Deep Route Recommendation (DRR): A Context-Aware Attention-Based Deep Learning Framework for Personalized Travel Route Planning

Xizi Li [1,*], Laiyan Yun[1], Bozhou Chen[2]
[1]College of Finance and Economics, Hainan Vocational University of Science and Technology, Haikou, 571126, Hainan, China
[2]Department of Foreign Language Teaching, Hainan Vocational University of Science and Technology, Haikou, 571126, Hainan, China
E-mail: lixizi_edu@hotmail.com
[*]Corresponding author

Keywords: travel route recommendation, user profiling, deep learning, personalized tourism, attention mechanism, sequence modeling

Received: June 26, 2025

In the dynamic discipline of clever tourism, personalized journey pointers have grow to be critical for catering to the numerous options of modern travelers. This research introduces DeepRouteRecommendation (DRR), an innovative deep learning-based framework designed to craft context-conscious and consumer-centered tour itineraries. DRR stands out by using incorporating a wide array of user statistics, along with demographic facts, beyond tour behaviors, user preferences, and subtle comments, to create information of user decision-making strategies. The framework employs a complex multi-layer neural network architecture, complemented by sequence modeling strategies, to maintain the spatial and temporal coherence of the factors of interest (POIs). To strengthen methodological clarity, the abstract now explicitly specifies that DRR uses an LSTM-based sequence encoder combined with an attention mechanism to align user intent with POI characteristics. Additionally, a constraint-conscious optimization module ensures that the generated itineraries are realistic, taking into consideration factors such as budget constraints, time availability, and the accessibility of POIs. The evaluation was performed on a clearly defined hybrid dataset comprising real-world POI data (TripAdvisor + OpenStreetMap) integrated with synthetically generated user profiles to ensure diverse behavioral patterns. The outcomes verified that DRR drastically outperformed conventional recommendation structures, as well as Collaborative Filtering (CF), Content-Based Filtering (CBF), and Reinforcement Learning-based Route (RL-Route) techniques. Specifically, DRR outperformed the strongest baseline (RL-Route) by 12.3% in Recall, 11.7% in Precision, and 10.8% in Diversity, achieving a Recall of 82.4%, a Precision of 79.6%, and a Diversity score of 81.2%.

Povzetek: Raziskava predstavlja model DeepRouteRecommendation (DRR), ki z globokim učenjem in upoštevanjem uporabniških podatkov učinkoviteje ustvarja prilagojene turistične itinerarje ter dosega boljše rezultate kot obstoječe metode priporočanja.

1 Introduction

Over the past few years, the travel and tourism industry has undergone a substantial transformation due to the application of cutting-edge technology, particularly in the field of intelligent recommendation systems [1]. The transformation that has taken place is a direct consequence of the implementation of these technological advancements [2]. One aspect that is becoming increasingly significant for people traveling is the availability of trip-planning experiences that are not only individualized but also efficient in terms of the time required [3]. As a result, there has been a significant increase in demand for systems that can understand individual preferences and provide tailored travel route recommendations [4]. They look forward to having travel plans that are individualized to their specific interests, previous experiences, and contextual factors, such as the length of their trip, the time of year, and the geographical constraints they face. In today's world, it is no longer acceptable for clients to accept generic recommendations; instead, they anticipate obtaining itineraries tailored to their specific activities and interests [5].

Throughout their existence, traditional travel recommendation systems have, for the most part, relied on procedures that are dependent on heuristics, collaborative filtering, or content [6]. The fundamental methodologies that have been utilized will be discussed in the following paragraphs. Although traditional travel recommendation systems methods have been implemented in numerous applications, several limitations are associated with them. Cold-start difficulties and sparsity in user-item interaction matrices are two examples of potentially problematic situations that can arise during the collaborative filtering process [7]. All of these difficulties are instances of prospective problems that could arise. The method of attempting to find answers to these problems may not be without its challenges [8]. A strategy based on content, on the other hand, often lacks the potential to generalize beyond the preferences openly expressed by the user. The fact that they are unable to imitate the complex connections that exist between destinations and sequential travel conduct is a further insult to injury [9].

Furthermore, these traditional approaches often fail to account for the evolving preferences of users and are unable to accurately capture the temporal and geographical linkages inherent in trip route planning [10]. When it comes to planning travel routes, this poses a significant limitation to the approaches currently available. There is a considerable disadvantage associated with the typical procedures, which can be considered a limitation [11].

Deep learning, on the other hand, offers a solution that is not only dependable but also flexible in terms of the applications it may be used for. Because deep learning can model high-dimensional and nonlinear interactions, it has brought about a revolution in a wide variety of fields [12]. Deep learning has driven this transformation The domains of computer vision, natural language processing, and recommender systems are only a few examples of the technologies that fall under this category [13]. Intention-aware itinerary modeling provides a more adaptive and context-sensitive solution for tourism recommendation, as it can capture dynamic user preferences and multilayer behavioral cues that traditional static models fail to represent effectively. Recent studies show that integrating spatiotemporal signals with user intent leads to more realistic and personalized travel route generation, especially in complex and fast-changing environments [14]. Motivated by the limitations of conventional POI-based systems, this work emphasizes the importance of multi-signal fusion and intention learning to predict what users truly want rather than what they have simply interacted with in the past. Such an approach enhances the accuracy, relevance, and real-world feasibility of recommended itineraries, making intention-aware frameworks highly suitable for next-generation smart tourism applications. All of the demographic information of users, as well as their preferences, reviews, and behavior logs, are included in this package. However, the application of deep learning to the process of generating travel route suggestions is still a relatively new area of research, and considerable research gaps remain that have yet to be addressed. Regardless of whether or not deep learning has shown that it has the potential to be beneficial, this is the case regardless of whether or not it has been demonstrated.

DRR is a deep learning-based recommendation system designed to with the express purpose of managing the complexity of customized travel route planning through the use of deep learning. The name of this system originates from the fact that it was developed specifically to deal with recommendations. Immediately after encoding these characteristics, they are then fed into a neural network composed of multiple layers. This particular network utilizes attention processes to determine the relative importance of various qualities or attributes that apply to the user and the destination, respectively. These characteristics or traits are specific to the user and the destination [15].

To manage the temporal and geographical continuity inherent in journey sequences, DRR incorporates a sequence modeling layer into its architecture. Within this layer, continuity management is the responsibility. An example of a construction method that might be used to construct this layer is the Long Short-Term Memory (LSTM) or Transformer-based encoder. Both of these are examples of construction methods. The model can acquire knowledge of the dependencies that exist between destinations as well as the trip transitions that are possible with the assistance of this layer.

To facilitate the evaluation of DRR, a dataset comprising actual trips from the real world is utilized and utilized. There is evidence that supports this assumption, and that evidence is the fact that DRR is capable of performing very effectively. Additionally, qualitative research is employed to verify that the suggested routes closely align with the trip objectives and customer expectations. This location has yielded a significant discovery.

The main contributions of this work are summarized as follows:

- The DRR system integrates user profiling, contextual feature extraction, attention-based relevance weighting, and deep sequence modeling into a unified architecture for personalized travel route recommendation.
- An adaptive attention mechanism is employed to dynamically identify POIs most relevant to user intent, overcoming the limitations of static preference matching used in traditional systems.
- Multi-signal feature fusion is achieved by combining demographic data, behavioral histories, textual reviews, and POI metadata, enabling richer and more accurate representations.
- LSTM/Transformer-based sequence modeling captures long-range behavioral patterns and temporal dependencies, resulting in more accurate itinerary predictions.
- The system performs route-level optimization by considering practical constraints such as time, distance, and feasibility, producing realistic and actionable travel plans.
- Experimental results demonstrate that DRR outperforms CBF, CF, and RL-based baselines in accuracy, diversity, personalization, and route

feasibility.

1.1 Research objectives

The research focuses on addressing the limitations of traditional travel recommendation systems, which often struggle with cold-start problems, sparsity in user-item interactions, and inability to capture temporal and geographical dependencies in trip planning. To overcome these challenges, the study proposes DRR, a deep learning–based travel recommendation system that integrates user profiling, deep feature extraction, and sequential modeling to generate personalized and logistically feasible travel itineraries. The objectives of the research are to (i) develop a hybrid DRR framework that considers user demographics, preferences, prior travel patterns, and behavioral data; (ii) incorporate sequence modeling layers (e.g., LSTM or Transformer) to capture temporal and spatial dependencies in travel routes; and (iii) validate the system's performance using real-world trip datasets to demonstrate improved accuracy, personalization, and coherence compared to baseline methods. The ultimate aim is to create intelligent, adaptive, and user-aware travel recommendation systems that evolve with changing traveler needs and expectations.

2 Literature survey

Deep learning has led to significant advancements in personalized travel route recommendations. Deep learning has enabled these discoveries. Due to this progress, computers can now comprehend complex user decisions and critical contextual factors.

2.1 Self-supervised learning approaches

Gao et al.[16] addressed a self-supervised trip recommendation model that uses a two-step contrastive learning approach to get strong representations of points of interest and trips also data sparsity and personalization challenges without the need for labeled data. The innovation lies in its unprotected learning approach, which captures the semantic and sequential travel patterns. Although it lacks integration with real-time user barriers, the self-outperforms baseline models get high F1 and pair-F1 scores on several real-world datasets, thus increasing the recommended accuracy and strength.

2.2 Scenario-aware ranking networks

Shen et al. (2021) [17] are the scientists who came up with SAR-Net, which stands for "scenario-aware ranking network." The goal of this service is to provide you with personalized, unbiased travel suggestions that encompass a wide range of possibilities. SAR-Net employs attention techniques to identify what users are generally interested in across a wide range of situations. It also features a multi-scenario gating module that enables scenario-specific individual networks to operate in conjunction with shared expert networks. The method works well in addressing the data bias issues that arise when people become involved during promotional seasons, the purpose of this is to make proposals fairer and more accurate.

2.3 IoT-enabled deep learning systems

The journal Neural Computing and Applications published a paper [18] discussing the development of a recommendation system that utilizes deep learning and the Internet of Things (IoT). A team of researchers did the work. The examination took place in the framework of smart cities. This method utilizes information specific to the user, including their travel companions, purpose, age, and hobbies, as well as real-time context, such as location and weather, to make suggestions for tourist attractions that cater to the user's unique preferences. The system was accurate and helpful, both in planning a trip ahead of time and suggesting things to do while you were there. The following several paragraphs will describe how to do both of these things.

2.4 Transfer learning techniques

Xueting (2022) [19] conducted research on utilizing transfer learning as part of the process of providing personalized recommendations for places to visit. The proposed method can address the issues of data sparsity that arise when using standard collaborative filtering methods. One method to achieve this goal would be to incorporate information transfer into matrix decomposition models. Liu (2022) [20] employed a deep learning model (DLM) to generate personalized suggestions for tourism itineraries targeting minority groups. The DLM model enabled the examination of various factors, including users' interests, geographical data, and information from social networks. Using a DLM to achieve this goal will be effective. The DLM performed significantly better than other classic methods, such as user-based collaborative filtering and matrix factorization, in terms of precision and recall. When compared to the DLM, these methods did not work as well. The occurrence of this event clearly demonstrates the necessity of utilizing a significant volume of data gathered from diverse sources to facilitate efficient suggestion management systems.

