis With Interpretable Machine Learning Models: A Case Study in China," *IEEE Access*, vol. 13, pp. 27852–27867, 2025, doi: 10.1109/ACCESS.2025.3540082.

[10] Z. Jia-Xiang, M. Zhan-Xiao, and G. Yan, "Mixed Traffic Mode Based Travel Route Recommendation," in *Proc. 2018 Int. Conf. Netw. Inf. Syst. Comput. (ICNISC)*, Wuhan, China, 2018, pp. 372–376, doi: 10.1109/ICNISC.2018.00082.

[11] J.-H. Chang, C.-E. Tsai, and J.-H. Chiang, "Using Heterogeneous Social Media as Auxiliary Information to Improve Hotel Recommendation Performance," *IEEE Access*, vol. 6, pp. 42647–42660, 2018, doi: 10.1109/ACCESS.2018.2855690.

[12] M. Li, H. Wu, and H. Zhang, "Matrix Factorization for Personalized Recommendation With Implicit Feedback and Temporal Information in Social Ecommerce Networks," *IEEE Access*, vol. 7, pp. 141268–141276, 2019, doi: 10.1109/ACCESS.2019.2943959.

[13] J. Jia and W. Wang, "Review of reinforcement learning research," in *Proc. 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Zhanjiang, China, 2020, pp. 186–191, doi: 10.1109/YAC51587.2020.9337653.

[14] K. Wang and F. Tsung, "Sparse and Robust Multivariate Functional Principal Component Analysis for Passenger Flow Pattern Discovery in Metro Systems," in *IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7*, July 2022, pp. 8367-8379, doi: 10.1109/TITS.2021.3078816.

[15] O. Yilmaz, L. Görgü, M. O'Grady, and G. O'Hare, "Cloud-Assisted Mobile Crowd Sensing for Route and Congestion Monitoring," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3129932.