Itinerary Planning with Real Time Data Integration of Weather and Traffic

Harshan K A
Computer Science Department
PES University
Banglore ,India
pes2202100518@pesu.pes.edu

K Bhargavi Reddy
Computer Science Department
PES University
Banglore,India
pes2202100801@pesu.pes.edu

Kaushik Challapalli
Computer Science Department
PES University
Banglore,India
pes2202101247@pesu.pes.edu

Kolisetty Venkata Aditya
Computer Science Department
PES University
Banglore,India
pes2202101154@pesu.pes.edu

Nazmin Begum
Computer Science Department
PES University
Banglore,India
nazmin.b@pes.edu

Abstract— — **Many people have to deal with the hassle of last minute planning in trips during all kinds of situations, without easy access to such vital details like local weather forecasts, town festivals, and better locations. The underlying question this project aims to address is the fact that travelers usually do not know exactly where they are going towards. The application developed in this project is designed for user ease under which users can input upcoming destinations. The platform will utilize in-time data to develop planning travel program that will include all the information related to weather, events, festivals and also route map to ensure users are guaranteed of a smooth journey.**

***Index Terms*— Itinerary Generation, Real Time Data; API's, K-Means Cluster Algorithm, Graph Neural Networks**

I. INTRODUCTION

In today's fast-paced world, individuals often struggle to find time for meticulous trip planning due to demanding schedules. The absence of critical details such as weather forecasts, local festivals, and recommended destinations can significantly impact their travel strategies and overall experience. Addressing this challenge, we propose a user-friendly travel planning application that leverages real-time data to generate organized and personalized itineraries based on user preferences and destinations.

Our project is built on the principle of enabling stress-free and efficient trip planning through advanced technologies. The application integrates live updates, Graph Neural Networks (GNNs) for real-time data analysis, and customized travel plans, ensuring users have access to comprehensive information. These include weather forecasts, upcoming events, traffic conditions, and tailored recommendations for attractions and activities. By consolidating this information, the platform minimizes the challenges associated with trip planning and enhances the user experience, allowing for seamless and enjoyable travel.

This paper outlines the design and implementation of the travel planning application, highlighting its capabilities in real-time data integration and user-centric itinerary generation. The proposed approach not only alleviates the logistical complexities of planning but also maximizes the value of trips by ensuring that users are well-informed and prepared.

II. LITERATURE SURVEY

In this paper, K-Means clustering and Graph Neural Networks (GNN) are utilized to optimize travel itineraries. In this section, the methods used in Itinerary Generation are reviewed, with specific emphasis on how K-Means clustering is employed for segmentation and how GNN is leveraged for itinerary scoring. Key differences and improvements in these approaches compared to existing methods are also highlighted.

A. Solution Representation

Before delving into the details of K-Means clustering and GNN-based approaches, the representation of travel itineraries is explained.

There are two primary ways of representing itineraries:

Point-Based Representation: Each Point of Interest (POI) is represented as a node in a graph. Features such as distance, rating, and time required for each POI are encoded as node attributes, while edges represent travel paths between POIs. This representation is particularly effective for applying GNN models.

Cluster-Based Representation: To structure the itinerary into days, POIs are grouped into clusters. Each cluster represents the POIs to be visited in a single day, ensuring that hard constraints (such as total travel time and priority POIs) are adhered to.

B. K-Means Clustering Based Approach

The K-Means algorithm is applied to split all POIs into clusters, where each cluster is a possible day in the travel plan.

Feature Engineering: Each POI is encoded into a feature vector which contains the coordinates, travel time, popularity, and user preferences.

Dynamically selection of K: Unlike fixed number of clusters, the value of k (number of days in the travel plan) is determined dynamically depending on the preferences of users, such as the number of free days and rate of travelling.

Iterative Refinement: The clusters obtained are fine-tuned so that each cluster satisfies constraints like the maximum time taken for travel and maximum budget per day. This refinement preserves the optimality whereas assuring feasibility of the grouping.

C. Graph Neural Networks (GNN)-Based Approach

The GNN model is employed to sequence the POIs within each cluster and score the itineraries based on user preferences and constraints.

Graph Construction: A graph is built for each cluster, with POIs as nodes and edges representing travel paths. Node attributes include features such as POI popularity, visit duration, and ratings, while edge weights encode travel distance and time.

Model Architecture: A GNN architecture (e.g., Graph Attention Networks) is used to propagate information across nodes, capturing both POI-level features and their relationships. This ensures that the model accounts for both individual POI importance and contextual relevance within the cluster.