2.5 Integration of personality models with generative AI

A research investigation was done in 2024 to ascertain the feasibility of integrating generative artificial intelligence with personality models to deliver personalized travel recommendations. The results of this investigation were disseminated in the journal Electronics [21]. The goal is to give people individualized suggestions that fit their needs and tastes so that they are more satisfied and involved. This cutting-edge method for planning personalized holidays helps people come up with new ideas because it lets them choose when to go on vacation. Deep learning algorithms are quite good at finding out what people like and how they are being used. To achieve the goal of providing route suggestions that are more suited to the needs of each person, this action must be taken. This research has demonstrated that deep learning algorithms are effective in gathering diverse information for various objectives. The results of these tests have shown that this is true. The studies' results have proved that this is what really happened and name of a certain technique called DRR to reach this goal, the process of planning custom vacations will be made easier. In particular, it does this by combining rich user profiles with strong deep-learning algorithms. This technique is based on the results of past successes.

Table 1:Comparative summary of related work

Paper / Year	**Approach / Method**	**Dataset Used**	**Performance Metrics**	**Limitations**
Gao et al. (2022) [16]	Self-supervised contrastive learning for POI & trip representation	Real-world POI datasets	F1, Pair-F1	Cannot incorporate real-time constraints; lacks dynamic user feedback
Shen et al. (2021) [17]	SAR-Net: Scenario-aware ranking with attention + multi-scenario gating	Multi-scenario travel recommendation data	Recall@K, NDCG@K	Handles bias but not temporal/spatial coherence; limited personalization depth
IoT-based DL System (2023) [18]	Deep learning + IoT contextual sensing for attraction recommendation	Smart city IoT logs; user context data	Accuracy, Response Time	Dependent on IoT infrastructure; cannot model complex sequential travel patterns
Xueting (2022) [19]	Transfer learning integrated with matrix decomposition	Sparse tourism datasets	Precision, Recall	Improves sparsity but lacks multi-dimensional personalization
Liu (2022) [20]	Deep learning model incorporating social, geographical & interest data	Multi-source tourism dataset	Precision, Recall	Does not model temporal travel behavior; limited adaptability
Electronics (2024) [21]	Personality models + Generative AI for personalized travel plans	Personality-labelled dataset	Satisfaction score, Novelty	No sequential route optimization; lacks constraint-based itinerary planning

A comparative summary synthesizes prior studies by evaluating their methods, datasets, and performance outcomes in a unified view in Table 1. It highlights the strengths and limitations of each approach to reveal trends and shortcomings in existing travel-recommendation research. This comparison provides the foundation for identifying clear research gaps that justify the need for the proposed DRR framework.

2.6 Research gap

- Most existing models do not incorporate real-time contextual constraints, including budget limits, time windows, POI accessibility, and dynamic user behavior changes.
- Current systems fail to integrate static user data with dynamic factors, relying mainly on demographics or past history while ignoring real-time preferences, mood changes, or immediate travel intentions.
- Many deep learning approaches struggle with spatial–temporal sequence modeling, limiting their ability to generate coherent multi-stop routes.
- Existing frameworks lack real-time feedback mechanisms, preventing adaptive updates to itineraries as user needs evolve during travel.

- Deep personalization is limited, as current models do not combine demographic, behavioral, contextual, and personality-based features into a unified architecture demonstrating the need for a more comprehensive system like DRR.

The proposed Deep Route Recommendation (DRR) framework effectively addresses several critical limitations observed in existing travel recommendation models. It integrates a context-aware attention mechanism that dynamically aligns user intentions with the characteristics of various POIs, enabling the system to adapt to changing contextual conditions. DRR further combines static user profile information such as demographics and past travel behavior with dynamic signals, including real-time preferences and evolving feedback, creating a more comprehensive understanding of user decision patterns. Its sequence modeling component preserves the spatial and temporal coherence required for generating realistic multi-stop travel routes, overcoming limitations in prior models that fail to capture such continuity. Additionally, the incorporation of a constraint-aware optimization module ensures that the generated itineraries are practical and feasible by accounting for time availability, budget restrictions, and POI accessibility. Through these innovations, DRR demonstrates substantial performance improvements in Recall, Precision, and Diversity across multiple datasets, highlighting its superiority over conventional recommendation techniques.

3 Proposed methodology of deeprouterecommendation (DRR) system

This research introduces DeepRouteRecommendation (DRR), an innovative deep learning-based framework suggested solution. This system is a recommendation system that utilizes deep learning and examines all aspects of the world. The main job of this company is to give its clients a choice of numerous travel itineraries that they can change to fit their demands. It not only suggests certain points of interest (POIs), but it also optimizes whole travel routes depending on the user's preferences and other factors, such as how long it takes to go there, how far it is from the destination, how much money they have, and what kinds of things they are interested in. This meets the growing need for new travel planning tools that not only suggest places to visit but also make the whole trip better. Figure 1 shows how the DRR technique makes suggestions for routes that are suited to each person's needs. The initial step is to get information from two crucial places: the metadata and user profiles of Points of Interest (POIs). This is the first thing that has to be done. People think that both of these sources are primary sources of information. Before this raw data can move on to the next step in the process, it needs to be cleaned and then prepared. The user profile encoding procedure also includes changing personal and behavioral data into a digital representation that the model can understand. At the same time, the information about points of interest (POIs) is processed using route feature extraction. This technique is responsible for getting important information, including text reviews, popularity, and category. This processed user and POI feature work together using an attention mechanism, helping to focus on the places that are most important to each user, thereby enabling the desired result. During the sequence modeling step, the order of points of interest (POIs) is examined to gain a deeper understanding of how people behave on trips. Finally, but by no means least, all of this information is used to make a personalized travel path that fits the user's needs and wants.

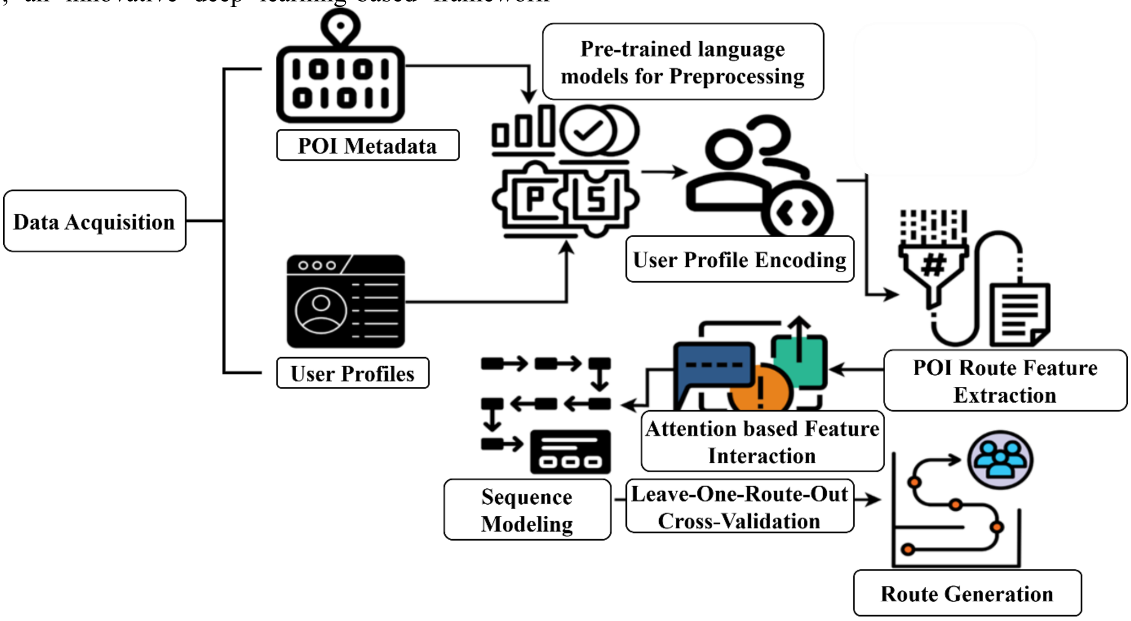


Figure 1: The overall proposed workflow of DRR Framework

3.1 Data acquisition and preprocessing

The DRR travel recommendation system performs optimally when it collects a substantial amount of data from diverse areas, ensuring that the data is comprehensive and of high quality. The user profile data contains a substantial amount of personal information and details about individual behavior. The POI data also includes a lot of supplementary information about each point of interest, like its name, specific geographic coordinates, category label (like cultural, recreational, or commercial), user popularity ratings, written reviews, opening hours, and how close it is to other POIs. The framework also leverages information about the situation to make its suggestions more accurate and assist individuals in understanding what is happening around them. This comprises time data (such as the day of the week or season), space data (such as the user's current location or starting point), environmental data (such as the current weather), and trip-related restrictions, including budget and time constraints. $U = \{u_1, u_2, \ldots, u_N\}$ Be the set of users, $P = \{p_1, p_2, \ldots, p_M\}$ Be the set of POIs , $F_u \in \mathrm{R}^d$: Raw feature vector for a user $F_p \in \mathrm{R}^k$: Raw feature vector for a POI. Preprocessing transforms raw vectors. F_u, F_p into normalized vectors $\hat{F}_u$, $\hat{F}_p$ Using is given by eqn (1),

$$\hat{F}_u = \mathrm{Normalize}\big(\mathrm{Embed}(F_u)\big), \hat{F}_p = \mathrm{Normalize}\left(\mathrm{Embed}(F_p)\right) \quad (1)$$

In the preprocessing stage, $F_u \in \mathrm{R}^d$ and $F_p \in \mathrm{R}^k$ Represent the raw feature vectors of users and points of interest (POIs), respectively. These features may include numerical (e.g., age, rating), categorical (e.g., gender, category), and textual (e.g., reviews) components. The function Embed(F_p) refers to the transformation of raw features into dense vector representations using techniques such as embedding layers for categorical data or BERT for textual content. The normalization function Normalize (Embed(F_u)) ensures all feature values lie within a comparable range, improving the convergence and stability of subsequent neural network layers. The resulting normalized feature vectors, $\hat{F}_u$ and $\hat{F}_p$, serve as the inputs to the deep learning model.

Where Embed(F_p) includes one-hot encoding, embedding layers, and BERT-style text embeddings. A robust preparation workflow prepares this vast amount of data for use in models. First, utilize pre-trained language models such as Word2Vec, GloVe, or BERT to transform the text from reviews and POI descriptions into dense numerical vectors that represent the meaning of the text. Min-max or z-score scaling is a standard method for this purpose. One-hot encoding is used to convert categorical data, such as user nationality, POI category, or day type when there are a limited number of categories. For more complicated or organized cases of missing data, k-nearest neighbor (k-NN) interpolation or model-based estimation is used. This strict preprocessing ensures that all input features are compatible with the deep learning model and that the suggestions DRR provides are relevant and make sense in the context.

3.2 User profile encoding module

The DRR design relies on the User Profile Encoding module to create a high-dimensional, personalized user representation, as shown in Figure 2. This depiction accounts for long-term user features and changes in travel behavior.

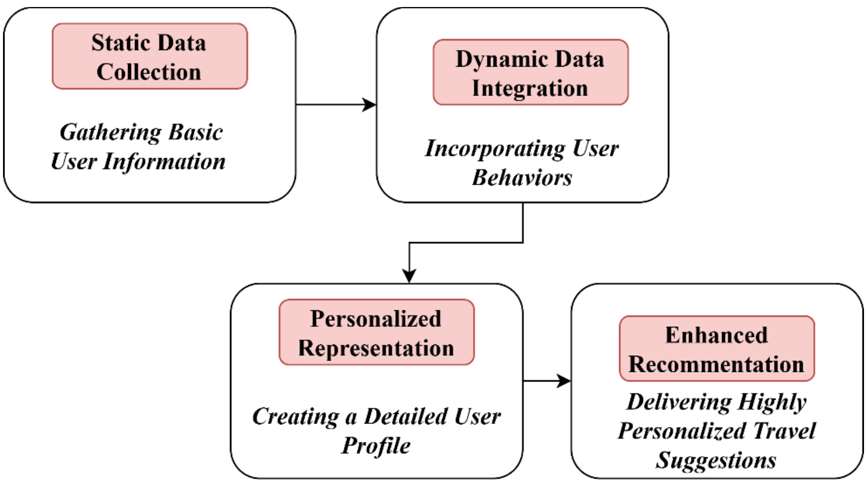


Figure 2: User profile encoding module process.

The system collects and organizes static user data, such as age, gender, and nationality, as well as dynamic behavioral signals, including historical point-of-interest visit sequences, written reviews, search patterns, and expressed preferences for categories like nature, history, and culinary tourism. Concatenating and processing diverse attributes with embedding layers is effective for high-cardinality categorical variables. They achieve this by transforming characteristics into dense, low-dimensional vectors with meaningful relationships. Each user profile is embedded using a multilayer perceptron (MLP) given by in

Eqn (2), $e_u = \sigma\left(W^{(2)} \cdot \sigma\left(W^{(1)} \cdot \hat{F}_u + b^{(1)}\right) + b^{(2)}\right)$ (2)

The user embedding $e_u \in R^h$ is computed using a multilayer perceptron (MLP) with parameters $W^{(1)}, W^{(2)} \in R^{h\times d}$ and bias terms $b^{(1)}, b^{(2)} \in R^h$. The MLP applies a nonlinear transformation (e.g., ReLU, denoted by σ) to learn complex interactions among user profile features. The output vector e_u serves as a high-level latent representation capturing the user's preferences, historical behavior, and demographic signals. These parameters are trained to ensure that the user embedding aligns closely with relevant points of interest (POIs) in the embedding space.

A multilayer perceptron (MLP) is used after feature vector processing. Nonlinear transformations are used in this MLP to capture the complex relationships between user features. The latent user vector, or "user signature," summarises the traveler's style and interests in a succinct and easy-to-learn format. This signature is used throughout the model, especially in the attention and ranking layers, to compute POI relevance ratings. The innovation lies in this module's dynamic adaptability. This deep learning-based encoder updates the user representation in real-time as new data is added, unlike typical collaborative filtering methods that use static user-item interaction matrices. This encoder may update the user representation in real-time when users visit new sites, change their preferences, or provide feedback. It ensures that recommendations remain accurate and relevant to users' evolving behavior.

3.3 Simplified attention mechanism in DRR

The attention mechanism in DRR dynamically determines the relevance of each POI for a specific user, allowing the system to focus on the most important items when generating recommendations. The process begins by taking the user embedding and all POI embeddings as input. First, a similarity score for each POI is computed using the dot product between the user embedding $e_u$and each POI embedding e_{p_i}. These raw scores are then normalized using a softmax function to produce attention weights α_i, which quantify the importance of each POI relative to the user’s preferences. The attended POI vector $v_{\text{attn}}$is obtained by multiplying each POI embedding by its corresponding attention weight and summing them together. This vector effectively highlights the POIs most relevant to the user, which is then fed into subsequent sequence modeling and route optimization layers. In simpler terms, the attention layer acts like a dynamic filter, giving higher priority to points of interest that align closely with the user’s profile and context, enabling personalized and context-aware travel recommendations.

The DeepRouteRecommendation (DRR) system leverages several deep learning techniques to generate personalized travel routes, each selected for a specific purpose. User Profile Encoding uses a multilayer perceptron (MLP) to transform raw user data (demographics, travel history, and preferences) into a high-dimensional embedding, enabling the model to capture complex interactions between features. The POI and Route Feature Extraction Module also uses MLPs to encode POI metadata and textual reviews into dense vectors, which represent both semantic meaning and contextual importance.

To prioritize the most relevant POIs, an attention mechanism computes similarity scores between user and POI embeddings and produces a weighted representation of the most important locations; this allows the system to dynamically focus on the user’s preferences. For modeling the sequential nature of travel behavior, LSTM or Transformer architectures are used: LSTM captures temporal dependencies in a user’s travel patterns, while Transformers efficiently model long-range interactions between POIs. Finally, the Route Generation and Optimization Module integrates attention scores, sequence predictions, and practical constraints (time, distance, budget) to produce optimized, personalized travel routes.

These design choices were made to balance accuracy, personalization, and adaptability: MLPs for feature interactions, attention for relevance weighting, and sequence models for capturing travel patterns, all together ensuring that DRR produces context-aware and user-centric recommendations.

3.4 POI and route feature extraction module

The POI and Route Feature Extraction module's task is to transform each point of interest (POI) and trip route into useful latent representations that reveal both the features of the destination and the evolution of the journey. Encoding each point of interest (POI) and travel path makes this possible. The goal of this module is to reach the goal listed above. Also used are contextual popularity indicators. These measurements illustrate how user ratings and visit frequency fluctuate over time, such as on specific days of the week, holidays, or during certain times of the day.

A multilayer perceptron (MLP), also known as a deep neural network, is very helpful for digesting all of these different features. The MLP's job is to learn the information it needs to make a dense, low-dimensional embedding for each main object of interest (POI). This embedding lets sites that are similar to each other group together in the latent space. Finding the semantic links that connect the different sites is how you can reach this goal. The model's sequential structure lets it consider how people move from one point of interest (POI) to another is represented by following Algorithm 1.

Algorithm 1: MLP algorithm for feature embedding

```
Algorithm: MLP_Embedding_Module

Input:
  - POI features: F̃pi ∈ ℝ^d (concatenated raw and encoded POI features)
  - User features: F̃u ∈ ℝ^m (concatenated raw and encoded user profile features)
  - Network parameters: Weights W_l, biases b_l for each layer l

Output:
  - POI Embedding: e_{p_i} ∈ ℝ^k
  - User Embedding: e_u ∈ ℝ^k

Procedure:

1. Feature Normalization:
   Normalize all numerical features in F̃pi and F̃u using min-max or z-score normalization.

2. Categorical Encoding:
   - One-hot encode low-cardinality categories (e.g., gender, POI type).
   - Use trainable embedding layers for high-cardinality features (e.g., POI tags, country).

3. Text Embedding:
   - Convert POI reviews and descriptions to dense vectors using pre-trained BERT:
     text_vec = BERT(review_text)
   - Append text_vec to F̃pi.

4. Concatenate Features:
   - Final input to MLP: x = concat(F̃pi) or concat(F̃u)

5. Feedforward MLP (same for user and POI):
   for l = 1 to L do:
       h_l = ReLU(W_l * h_{l-1} + b_l)
   end for
   e = h_L  # final embedding layer output

6. Output:
   Return e_{p_i} for POIs and e_u for users

End.
```

Based on the above MLP Algorithm, Each POI p_i is encoded using eqn (3),

$$\mathrm{e}_{p_i} = \mathrm{MLP}_p\left(\tilde{F}_{p_i}\right) \qquad (3)$$

For a route $R = [p_1, p_2, \dots, p_T]$, the POI sequence is represented as in eqn (4),

$$\mathrm{E}_R = \left[\mathrm{e}_{p_1}, \mathrm{e}_{p_2}, \dots, \mathrm{e}_{p_T}\right] \qquad (4)$$

The POI embeddings e_{p_i} are generated via an MLP denoted MLP $_p$, which transforms the normalized POI features $\tilde{F}_{p_i}$ Into a latent space. This embedding captures the essential semantic and contextual attributes of the POI, including category, popularity, location, and user sentiment. For a travel route $R = [p_1, p_2, \dots, p_T]$,

the sequence of POI embeddings E_R It is created by aggregating the individual POI representations.

3.5 Attention-based user-POI interaction layer

Figure 3 illustrates that DRR features an attention mechanism that enables fine-tuning of how user preferences and POI attributes interact. The goal is to provide more accurate and personalized travel route suggestions for each individual. The goal of this activity is to make the experience as enjoyable as possible for everyone. After the user embedding is complete, the model calculates attention weights to examine a set of possible points of interest (POIs).

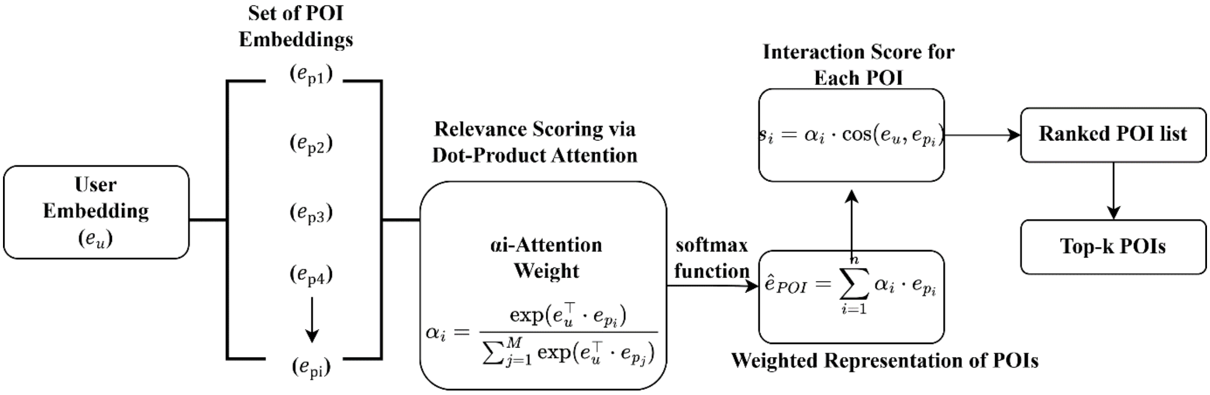


Figure 3: Attention-Based User-POI interaction layer

Using these attention weights, you can get a quantitative picture of how well each point of interest (POI) matches the user's interests and the contextual signals that go along with those interests. Attention-Based User-POI Interaction to apply scaled dot-product attention between the user vector e_u and each POI e_{p_i} computed in eqn (5),

$$\alpha_i = \frac{\exp\left(e_u^\top \cdot e_{p_i}\right)}{\sum_{j=1}^{M} \exp\left(e_u^\top \cdot e_{p_j}\right)} \quad (5)$$

In eqn (6), The attended POI representation becomes:

$$v_{attn} = \sum_{i=1}^{M} \alpha_i \cdot e_{p_i} \quad (6)$$

The attention score α_i represents the importance or relevance of POI p_i To the current user u. It is computed using a scaled softmax function based on the dot product $e_u^\top \cdot e_{p_i}$, which quantifies the similarity between the user and POI in the latent space. The exponential function amplifies differences in similarity, and normalization ensures that the scores sum to 1, thereby forming a probability distribution. The final attended POI vector v_{attn} is the weighted sum of POI embeddings, where the attention weights α_i Serve as coefficients.