Scoring and Optimization: The GNN predicts a score for each possible sequence within the cluster, helping to select the optimal sequence of POIs. The scoring function balances user preferences, time constraints, and the overall experience quality.

III. ARCHITECTURE OF THE SYSTEM

In this paper, we propose a travel planning portal, Journey Craft, that utilizes real-time information, including user inputs, locations, weather conditions, local traffic, and events, to generate personalized and optimized travel itineraries. The methodology involves several steps: data collection, algorithmic computation, system design, API integration, and user interface development.

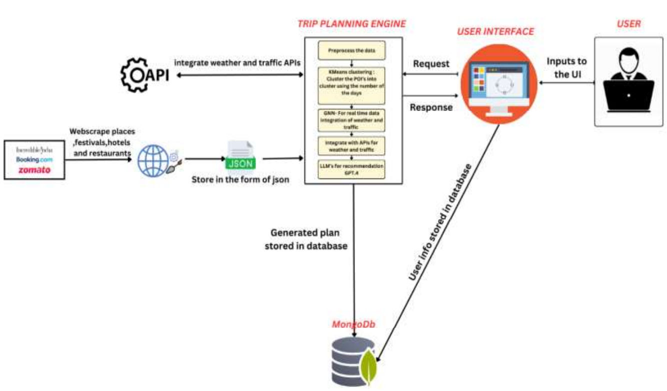

Figure 3.1 Architecture of Travel Planning Engine

A. Data

To create a dynamic and comprehensive travel planning system, data is sourced from both user inputs and external APIs.

User Inputs: Preferences such as travel dates, destinations, budget, and travel pace.

Real-time data is gathered from APIs such as:

- Google Maps API: Provides location services and distance estimations.
- Open-Meteo API: Provides Weather forecast and conditions.
- Geocoding API: Convert places into its geolocations.
- Places API: Get all the attractions of a place within a particular range.

B. Architecture

The architecture involves several key modules and classes that work together to generate a personalized itinerary plan with real time data of weather and traffic.

- The user interface allows users to input the travel destinations and preferences as required. The UI does this by forwarding the information to the Trip Planning Engine. The Trip Planning Engine contacts external APIs to get any live weather forecasts, local events etc. that a user will require for their trip.
- The Trip Planner Database stores travel plans or user preferences into the database while the User Verification layer ensures user authentication, authorization, and profiles management. User profiles and authentication tokens are kept in the database as part of the database management systems.
- Administrators collaborate with User Management System to manage user accounts, and sometimes there is a need to alter the data about hotels and restaurants.

C. Algorithm

The project utilizes a combination of clustering and optimization algorithms to create a travel itinerary based on user input, attractions, and traffic conditions. Here's a detailed explanation of the algorithm and its components:

- User Input Collection: Gather the number of days for the itinerary and a list of places the user wishes to visit.
- Geocoding: Convert the provided place names into geographic coordinates (latitude and longitude) using a geocoding API.
- Attraction Retrieval: For each location, fetch nearby tourist attractions using the Google Places API.
- Distance Calculation: Use the Google Distance Matrix API to calculate travel times between all attractions, taking into account current traffic conditions.
- Clustering of Attractions: Apply K-Means Clustering Algorithm to group attractions into clusters based on

their geographical proximity.

- Route Optimization: For each cluster, use the Nearest Neighbour heuristic to create an optimized route that minimizes travel time.
- GNN: For creating real time data graphs for weather and traffic.
- Visualization: Generate visual maps showing the optimized routes and clustered attractions for better user understanding.

IV. PROPOSED APPROACH

A. Data Collection and Integration

This phase involves gathering data from diverse sources to ensure comprehensive and up-to-date trip planning.

Data Sources:

- User inputs such as preferences and destinations.
- Public APIs (e.g., Google Maps, Open Weather, Event APIs).

B. Algorithm Design

K-Means Clustering:

- Clusters user-selected destinations based on their geographical proximity, forming day-wise travel plans.
- Helps optimize travel distance by grouping nearby POIs together.
- Ensures that clusters are balanced to maximize the coverage of attractions per day.

Graph Neural Networks (GNN):

- Represents the travel network as a graph where nodes are POIs and edges represent travel routes or distances.
- Learns relationships between destinations to predict optimal routes within a cluster.
- Dynamically adjusts itineraries by incorporating real-time factors like weather, traffic, and user preferences.