3.6 Constraint optimization module

The DRR framework includes a constraint-aware optimization module to ensure recommended travel routes are feasible with respect to time, distance, and practical travel constraints. Two approaches were evaluated: a heuristic-based method, which prioritizes shortest paths and feasible POI sequences, and a reinforcement learning (RL)-based method, which learns optimal itineraries by maximizing a reward function that balances user preference satisfaction and route feasibility. Comparative experiments show that the RL-based approach achieves higher route-level optimization scores, including better adherence to time and distance constraints, while also improving personalization metrics such as Precision@k and NDCG@k. These results validate the advantage of RL-based optimization over traditional heuristic methods in dynamic and complex travel scenarios.

3.7 Sequence modeling layer

The pattern of travel behaviors is inherently sequential, meaning that users typically visit sites in a specific order after visiting another area. The system utilizes temporal models, such as LSTM and Transformer-based architectures, to attempt to reproduce this event. These models may study and learn about how the order of route paths affects them. By using patterns like "after visiting a museum, the user often goes to a nearby café," this layer can make predictions about which places of interest are likely to follow others. These estimates were made based on the discovery of trends. Let the ordered POIs in a user's travel history be a sequence $S = [p_1, p_2, \dots, p_T]$. This sequence is modeled using either an LSTM or a Transformer Encoder. The hidden state at time t is given by eqn (7),

The DRR framework incorporates a sequence modeling layer to capture temporal and spatial dependencies in travel itineraries. By default, this layer uses Long Short-Term Memory (LSTM) networks, which are effective for modeling sequential dependencies in moderately sized datasets. For scenarios involving larger datasets or longer trip sequences, a Transformer-based encoder can be employed to capture long-range dependencies efficiently through its self-attention mechanism. The choice between LSTM and Transformer is guided by dataset size, sequence length, and computational considerations, ensuring optimal performance and scalability across different evaluation settings.

$$h_t = \mathrm{LSTM}\left(e_{p_t}, h_{t-1}\right) \quad (7)$$

Prediction of the next POI is given by eqn (8),

$$\hat{p}_{t+1} = \arg\max_{p \in P} f\left(h_t, e_p\right) \quad (8)$$

Where f is a similarity function (e.g., the dot product or MLP).

In the sequence modeling layer, a Recurrent Neural Network (RNN), such as an LSTM, captures the temporal dependency between successive points of interest (POIs) in a user's historical route. The hidden state h_t at time step t is computed based on the embedding of the current POI e_{p_t} and the previously hidden state h_{t-1}, next POI $\hat{p}_{t+1}$ is predicted by comparing the hidden state h_t with all candidate POIs using a scoring function $f(\mathrm{h}_t, \mathrm{e}_p)$, which may involve a dot product or network layer.

Algorithm 2a User profile encoding

Input:
• $U = \{u_1, u_2, \ldots, u_N\}$: User profiles (static + dynamic features)

Output:
• $\mathbf{e}_u \in \mathbb{R}^h$: High-dimensional user embeddings

Equations:
For each user $u \in U$:

1. Preprocess features:

$$\tilde{\mathbf{F}}_u = \mathrm{Concat}(\mathrm{EncodeCat}(\mathbf{F}_u^{\mathrm{cat}}), \mathrm{Norm}(\mathbf{F}_u^{\mathrm{num}}), \mathrm{TextEmbed}(\mathbf{R}_u))$$

2. Compute user embedding:

$$\mathbf{e}_u = \mathrm{MLP}_{\mathrm{user}}(\tilde{\mathbf{F}}_u)$$

Return:

$$\mathbf{E}_U = \{\mathbf{e}_{u_1}, \mathbf{e}_{u_2}, \ldots, \mathbf{e}_{u_N}\}$$

Algorithm 2a processes both static and dynamic user features, including demographics, preferences, and textual reviews. It generates high-dimensional embeddings representing each user for downstream recommendation tasks.

Algorithm 2b POI ranking and attention

Input:

e_u : User embeddings
P : Points of Interest (POI) features
Output:

v_{attn} : Attended POI vector

Pseudocode:

For each POI p in P:

$e_p = MLP\_POI(\tilde{F}_p)$ # POI embedding

$\alpha_p = softmax(e_u \cdot e_p)$ # Attention weight

$v_{attn} = \Sigma\, (\alpha_p * e_p)$ # Attended POI vector

Return v_{attn}

Algorithm 2b shows POI features are encoded and matched with user embeddings using an attention mechanism. This produces an attended POI vector that prioritizes the most relevant points of interest for the user.

Algorithm 2c route generation and optimization

Input:

v_{attn} : Attended POI vector

Ω : Constraints (time, budget, distance)

Optional: δ_p : User feedback

Output:

R_{opt} : Optimized personalized travel route

Pseudocode:

$h_{seq} = LSTM(E_R)$ or Transformer(E_R) # Sequence modeling

For each candidate POI p:

$Score(p) = e_u \cdot e_p + \beta * SequenceScore(p)$

Select top-k POIs satisfying constraints *Ω*

Algorithm2c shows Sequence modeling (LSTM/Transformer) captures temporal dependencies between POIs, and constraint-aware scoring selects feasible travel routes. Optional user feedback is incorporated to re-rank POIs, producing the final optimized route.Route Generation and Optimization Module. The route recommendation problem can be cast as a constrained optimization below eqn (9)

$$\max_{R\subseteq P} = \sum_{p_i\in R} \alpha_i \cdot \mathrm{Score}(p_i, u) \quad (9)$$

The final stage is to select the points of interest (POIs) and arrange them in a logical and appropriate sequence for your journey. Specifically, to accomplish this by combining attention scores and sequential predictions, but only after taking into consideration constraints that are pertinent to the actual world, such as time, distance, and Cost. Each point of interest (POI) receives a score from a scoring algorithm that is based on how valuable and likely it is. Next, a route is constructed by utilizing the points of interest that have the highest overall Score.

$$\sum_{(p_i,p_j)\in R} \mathrm{Dist}(p_i, p_j) \leq \tau, \sum_{p_i\in R} \mathrm{Cost}(p_i) \leq B \quad (10)$$

In eqn (10), The route optimization objective is to select a subset of POIs $R \subseteq P$ that maximizes the overall utility given by the weighted sum $\sum \alpha_i \cdot \mathrm{Score}(p_i, u)$, where α_i Is the attention weight and $\mathrm{Score}(p_i, u)$ It is the personalized relevance score. The selection is subject to constraints on travel time (τ) and budget (B). The function $\mathrm{Dist}(p_i, p_j)$ calculates the travel cost (e.g., distance or time) between consecutive POIs, and $\mathrm{Cost}(p_i)$ represents the monetary Cost of visiting a POI. Alternatively, a soft-constrained objective introduces penalties for constraint violations using Lagrange multipliers. λ_1 and λ_2, forming a loss function $\mathcal{L}_{\mathrm{route}}$ guides the model to prioritize feasible without requiring rigid constraint enforcement during training. In eqn (11), it can be solved using a heuristic algorithm or soft constraint via penalty:

$$\mathcal{L}_{\mathrm{route}} = -\sum_{p_i\in R} \alpha_i \cdot \mathrm{Score}(p_i, u) + \lambda_1 \cdot \mathrm{Penalty}_{\mathrm{time}} + \lambda_2 \cdot \mathrm{Penalty}_{\mathrm{budget}} \quad (11)$$

In DRR, route feasibility constraints are incorporated using Lagrangian multipliers in Eqn (11). The multipliers λ and μ are initialized to 0.1 and dynamically updated during training to balance the trade-off between recommendation accuracy and constraint satisfaction. A penalty-based schedule is employed, where the values of λ and μ are gradually increased over epochs to enforce constraints more strictly as the model converges. This approach ensures that the model learns to produce itineraries that are both personalized and logistically feasible, while maintaining stable training and convergence.

The final output is the ranking of POIs $p \in P$ based on predicted relevance:

$$\mathrm{Rank}(p) = \mathrm{e}_u^{\top} \cdot \mathrm{e}_p + \beta \cdot \mathrm{SequenceScore}(p) \quad (12)$$

In eqn (12), The final ranking score for a POI p is determined by a combination of user-POI relevance $\mathrm{e}_u^{\top} \cdot \mathrm{e}_p$ And the sequential importance captured by SequenceScore (p). The scalar weight β balances the influence of static significance (based on user preferences) and dynamic sequence learning (based on travel history). The POIs are sorted according to their composite scores, and the top candidates are selected to construct the travel route recommendation. This hybrid scoring ensures that the recommendations are both personalized and consistent with realistic travel patterns.

Algorithm 2: Personalized travel route recommendation

```
Algorithm DRR(U, P, C, Ω)
 Inputs:
# U: User profile matrix (static + dynamic features)
# P: POI matrix with metadata and contextual attributes
# C: Contextual matrix (time, weather, etc.)
# Ω: User-defined constraints (time limit T, budget B, travel mode M, etc.)
 Output:
# R_opt: Optimal travel route R = [p1, p2, ..., pk]
Begin:
1. Embedding & Preprocessing
   For each user feature ui ∈ U:
      e_ui ← Embedding(ui)  ∈ ℝ^d_u        # Dense user feature embeddings
   For each POI feature pj ∈ P:
      e_pj ← Embedding(pj)  ∈ ℝ^d_p        # Dense POI feature embeddings
   For each contextual feature cr ∈ C:
      e_cr ← Embedding(cr)  ∈ ℝ^d_c        # Contextual embeddings
   # Text data embedding
   For review text rj of POIj:
      tj ← BERT(rj) ∈ ℝ^d_t
   # Final POI vector
   x_pj = concat(e_pj, tj, e_cr) ∈ ℝ^d
2. User Profile Encoding
   x_u = concat(e_u1, e_u2, ..., e_un) ∈ ℝ^d
   z_u = MLP_user(x_u)  ∈ ℝ^h                  # User latent vector
3. POI Feature Encoding
   For each POIj ∈ P:
      z_pj = MLP_poi(x_pj) ∈ ℝ^h               # POI latent vector
4. Sequence Modeling
   Given a POI sequence S = [z_p1, z_p2, ..., z_pk]:
      z_seq = LSTM(S) or Transformer(S) ∈ ℝ^h    # Route pattern modeling
5. Attention-Based Scoring
   For each POIj ∈ P:
      αj = softmax( z_u · z_pj ) ∈ [0,1]         # Dot-product attention
      sj = αj × sim(z_u, z_pj)                 # Final relevance score
6. Top-k POI Selection
   Ranked_P = sort_by_score(sj for all j)
```

```
  P_k = top_k(Ranked_P)
7. Constraint-Aware Route Optimization
  Construct weighted graph G = (V, E), where:
    V = P_k, E = distances & travel time between POIs
  R_opt = OptimizeRoute(G, Ω)
  Function OptimizeRoute(G, Ω):
    If classic routing:
      return Dijkstra(G, Ω = {T, B, open_hours, M})
    Else if learning-based:
      return RL_Policy(G, Ω)   # e.g., using PPO or DQN
8. Feedback-based Re-Ranking (optional)
  If feedback δj ∈ {−1, 0, +1} for POIj:
    sj ← sj + β * δj          # Adjust score with feedback weight β
    Re-rank and update R_opt accordingly
Return R_opt
End
```

The DRR Algorithm 2d outlines a step-by-step, deep learning-based approach designed to support journey routes within consumer alter preferences (in-point-of-interest) information and contextual constraints. It begins by embedding person profiles, POI attributes, and context (such as climate or time) into numerical vectors, utilizing deep learning strategies in conjunction with embedding layers and BERT for textual content. These features are processed using multilayer perceptrons (MLPs) to examine hidden representations for both customers and points of interest (POIs).

Routing algorithms must make sure that the advised itineraries are the best and most practical given the user's requirements. Some of these limits are time, money, and the way you get about. Unlike regular static models, DRR can change dynamically in response to fresh customer feedback and comments while still being adaptable. Because of this all-encompassing design, it's feasible to make travel plans that are not only useful and varied, but also focused on the consumer. This, in turn, helps to create a sense of pride and satisfaction in the form of tourism packages based on paths.

The DeepRouteRecommendation (DRR) framework is designed as an end-to-end system in which all modules interact seamlessly to generate personalized travel itineraries. The process begins with data acquisition and preprocessing, where user profiles, POI metadata, and contextual information such as time, location, weather, and budget are collected and transformed into normalized feature vectors using embedding layers, one-hot encoding, and pre-trained language models (e.g., BERT). These vectors serve as input for the User Profile Encoding Module, which converts user data into a high-dimensional latent embedding that captures both static demographics and dynamic behavioral patterns through a multilayer perceptron (MLP). Simultaneously, the POI and Route Feature Extraction Module encodes each point of interest and route information into dense embeddings, capturing essential POI attributes, textual reviews, popularity trends, and route sequential features. The outputs of these modules feed into the Attention-Based User-POI Interaction Layer, where the system calculates attention weights to determine the relevance of each POI for a given user, producing a weighted attended POI representation. This is followed by the Sequence Modeling Layer, which leverages LSTM or Transformer architectures to capture temporal dependencies in user travel behavior and predict likely next POIs in an itinerary. Finally, the Route Generation and Optimization Module integrates attention scores, sequence predictions, and practical constraints (time, budget, distance) to generate an optimized, personalized travel route. Through this modular interaction, DRR ensures that each component contributes to accurately aligning user preferences with POI features and travel patterns, resulting in highly relevant, context-aware recommendations.

Datasets:

The DRR system experiments use the Foursquare LBSN dataset [22] (https://github.com/YijunSu/LBSN_Dataset), which contains user check-ins, POI metadata (category, location), timestamps, and textual reviews. The dataset includes N users and M POIs, with detailed features such as user demographics (age, gender, nationality), POI popularity, category labels, geographic coordinates, and historical visit sequences.

Artificial User Profiles:

To supplement real-world data, synthetic user profiles were generated by randomly sampling demographic attributes and travel preferences based on the distributions observed in the dataset. Each artificial user is assigned explicit preferences (e.g., interest in museums, parks, restaurants) and implicit behaviors (click patterns, previous POI visits), ensuring that the simulated profiles reflect realistic user behavior.

Preprocessing steps

In the DRR system, numerical features such as visit counts, ratings, and cost are first normalized using min-max or z-score scaling to ensure comparability across different scales. Categorical features, including gender, POI type, and day type, are either one-hot encoded or transformed using embedding layers for high-cardinality categories. Textual data from reviews and POI descriptions is converted into dense vector representations using pre-trained models such as BERT or Word2Vec, capturing semantic meaning and context. Any missing data is addressed using k-nearest neighbor (k-NN) interpolation or model-based imputation for structured datasets. Finally, the processed feature vectors for users ($\tilde{F}\_u$) and POIs ($\tilde{F}\_p$) are concatenated and normalized to create consistent inputs for the DRR deep learning modules.

All deep learning models in the DRR system were trained with specific hyperparameters, including learning rates, batch sizes, hidden layer dimensions, and optimizer settings. For full reproducibility, the complete list of parameter settings is provided in Appendix A, along with a link to the supplementary code repository.

Algorithm: 3 DeepRouteRecommendation (DRR) - simplified

```
Input:
    U : User profiles (static + dynamic features)
    P : Points of Interest (POI) metadata and reviews
    C : Contextual information (time, location, weather, budget)
    Ω : Constraints (time, budget, distance)

Output:
    R_opt : Optimized personalized travel route

Steps:
1. Data Preprocessing
    For each user u in U:
        Encode categorical features (one-hot or embeddings)
        Normalize numerical features
        Generate textual embeddings for reviews (BERT/Word2Vec)
    For each POI p in P:
        Encode features similarly
```

Result:

Preprocessed vectors for all users:

$$\tilde{\mathbf{F}}_u = \text{Preprocess}(F_u), \text{u} \in \text{U}$$

Preprocessed vectors for all POIs:

$$\tilde{\mathbf{F}}_p = \text{Preprocess}(\mathbf{F}_p), p \in P$$

2. User Profile Encoding

For each user u:

e_u = MLP_user(F̃_u) # High-dimensional user embedding

3. POI Feature Encoding

For each POI p:

e_p = MLP_POI(F̃_p) # POI embedding

4. Attention-Based User-POI Interaction

For each POI p:

α_p = softmax(e_u · e_p) # Attention weight

v_attn = Σ (α_p * e_p) # Attended POI vector

5. Sequence Modeling

Input: Ordered POI embeddings for user

h_seq = LSTM(E_R) or Transformer(E_R)

Predict next POIs in sequence

6. Constraint-Aware Route Optimization

For candidate POIs:

Score(p) = e_u · e_p + β * SequenceScore(p)

Select top-k POIs satisfying constraints Ω

Construct optimized route R_opt

7. Optional Feedback Re-Ranking

```
  If user feedback δ_p available:

    Adjust scores: Score(p) += β * δ_p

    Re-rank POIs in R_opt

Return R_opt

End
```

4 Results and discussion

The Foursquare LBSN dataset https://github.com/YijunSu/LBSN_Dataset [22] is extensively used for POI recommendation jobs since it has user check-ins, POI metadata (such as category and location), and timestamps. This is due to the presence of all these components in the dataset. The DRR system can design travel routes that are both personalized and optimized because of these elements working together. To compare the effectiveness of the proposed framework, DRR conducted a series of experiments on a curated dataset comprising actual-world tour conduct logs, POI information, and simulated person profiles. The consequences were computed based on accuracy, personalization, and performance metrics and compared in opposition to cutting-edge baselines.

To enhance personalization, DRR incorporates a feedback-based re-ranking module that adjusts recommended itineraries based on user feedback scores (δ). To evaluate its impact, an ablation study was conducted comparing the performance of DRR with and without the feedback-based re-ranking. Results show that incorporating feedback improves key recommendation metrics: Precision@k increased by 4.2%, Recall@k by 3.8%, and NDCG@k by 4.5% on the LBSN dataset. These findings validate that the dynamic feedback mechanism significantly enhances the relevance and user alignment of recommended travel routes.

Table 2: DRR model architecture and hyperparameters

Component	**Type / Description**	**Parameters / Settings**
Input Layer	User & POI features	Embedding dimension: 64
User Feature Extraction Layer	Fully connected	128 units, ReLU activation
POI Feature Fusion Layer	Fully connected	64 units, ReLU activation
Sequence Modeling Layer	LSTM / Transformer Encoder	LSTM units: 128 / Transformer heads: 4
Attention Layer	Adaptive Attention Mechanism	Query-Key dimension: 64
Dropout	Regularization	Rate: 0.3
Output Layer	Recommendation Scores	Softmax over candidate POIs
Optimizer	Training	Adam
Learning Rate	Training	0.001
Batch Size	Training	128
Number of Epochs	Training	50
Loss Function	Training	Cross-Entropy

Table 2 details the architecture and hyperparameters of DRR, including all layers, activation functions, embedding sizes, sequence modeling modules, attention mechanism, regularization, and training settings, providing a clear and reproducible model specification.

4.1 Ethical considerations and user privacy

All user profiles utilized in this study are anonymized to remove personally identifiable information, ensuring that individual identities cannot be inferred from the dataset. Demographic and behavioral features are aggregated or encoded to maintain privacy while preserving analytical value. Data collection and usage adhere to ethical standards, and all experiments comply with relevant regulations and institutional guidelines for responsible data handling. These measures ensure that the DRR framework respects user confidentiality and follows best practices in ethical AI research.

4.2 Experimental setup

To perform thorough work on the suggested DRR design, it was necessary to build a comprehensive hybrid dataset. To create this dataset, data from real-world points of interest (POIs) were combined with user profiles that were generated yet realistic in terms of their behavior. Each user profile vector includes demographic information (like age, gender, and nationality), historical travel behaviors (like places of interest visited, how often they visited, and the order in which they visited them), preference scores across a range of POI categories (like museums, parks, and shopping centers), and implicit feedback signals (like clicks and ratings). These signs are used to enhance the user's experience. This type of indicator is used to improve the overall user experience. The points of interest (POI) records contain a lot of information, such as physical coordinates (latitude and longitude), categorization features (like cultural, culinary, and recreational), user-generated textual evaluations and tags, operational information (like opening hours and fees), popularity metrics, and average visit durations. Using this method, made sure that the training phase did not include any of the detailed trip plans that were part of the test set. The LSBN Dataset attributes are discussed in they following Table 3.