C. Optimization Technique

Itinerary Optimization with K-Means and GNN:

- K-Means groups destinations to minimize intra-cluster distances for efficient travel.
- GNN identifies the shortest and most suitable routes within clusters by leveraging graph structures.
- Considers dynamic factors like weather, traffic conditions, and user preferences.

Real-Time Updates:

- Continuously updates itineraries based on real-time data changes to ensure accuracy and reliability.

D. API Integration

Key APIs:

- Google Maps API: For navigation and travel distance estimation.
- Places API: Supplies details about specific attractions.
- Geocoding API: Converts locations into latitude and longitude for spatial analysis.
- Open-Meteo API: Delivers real-time weather data.

Request Optimization:

Manages API rate limits through efficient request strategies and caching mechanisms.

The combination of K-Means clustering for destination grouping and GNN for route optimization ensures an efficient and dynamic travel planning system. The methodology is designed to deliver accurate, personalized, and up-to-date itineraries while adapting seamlessly to real-time conditions.

V. IMPLEMENTATION

The initial steps of the implementation are as follows:

User Input Handling:

Implemented a user-friendly interface to collect the number of days for the itinerary and the desired attractions.

Developed input validation to ensure proper geographic data is collected.

Geocoding and Attraction Retrieval:

Integrated the Google Geocoding API to convert place names into latitude and longitude coordinates.

Utilized the Google Places API to fetch nearby tourist attractions based on user-provided locations.

Debugged API responses to handle cases where geocoding failed or no attractions were found.

Distance Calculation:

Employed the Google Distance Matrix API to calculate travel times between attractions, considering current traffic conditions.

Implemented batching of requests to adhere to API limits and avoid timeout issues.

Clustering Implementation:

Implemented K-Means clustering to group attractions based on geographic proximity.

Adjusted the number of clusters based on user input (number of days) and performed exploratory testing to determine the optimal clustering parameters.

Itinerary Optimization:

Developed a nearest neighbour algorithm to optimize the visiting order of attractions within each cluster.

Fine-tuned the algorithm to minimize overall travel time by considering traffic data.

Visualization:

Created interactive maps using Folium to display optimized itineraries and clustered attractions.

Implemented map markers for each attraction and visual routes to enhance user experience

VI. RESULTS

A. Efficiency

The system demonstrated robust performance in handling diverse datasets, particularly leveraging advanced algorithms for optimization.

K-Means Clustering:

Performed exceptionally well for large-scale datasets, effectively grouping Points of Interest (POIs) based on proximity and relevance.

Provided efficient clustering with minimal computational overhead, making it ideal for creating day-wise plans.

Graph Neural Networks (GNN):

Enhanced the optimization process by modeling relationships between POIs, resulting in more precise route planning.

Reduced redundant computations and improved the scalability of the system.

B. Real-Time Adaptability

Dynamic Updates:

Itineraries were recalibrated dynamically based on live updates from APIs such as Google Maps and Open Weather.

Enabled real-time adaptation to factors like traffic, weather changes, and event modifications.

Minimal Latency:

The architecture ensured low-latency updates, enhancing the user experience by providing accurate and timely information.

C. User Experience

Organized Itineraries:

Test users found the itineraries logical and easy to follow, appreciating the clear day-wise grouping of destinations.

Adaptive Features:

Users valued the system's ability to adjust to real-time changes and provide personalized recommendations.

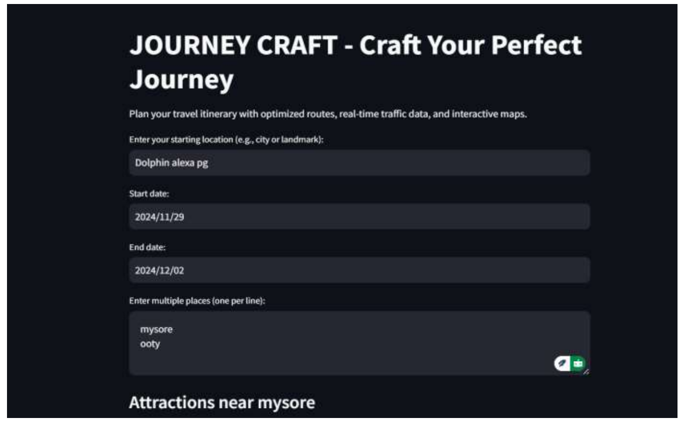

Figure 5.1 User Input to the System

This image represents the input page of your itinerary generation system, where users provide necessary travel details to plan their trips.