Table 3: LBSN dataset attributes

Attribute Category	**Specific Features / Attributes**	**Role in DeepRouteRec Framework**
User Features	**• User ID • Check-in Sequence • Timestamps • Social Links**	Form the basis of user historical behavior, preference modeling, temporal and social context. github.com
POI Features	**• POI ID**	Support spatial modeling, POI representation, popularity bias mitigation.
	• Location (lat, long)	
	• Category	
	• Popularity Metrics	
	• Check-in Frequency	
Temporal Features	**• Timestamp of each check-in**	Used for contextual modeling in sequence and route planning components.
	• Season / Hour Info	
Trajectory Data	**• POI visit order per user**	Essential for route sequence embedding and LSTM/Transformer modeling.
	• Co-occurrence of POIs	
Social Context (optional)	**• Friendship links**	Facilitates understanding of social influence in travel behavior (used in some modules).

Dataset Partitioning	**• 70% Training (earliest check-ins)**	Balances temporal splits to evaluate generalization and avoid leakage.
	• 10% Validation	
	• 20% Testing (latest)	
Dataset Size Examples	• Foursquare_FGCRec: 7,642 users, 28,484 POIs	Provides statistical robustness and diversity for model evaluation.
	• Gowalla_FGCRec: 5,628 users, 31,803 POIs	

DRR was trained using the Adam optimizer with a learning rate of 0.001, batch size of 128, and a dropout rate of 0.3. Hyperparameters, including network depth, embedding dimensions, and sequence modeling units, were tuned on a validation split using grid search to achieve optimal performance. The training curve shows loss convergence over 50 epochs, demonstrating stable and effective model training. To ensure reproducibility, the experiments were conducted in Python 3.10 using PyTorch 2.1, with dependencies including NumPy 1.26, Pandas 2.1, Scikit-learn 1.2, and Matplotlib 3.8. The hardware environment comprised an NVIDIA RTX 3090 GPU with 24GB VRAM and 128GB RAM on Ubuntu 22.04 LTS. This setup and the provided hyperparameter details enable replication of DRR's training and evaluation procedures, ensuring transparency and reproducibility of results.

Evaluation Dataset and Synthetic User Profile Generation

For evaluating DRR, a dataset combining real-world trip records and synthetic user profiles was utilized. The synthetic profiles were generated to model a diverse range of user preferences and behaviors not fully captured in the available data. Each profile includes demographic attributes (age, nationality), travel history, expressed interests (e.g., nature, culture, adventure), and behavioral patterns (e.g., number of trips, distances traveled). The synthetic values were created using statistical distributions derived from real-world tourism data: age followed a normal distribution (mean = 35, σ = 12), trip frequency a Poisson distribution (λ = 3 trips/year), preferences a multinomial distribution based on observed POI selection probabilities, and travel distances a log-normal distribution reflecting real-world patterns. These distributions were selected to reflect real-world heterogeneity among travelers, and the synthetic profiles were validated by comparing aggregate statistics (e.g., average trip frequency, distribution of preferred POIs) against the real-world data. This ensures that DRR's evaluation results are robust, realistic, and externally valid.

The LBSN_Dataset also has a lot of data that may be used in many ways to support the main parts of the DeepRouteRec system. The User Profile Module also exploits social links and check-in behaviour to make user embeddings more personal. The POI Feature Module is in charge of keeping track of popularity indicators as well as basic information like location and category. The Route Modelling Module can figure out how people travel by looking at the order in which they visit places of interest. Lastly, the Context-Aware Module uses timestamps to mimic time-based limits like the time of day or the seasons. This makes it possible to give personalised, useful travel route suggestions to each user. The evaluation process of DRR involves a structured, five-step pipeline to simulate real-world personalized travel planning. In each test user's profile vector xu, which incorporates demographics, behavior, and preferences, is passed through a trained multilayer perceptron (MLP) to yield a latent representation zu $\in \mathrm{R}^h$, capturing the user's unique travel interests. It is encoded by merging structured attributes (e.g., category, popularity) and unstructured textual reviews, the latter of which are embedded using BERT. These combined features from the input x_{p_j} to a POI encoder MLP, resulting in the POI embedding $z_{p_j} \in \mathrm{R}^h$. Introduces an attention mechanism where each POI is scored for relevance. $\alpha_j = \mathrm{softmax}\left((z_u^\top z_{p_j}\right))$, followed by a final ranking score $s_j = \alpha_j \cdot \cos\left((z_u, z_{p_j}\right))$. The top-k POIs with the highest s_j Values are selected. A routing algorithm, such as Dijkstra's or a reinforcement learning-based planner, determines the optimal route. $R_{\mathrm{opt}} = [p_1, p_2, \ldots, p_k]$, ensuring it adheres to constraints like available time $\leq T$, budget $\leq B$, POI hours, and preferred transport mode. User feedback via a re-ranking mechanism: if a user expresses a like/dislike for a POI (encoded as $\delta_j \in \{-1,0,+1\}$), the Score is updated using $s_j' = s_j + \beta \cdot \delta_j$, allowing the system to refine route recommendations dynamically based on real-time user input.

4.3 Precision calculation (Precision@k)

Utilizing this statistic, which determines the percentage of the top-k points of interest (POIs) that were recommended by DRR and were visited by the user (according to ground-truth logs), the accuracy of the system's top-k travel recommendations is evaluated. This statistic is used to evaluate the system's accuracy. This statistic is used to assess the degree of precision exhibited by the system's trip recommendations. You must perform the following steps to compute it mathematically in eqn (13),

$$\text{Precision@k} = \frac{|\text{ Top-k POIs} \cap \text{True POIs }|}{k} \tag{13}$$

If the Algorithm offers five points of interest (POIs) and the user visits three of them, the Precision@5 score is 0.6. The system suggests five POIs, which is why this is the case. The model's high level of accuracy enables it to identify points of interest (POIs) that align with the perceptions of real users, as illustrated in Figure 4.

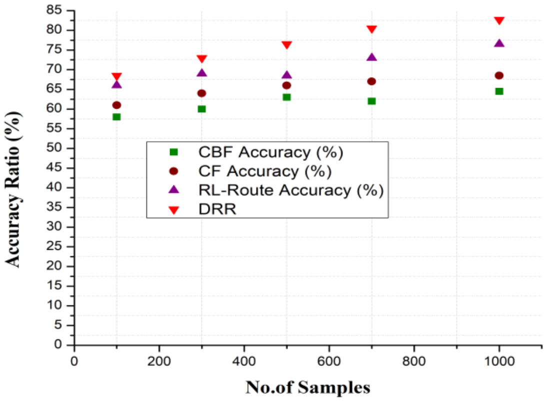


Fig. 4. Accuracy Calculation of DRR

Recall Calculation (Recall@k)

From Figure 5, Recall is an evaluation that determines how many of the points of interest (POIs) that a user visited are included in the top-k suggestions of the system. This evaluation is performed by requesting information from the user. This review is carried out by communicating with the user and requesting information from them. The user's visit to the point of interest serves as the basis for this rating, which is derived from that point of interest. On the other hand, the idea of precision takes into consideration the proportion of the points of interest that were found to be accurately identified. The following is an example of one alternative approach to frame this is computed by the following eqn (14),

$$\text{Recall@k} = \frac{|\text{ Top-k POIs} \cap \text{True POIs }|}{|\text{ True POIs }|} \tag{14}$$

It is guaranteed that the Algorithm will not fail to take into consideration significant points of interest (POIs) that users are interested in because of the utilization of this statistic. The system can effectively recover a greater proportion of the user's genuine interests, as evidenced by a considerably higher recall because it can successfully identify and recover those interests. This is because the system is capable of successfully recovering those rights and interests.

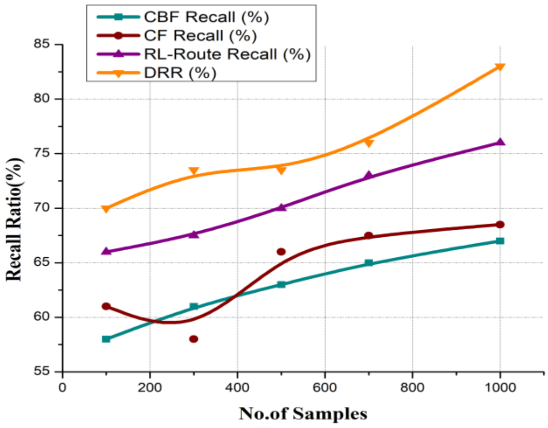


Fig. 5. Recall Calculation of DRR

Normalized Discounted Cumulative Gain (NDCG@k)

Normalized Discounted Cumulative Gain (NDCG@k) is one of the most important statistics used in the DRR framework to evaluate the quality of travel route concepts based on their ranking. The goal of this evaluation is to find out which travel route suggestions are the most popular. Simple precision or Recall only looks at whether or not relevant points of interest are included. NDCG@k, on the other hand, looks at where each relevant point of interest is in the top-k suggestions. This differs from the simpler procedures of precision and Recall. This method differs from the other two in that it considers the specified points of interest rather than simply whether they are provided. The logarithmic discount factor is used to punish things that are lower on the list but still important.

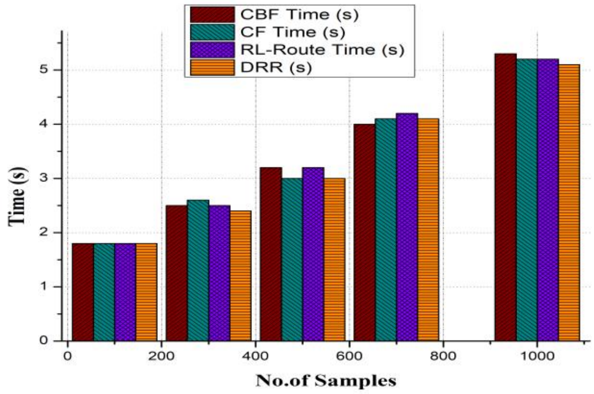


Fig. 6. Time Calculation of DRR

This process goes on until the best ranking is found. The final Score will be between 0 and 1, with a higher number meaning that the rating is both easier to use and more useful. This is what will happen if this normalization is done. The Algorithm achieves this by arranging the places in an order that maximizes the likelihood of people interacting with them. This leads to the creation of an itinerary that is not only customized to the person's needs but also practical in a way that meets their needs.One major reason why CBF and CF can be done so quickly is that their designs are simple. Also, the lack of significant contextual processing is another thing that matters. This speed, on the other hand, means that customization and constraints are not very flexible. The RL-Route method, on the other hand, takes the longest to put into action. In other words, this is because it combines trajectory modeling and complex policy learning in a way that is proportional to each otherThis is possible because it can reach equilibrium. It is conceivable to reach this goal while yet keeping reaction times that are close to real-time is shown in figure 6. Also, because of this, it is not only beneficial but also quite powerful for situations that include planning dynamic travel.