Figure 5.2 Attraction Selection

This image displays a list of tourist attractions near "Mysore", which is one of the destinations entered by the user. The system appears to retrieve a curated list of nearby attractions.

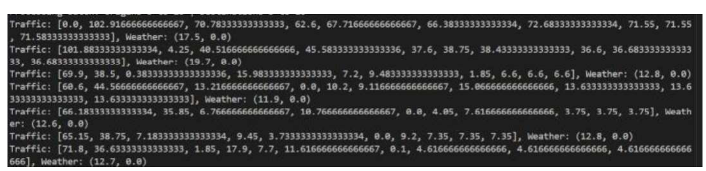

Figure 5.3 Traffic and Weather Constraints

This image displays the real time data of weather and traffic considerations, The system select the best route possible form the traffic matrix.

Traffic parameter is in the form of a matrix with possible routes and its traffic conditions. Low Value indicates that low traffic and high value indicates high traffic.

Weather is in the form of tuple of two values. The first is the temperature in degree celsius and the second is the precipitation in mm.

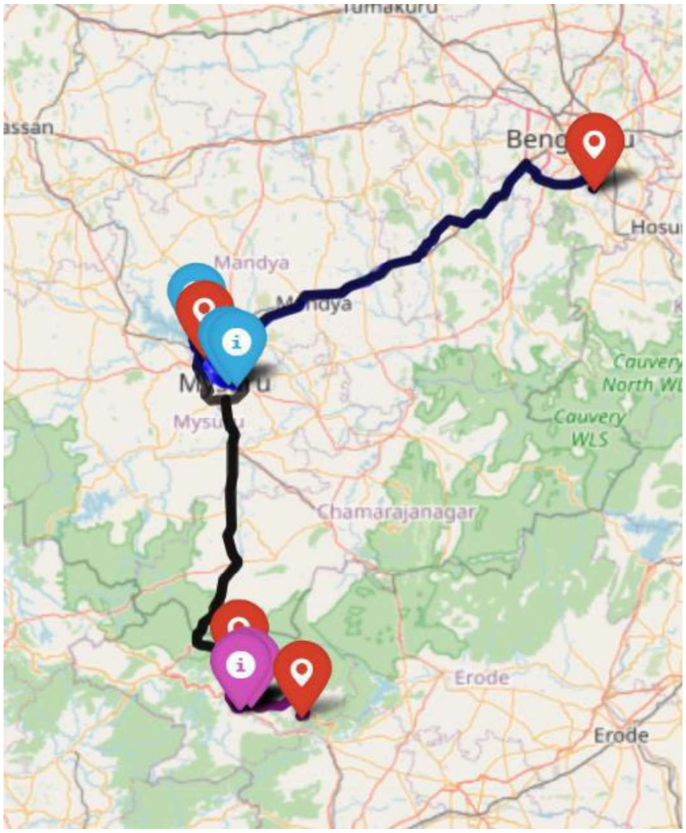

Figure 5.4 Itinerary Plan

This image shows the final plan generated. The output is in the form of maps created using folium

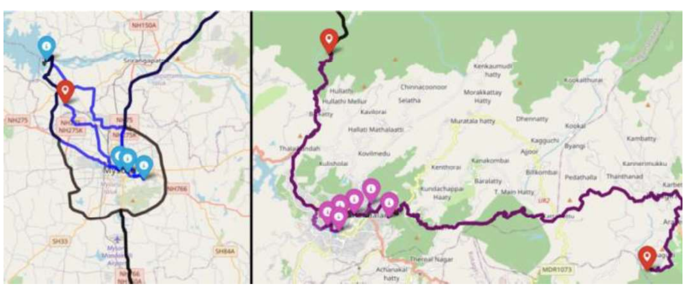

Figure 5.5 Clusters of attractions

This image shows the two clusters of the generated itinerary plan. The first is the Mysore cluster and the second one is the Ooty cluster.