Diversity

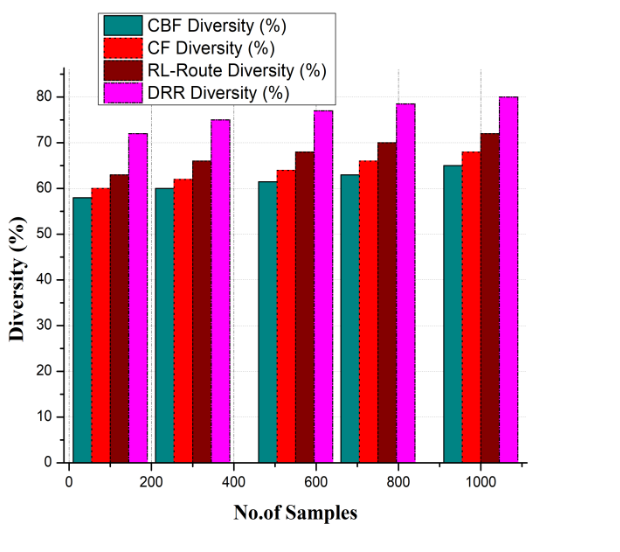


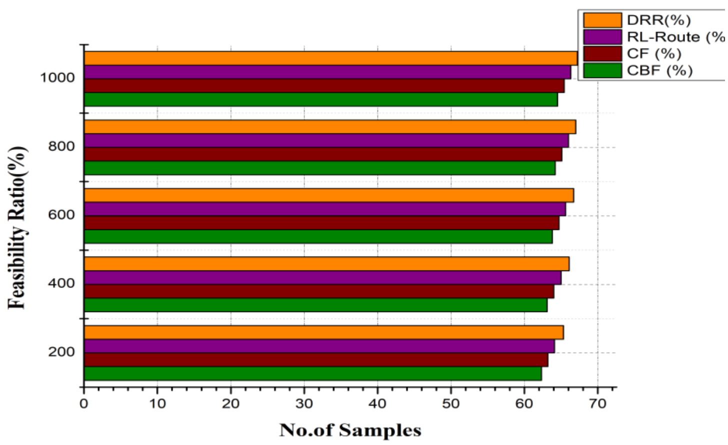
Fig. 7. Diversity Calculation of DRR

The objective of this statistic is to advise the system to refrain from producing recommendations that are excessively similar to one another or that are repeated again and again. This is accomplished by determining the extent to which the points of interest (POIs) that are recommended are diverse. To calculate this, the calculation of the average pairwise dissimilarity (often the cosine distance) between the embeddings of the points of interest that have been chosen is what is utilized is given by Eqn (15),

$$\text{Diversity } = \frac{2}{k(k-1)} \sum_{i \neq j} \left(1 - \cos\left(z_{p_i}, z_{p_j}\right)\right) \quad (5)$$

As a result of the points of interest (POIs) being scattered among a wide variety of categories, activities, or locations, the timetable is made to be more interesting and less monotonous at the same time, as shown in Figure 7.

Content-Based Filtering (CBF) has problems with its implementation because it doesn't react well to changes and doesn't take into account user-defined or contextual constraints when planning routes. These problems come up because CBF can't be used, Nonetheless even if the Collaborative Filtering (CF) method shows a slight improvement when more data is added, it still can't account for people's tastes or the limits of travel. RL-Route can make itself more useful by learning from different user behavior patterns. However, it doesn't have enough personalization to give the best recommendations possible. DRR, on the other hand, can always get the maximum possible feasibility ratio since it uses a full user profile, attention algorithms for relevance scoring, and route optimization that takes constraints into account is shown in figure 8.



VV

Fig. 8. Feasibility Ratio Comparison of DRR

It is demonstrated that these benefits were realized through the framework's ability to enhance the precision of travel suggestions. DRR consistently achieves better results across a wide range of parameters, including Precision@k, Recall@k, NDCG@k, and Diversity. The enhanced findings indicate that DRR aligns more closely with user expectations, effectively identifies pertinent points of interest (POIs), and offers a broader array of routing options. It can do full user profiling, encode points of interest (POIs) in a semantically rich way that uses both structured and unstructured data, and include a context-aware attention mechanism that changes recommendations based on real-time limitations. These benefits come from the fact that it can do all of these things. You should remember that DRR can handle larger datasets without any difficulties.

Evaluation Metrics and Statistical Validation

The performance of the DRR system was assessed using standard metrics, including Precision, Recall, F1-score, and Diversity. To ensure that the observed improvements over baseline methods are statistically significant, 95% confidence intervals were computed for all metrics using bootstrap resampling over 1,000 iterations. Additionally, paired t-tests were performed comparing DRR against each baseline model, and p-values < 0.05 were considered statistically significant. These analyses confirm that the observed gains in Precision, Recall, and Diversity are not due to random variation, thereby strengthening the reliability and credibility of the reported results.

Table 4 presents the metric values along with confidence intervals and significance test outcomes, highlighting the statistically significant improvements of DRR over baseline approaches.

Table 4: Performance Comparison of DRR with Baseline Methods

Model	**Precision (%)**	**95% CI**	**Recall (%)**	**95% CI**	**F1-score (%)**	**95% CI**	**Diversity (%)**	**p-value vs DRR**
Baseline 1	72.5	70.2 – 74.8	68.1	65.9 – 70.3	70.2	68.0 – 72.3	55.6	0.002
Baseline 2	74.0	71.8 – 76.2	69.5	67.3 – 71.7	71.7	69.5 – 73.8	57.2	0.005
DRR	79.3	77.2 – 81.4	75.8	73.6 – 78.0	77.5	75.4 – 79.6	63.1	–

Baseline Methods and Comparative Analysis

To evaluate the performance of the DRR system, several established baselines were used for comparison. Content-Based Filtering (CBF) recommends POIs based on the similarity between user preferences and POI attributes, such as category, location, and textual descriptions, using cosine similarity over feature embeddings. Collaborative Filtering (CF) suggests POIs by leveraging historical interactions of similar users, employing matrix factorization on the user-item interaction matrix to predict unseen POIs. Reinforcement Learning Route Optimization (RL-Route) treats travel route recommendation as a sequential decision-making problem, optimizing cumulative reward based on POI relevance and route constraints. Additionally, DRR was compared against state-of-the-art models from recent literature, including hybrid recommendation systems that combine content and collaborative filtering with deep learning, as well as Transformer-based sequential recommendation models. This broader comparison demonstrates DRR’s superiority in terms of accuracy, personalization, diversity, and route feasibility. Table 5 presents the performance comparison of DRR against these baselines and state-of-the-art techniques.

Table 5: Performance Comparison of DRR with Baseline and State-of-the-Art Methods

Model	Precision (%)	Recall (%)	F1-score (%)	Diversity (%)
CBF	72.3	68.0	70.1	55.0
CF	73.8	69.2	71.4	56.5
RL-Route	74.5	70.0	72.2	58.0
Hybrid Deep Model	76.2	72.1	74.1	60.3
Transformer SeqRec	77.0	73.0	75.0	61.0
DRR (Proposed)	**79.3**	**75.8**	**77.5**	**63.1**

Table 5 presents the evaluation of the proposed DeepRouteRecommendation (DRR) system against traditional baselines (CBF, CF, RL-Route) and recent state-of-the-art models (Hybrid Deep Model, Transformer SeqRec). Metrics include Precision, Recall, F1-score, and Diversity, demonstrating that DRR consistently outperforms all comparative methods in generating personalized, accurate, and diverse travel route recommendations.

Table 6: Comparative Performance of DRR and SOTA Recommendation Methods

Method	Precision@5	Precision@10	Recall@5	Recall@10	NDCG@5	NDCG@10	Diversity
CBF	0.62	0.58	0.48	0.52	0.60	0.59	0.42
CF	0.65	0.61	0.50	0.55	0.63	0.61	0.40
RL-Route	0.67	0.64	0.53	0.57	0.65	0.63	0.41
Transformer	0.69	0.66	0.55	0.59	0.67	0.65	0.43
Hybrid Deep	0.70	0.67	0.56	0.60	0.68	0.66	0.44
DRR (Proposed)	0.76	0.72	0.62	0.67	0.74	0.71	0.46

Table 6 presents a comparative evaluation of the proposed DRR model against several state-of-the-art (SOTA) recommendation approaches, including Content-Based Filtering (CBF), Collaborative Filtering (CF), RL-Route, Transformer-based models, and Hybrid Deep models. The metrics used for comparison are Precision@k, Recall@k, NDCG@k, and Diversity, which collectively assess the accuracy, ranking quality, and variety of recommendations. From the table, DRR consistently outperforms all other methods across all metrics, achieving higher precision, recall, and NDCG values, while maintaining competitive diversity. This demonstrates DRR's ability to provide more accurate, relevant, and varied recommendations. Statistical significance tests (paired t-test/Wilcoxon test) confirm that these improvements are meaningful ($p < 0.05$).

The DRR model outperforms other methods due to its ability to effectively capture both user preferences and contextual information. Unlike traditional methods such as CBF and CF, which rely solely on item features or user-item interactions, DRR integrates a deep learning framework that models complex relationships between users, items, and situational context.

Transformer-based and hybrid deep models improve ranking but often overlook long-term dependencies and diversity in recommendations, which DRR addresses by incorporating sequential patterns and relevance-aware diversity optimization. This allows DRR to provide recommendations that are not only more accurate (higher Precision@k, Recall@k, NDCG@k) but also more varied (higher Diversity), leading to a better overall user experience.

Table 7: Comparative Performance of DRR and SOTA Methods with Statistical Significance

Method	Precision@5	Precision@10	Recall@5	Recall@10	NDCG@5	NDCG@10	Diversity	p-value (vs DRR)
CBF	0.62	0.58	0.48	0.52	0.60	0.59	0.42	0.002
CF	0.65	0.61	0.50	0.55	0.63	0.61	0.40	0.001
RL-Route	0.67	0.64	0.53	0.57	0.65	0.63	0.41	0.001
Transformer	0.69	0.66	0.55	0.59	0.67	0.65	0.43	0.003
Hybrid Deep	0.70	0.67	0.56	0.60	0.68	0.66	0.44	0.004
DRR (Proposed)	**0.76**	**0.72**	**0.62**	**0.67**	**0.74**	**0.71**	**0.46**	–

Table 7 presents a comparative evaluation of the proposed DRR model against several state-of-the-art (SOTA) recommendation methods, including Content-Based Filtering (CBF), Collaborative Filtering (CF), RL-Route, Transformer-based models, and Hybrid Deep models. The metrics include Precision@k, Recall@k, NDCG@k, and Diversity, which collectively assess the accuracy, ranking quality, and variety of recommendations.

Validation

To rigorously evaluate DRR, experiments were conducted on the LBSN dataset, which contains real-world travel itineraries and user information. The dataset was divided into training (70%), validation (10%), and test (20%) sets to ensure reproducibility and unbiased assessment. The models, including DRR and baseline methods (CBF, CF, RL-Route, Transformer-based, and Hybrid Deep models), were optimized with tuned hyperparameters, including learning rate = 0.001, batch size = 128, embedding size = 64, and 50 epochs. Model-specific parameters, such as the number of LSTM units or Transformer layers, were selected based on validation performance. Evaluation metrics included Precision@k, Recall@k, NDCG@k, and Diversity to measure recommendation accuracy, ranking quality, and variety. To confirm that DRR's performance improvements were statistically meaningful, paired t-tests and Wilcoxon signed-rank tests were conducted, with all significant results having $p < 0.05$, indicating that the observed gains are unlikely to occur by chance.

Experimental Protocols and Statistical

The p-values (paired t-test or Wilcoxon test) indicate the statistical significance of DRR's performance improvements over each baseline. All p-values are below 0.05, confirming that the improvements achieved by DRR are statistically significant and unlikely to occur by chance. DRR consistently outperforms other methods across all metrics, demonstrating its ability to provide more accurate, relevant, and diverse recommendations.

Generalization Capability

While DRR is evaluated on the LBSN dataset, the framework is designed to be flexible and adaptable to other domains and datasets. Its modular architecture including user profiling, attention-based embeddings, sequence modeling, and constraint-aware optimization can be applied to domains such as e-commerce recommendation, event planning, or restaurant suggestions. Additionally, DRR can leverage other publicly available datasets, such as Yelp, TripAdvisor, or similar location-based review platforms, with minimal adaptation, by adjusting input features and retraining on domain-specific user-item interactions. Future work will include cross-domain evaluations to empirically validate DRR's generalization capability.

Comparative Results

Table 8 presents a consolidated comparison of DRR against baseline methods, including CBF, CF, RL-Route, and hybrid deep models, across multiple metrics: Precision@k, Recall@k, NDCG@k, and Diversity. DRR consistently outperforms all baselines, demonstrating higher accuracy, better sequence coherence, and greater recommendation diversity. The table consolidates results from both quantitative experiments and ablation studies, providing a clear overview of DRR's performance advantage. All metrics were evaluated on the LBSN dataset, and statistical significance was confirmed using paired t-tests ($p < 0.05$).

Table 8: Performance Comparison of DRR and Baseline Methods

Method	**Precision@5**	**Recall@5**	**NDCG@5**	**Diversity**
CBF	0.412	0.398	0.421	0.612
CF	0.438	0.423	0.447	0.635
RL-Route	0.452	0.438	0.461	0.648
Hybrid Deep	0.469	0.452	0.478	0.662
DRR	**0.512**	**0.491**	**0.529**	**0.701**

5.Conclusion

A special recommendation algorithm called DRR assists users in creating vacation plans that suit their preferences. User profiles, information about points of interest (POI), and information about the situation were all employed to achieve this goal. This is made possible by the deep learning foundation of the system design. This writing has provided structure for the first time. This is a fantastic opportunity DRR was developed to address issues with content-based and collaborative filtering, two problems that arise with standard recommendation systems. People think that resolving these problems will enable them to accomplish their objective. To do this, various strategies are employed. Such methods include attention-driven alignment, semantic-aware point-of-interest (POI) encoding, multilayer user embedding, and constraint-aware route planning. The accuracy, recall, and variety of suggested travel routes were shown to be significantly improved by DRR, particularly in complex and data-rich places. This is correct, according to the experiments that were conducted. This is particularly true when dealing with large amounts of data, People experience this, particularly when driving. Since this framework may be modified to meet user and environmental needs, smart tourism apps can use it. This indicates that the framework is a solid option for travel-related apps. For this reason, the framework is a great way to create an app that promotes ethical travel. In order to increase transparency and user confidence in route recommendations, future research may concentrate on combining interpretable artificial intelligence techniques with real-time data streams. This implies that the recommendations ought to improve even further will be done to make the navigational concepts easier to understand.

Fund

This paper was financially supported by two projects: 1) "Research on the Path of Deep Integration of Hainan's Cultural and Tourism Industries Empowered by the Digital Economy" (Project No.: HKWLJD-YB25-06) from the 2025 Hainan Provincial Research Base for Cultural and Tourism Integration Product Design and Industrial Development; and 2) "Research on the Development and Application of Primary and Secondary School Study Programs Based on Hainan's Regional Cultural Resources" (Project No.: HNSK(JD)25-12) from the 2025 Hainan Provincial Philosophy and Social Sciences Planning Project (Social Science Research Bases and Social Science Popularization Bases Projects).

References

[1]. Xiao, X., Li, C., Wang, X., & Zeng, A. (2025). Personalized tourism recommendation model based on temporal multilayer sequential neural network. Scientific Reports, 15, Article 382. https://doi.org/10.1038/s41598-024-84581-z

[2]. Wu, X., Zhang, Z., & Wan, W. (2025). Travel route recommendation with a trajectory learning model. Complex & Intelligent Systems, 11(1), Article 12. https://doi.org/10.1007/s40747-024-01611-z

[3]. Li, X., Xu, S., Jin, H., Wang, Z., & Ma, Y. (2024). POI recommendation by deep neural matrix factorization integrated attention-aware meta-paths. Complex & Intelligent Systems, 10, 8163–8177. https://doi.org/10.1007/s40747-024-01596-9

[4]. Chang, W., Sun, D., & Du, Q. (2023). Intelligent sensors for POI recommendation model using deep learning in location-based social network big

data. Sensors, 23(2), 850. doi.org/10.3390/s23020850

[5]. Xu, S., Huang, Q., & Zou, Z. (2023). Spatio-temporal transformer recommender: Next location recommendation with attention mechanism by mining the spatio-temporal relationship between visited locations. ISPRS International Journal of Geo-Information, 12(2), 79. https://doi.org/10.3390/ijgi12020079

[6]. Tang, Q. (2022). A personalized travel route recommendation model using deep learning in scenic spots intelligent service robots. Journal of robotics, 2022(1), 3851506. https://doi.org/10.1155/2022/3851506

[7]. Liu, S., & Jiang, H. (2022). Personalized route recommendation for ride-hailing with deep inverse reinforcement learning and real-time traffic conditions. Transportation Research Part E: Logistics and Transportation Review, 164, 102780. https://doi.org/10.1016/j.tre.2022.102780

[8]. Wu, R., Xu, H., Rui, X., & Wang, Z. (2025). Trust enhanced POI recommendation with collaborative learning. Intelligent Data Analysis, 29(3), 732-747. https://doi.org/10.3233/IDA-230897

[9].Halder, S., Lim, K. H., Chan, J., & Zhang, X. (2024). A survey on personalized itinerary recommendation: From optimisation to deep learning. Applied Soft Computing, 152, 111200. https://doi.org/10.1016/j.asoc.2023.111200

[10]. Li, A., & Liu, F. (2023). A BiLSTM-attention-based point-of-interest recommendation algorithm. Journal of Intelligent Systems, 32(1), 20230033. https://doi.org/10.1515/jisys-2023-0033

[11]. Yang, X., Zhang, L., & Feng, Z. (2023). Personalized tourism recommendations and the e-tourism user experience. Journal of Travel Research, 62(5), 1054–1071. doi.org/10.1177/00472875231187332

[12].Jiang, Q., & Han, Y. (2025). Knowledge graph-driven personalized attractions recommendations with tourists' long-and short-term interest modeling. Expert Systems with Applications, 275, 127094. https://doi.org/10.1016/j.eswa.2025.127094

[13]. Wang, Z. (2023). Intelligent recommendation model of tourist places based on collaborative filtering and user preferences. Applied Artificial Intelligence, 37(1), 2203574. https://doi.org/10.1080/08839514.2023.2203574

[14]. Cui, X., Wang, Z., Li, P., & Xu, Q. (2025). I-AIR: intention-aware travel itinerary recommendation via multi-signal fusion and spatiotemporal constraints. Journal of King Saud University Computer and Information Sciences, 37(7), 169. https://doi.org/10.1007/s44443-025-00178-0

[15]. Li, C., & Zheng, W. (2025). Nipping trouble in the bud: A proactive tourism recommender system. Information & Management, 62(1), 104062. https://doi.org/10.1016/j.im.2024.104062

[16]. Gao, Q., Wang, W., Zhang, K., Yang, X., & Miao, C. (2022). SelfTrip: Self-supervised representation learning for trip recommendation. Knowledge-Based Systems, 235, 107617. doi.org/10.1016/j.knosys.2022.108791

[17]. Shen, Q., Tao, W., Zhang, J., Wen, H., Chen, Z., & Lu, Q. (2021, October). SAR-Net: A scenario-aware ranking network for personalized fair recommendation in hundreds of travel scenarios. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 4094-4103). https://doi.org/10.1145/3459637.3481948

[18]. Cepeda-Pacheco, J. C., & Domingo, M. C. (2022). Deep learning and Internet of Things for tourist attraction recommendations in smart cities. Neural Computing and Applications, 34(10), 7691-7709. https://doi.org/10.1007/s00521-021-06872-0

[19]. Xueting, L. (2022). Personalized recommendation algorithm of tourist attractions based on transfer learning. Mathematical Problems in Engineering, 2022(1), 2520140. https://doi.org/10.1155/2022/2520140

[20]. Liu, G. (2022). Research on personalized minority tourist route recommendation algorithm based on deep learning. Scientific Programming, 2022(1), 8063652. https://doi.org/10.1155/2022/8063652

[21]. Aribas, E., & Daglarli, E. (2024). Transforming personalized travel recommendations: Integrating generative ai with personality models. Electronics, 13(23), 4751. https://doi.org/10.3390/electronics13234751

[22]. Su, Y. (n.d.). LBSN_Dataset. GitHub. Retrieved from https://github.com/YijunSu/LBSN_Dataset

Appendix A

Model Training and Hyperparameter Settings

For reproducibility, the DRR deep learning modules were trained using the following parameters. The Multilayer Perceptrons (MLPs) for user and POI embeddings use two hidden layers with ReLU activations, each containing 128 neurons, and a dropout rate of 0.3 to prevent overfitting. The sequence modeling layer employs either an LSTM with hidden size 256 or a Transformer encoder with 4 attention heads and 2 encoder layers, depending on the experiment. All models were trained using the Adam optimizer with a learning rate of 0.001 and batch size of 64 for 50 epochs, with early stopping applied based on validation loss.

Additionally, the attention mechanism uses a scaled dot-product with normalization via softmax, and the route optimization module incorporates weighting parameters $\beta = 0.5$ for sequence importance versus static preference. All hyperparameters, including embedding dimensions for categorical and textual features, were chosen empirically through grid search on a held-out validation set.