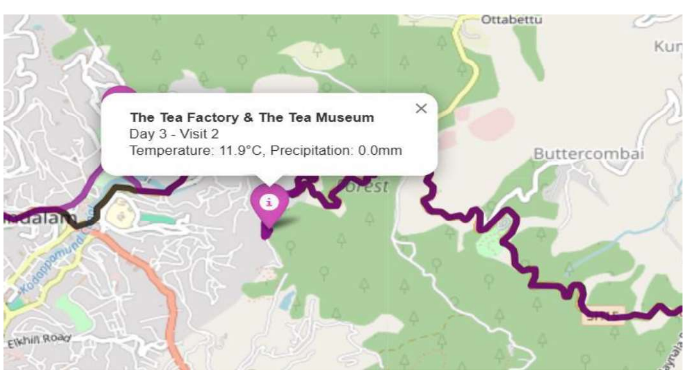

Figure 5.6 Attractions

This images shows the attractions in a cluster. Each attraction has a popup message when clicked on it. The name of the attraction, the day to be visited and the order in visiting that place is all shown in the popup. Also the weather conditions are displayed as seen in the image.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

In this paper we developed an optimized travel itinerary system that leverages Google Maps APIs to enhance the travel experience. By integrating key data sources such as geolocation, real-time traffic conditions, and weather information, we provided travelers with efficient routes and curated lists of attractions within their destination. Our approach utilized K-Means clustering algorithms, to group nearby points of interest, facilitating seamless and well-organized day plans. We have also used GNN for Real Time Data Integration. The combination of data-driven strategies and robust route optimization showcases the potential of using advanced algorithms and real-time updates to redefine travel planning.

Overall, Journey Craft successfully demonstrates the value of using data science and modern API integrations to tackle the complexities of itinerary optimization. The project is a step toward smarter, more efficient travel planning systems that cater to user preferences while optimizing time and convenience.

B. Future Work

The Journey Craft project opens the door to several avenues for future enhancements and improvements:

Enhanced Personalization: Incorporating user preferences and historical data to provide more personalized recommendations for attractions, restaurants, and activities based on past behavior and interests.

Machine Learning for Predictive Analysis: Utilizing machine learning models to predict traffic conditions, wait times at attractions, or even forecast popular times to visit certain locations. This would further refine the recommendations provided by the system.

Integration with Local Services: Expanding the system to include local transportation options, restaurant reservations, and ticket booking for attractions to make the travel planning experience even more comprehensive.

Interactive User Experience: Developing a more interactive and intuitive user interface that allows travelers to customize their itineraries easily, receive notifications about route changes, or get updates on weather and traffic.

Exploring More Advanced Algorithms: Experimenting with more sophisticated algorithms, such as reinforcement learning, for continuous itinerary adjustments based on real-time user feedback and dynamic environmental conditions.

REFERENCES

[1] Bin, G., Wei, L., & Zong, L. H. (2020, November). Multi-agent itinerary planning based Voronoi dividing and Discrete multi-objective Particle swarm optimization. In *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)* (pp. 1-5). IEEE.

[2] Tenemaza, M., Luján-Mora, S., De Antonio, A., & Ramirez, J. (2020). Improving itinerary recommendations for tourists through metaheuristic algorithms: an optimization proposal. *IEEE Access*, *8*, 79003-79023.

[3] Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert systems with applications*, *207*, 117921.

[4] Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., ... & Su, Y. (2024). Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*.

[5] Jia, J., Chen, Y., Liu, Y., & Khamis, A. (2022, March). Trip Itinerary Planning: A Bio-inspired Metaheuristic Approach. In *2022 IEEE International Conference on Smart Mobility (SM)* (pp. 32-37). IEEE.

[6] Mo, B., Xu, H., Zhuang, D., Ma, R., Guo, X., & Zhao, J. (2023). Large language models for travel behavior prediction. *arXiv preprint arXiv:2312.00819*.

[7] Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., ... & Yu, J. (2020, April). Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020* (pp. 1082-1092).

[8] Zhang, L., Xu, J., Pan, X., Ye, J., Wang, W., Liu, Y., & Wei, Q. (2023). Visual analytics of route recommendation for tourist evacuation based on graph neural network. *Scientific Reports*, *13*(1), 17240.

[9] Liang, Y., Ding, F., Huang, G., & Zhao, Z. (2023). Deep trip generation with graph neural networks for bike sharing system expansion. *Transportation Research Part C: Emerging Technologies*, *154*, 104241.

[10] Jin, G., Wang, M., Zhang, J., Sha, H., & Huang, J. (2022). STGNN-TTE: Travel time estimation via spatial–temporal graph neural network. *Future Generation Computer Systems*, *126*, 70-81.

[11] Zhu, G., Cao, J., Chen, L., Wang, Y., Bu, Z., Yang, S., ... & Wang, Z. (2023). A multi-task graph neural network with variational graph auto-encoders for session-based travel packages recommendation. *ACM Transactions on the Web*, *17*(3), 1-30.

[12] Amir, H., Gaibie, A., & Hammerschlag, D. Weather Prediction using Spatial-Temporal Graph Neural Networks.