
Journal of Informatics and Web Engineering

Vol. 5 No. 3 (October 2026)

eISSN: 2821-370X

An Intelligent Tourism Recommender System Using Retrieval-Guided Large Language Models

Wei-Jun Lim¹, Nabhan D. Salih², Hairulnizam Mahdin³, Ammar Alazab^{4*}

¹Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia.

²Department of Computer Science and Engineering, Ohio State University, Columbus, OH 43210-1210, USA

³Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Persiaran Tun Dr. Ismail, 86400 Parit Raja, Johor, Malaysia.

⁴Center for Artificial Intelligence, Research and Optimisation (AIRO), Torrens University, 88 Wakefield St, Adelaide SA 5000, Australia

**corresponding author: (ammar.alazab@torrens.edu.au; ORCID: 0000-0001-9443-937X)*

Abstract - Travel planning has become more complex nowadays due to the information overload from numerous online platforms causing travelers to be overwhelmed when selecting destinations, accommodations, and activities that match their preferences and constraints. Traditional tourism recommender systems usually provide static and generic suggestions that fail to match users' needs and cannot understand natural user language. This research develops and evaluates a Generative AI Chatbot for Tourism Recommendations that use Large Language Models (LLMs) to generate personalized travel itineraries through natural language interaction to solve these limitations. The system can process user natural language inputs including travel origin and destination, trip duration and dates, and budget in a conversational way. This research used the TravelPlanner dataset which contains real-world tourism information on attractions, restaurants, accommodations, and flights. The chatbot uses carefully designed prompts with strict generation rules to produce day-by-day itineraries that match user constraints. The preliminary results show the system's capability to generate human-readable travel plans that include correct flight selections, accommodation recommendations, and structured daily activities. The chatbot prototype also supports conversational refinement which allows users to modify their itineraries interactively. The evaluation was conducted using Bilingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence (ROUGE-L) metrics to measure response quality and achieving a BLEU score of 0.59, ROUGE-L of 0.68, and constraint accuracy of 93%. This research has contributed to AI-driven tourism planning by showing how generative AI can be used to create interactive and context-aware assistants in recommender systems. Future work will focus on including external APIs and conducting a comprehensive user testing to measure the usability and satisfaction level in the real world.

Keywords—Generative AI, Tourism Recommendation, Large Language Model, Travel Itinerary Generation, Conversational Recommender System, Retrieval-Guided Generation, Natural Language Processing

Received: 14 June 2025; Accepted: 23 December 2025; Published: 16 October 2026

This is an open access article under the [CC BY-NC-ND 4.0](#) license.



1. INTRODUCTION

In today's digital era, travel has become an important part of modern lifestyle for business, tourism, or personal growth. Travelers have to face a large amount of information including, hotels, flights, attractions, and user reviews due to the quick expansion of online travel platforms. While this amount of information gives travelers more choices and flexibility when planning their plan, it also causes information overload [1]. Many travelers spend their time on comparing the prices, the reviews, and organize their plan across different websites which makes the trip planning process time consuming and also complicated.

A well-planned travel itinerary is important to have nice travel experiences but different travelers might have different preferences while some will prefer shopping, others might prefer adventure. Traditional recommender systems such as content-based filtering, collaborative filtering, and hybrid methods, are available to help travelers in making decisions [2], [3]. Although these systems suggest places based on user behavior and past data, they usually produce static recommendations which limit personalization [4], [5], [6]. They also do not support conversational interfaces which make them less flexible to adapt to changes in user preferences over time [7].

The Large Language Models (LLMs) being as one of the improvements in current's generative artificial intelligence, it served as a way to improve recommendations in travel systems [8], [9]. Generative AI chatbots can understand different user inputs in natural language and are able to respond in a human-like way. These systems able to create and adjust different personalized travel itineraries that fulfill the user needs and preferences in the tourism industry [10], [11]. The chatbots can also generate more accurate and relevant suggestions by having real-time API connections for flights and weather conditions. By having generative AI, it makes travel planning more flexible, interactive, and also personalized when compared to traditional recommender system [12], [13].

Although several studies have explored AI-based recommender systems [14], many still focus on traditional collaborative filtering or hybrid models. These models cannot communicate with users in a conversational way, real-time data, or refine recommendations through dialogue. In addition, there is also limited research on comparing which LLM-based itinerary generation strategies are able to produce the most accurate and user-friendly travel plans [15].

To solve these gaps, this study proposes a Generative AI chatbot for tourism recommendations. The system combines natural language processing, real-world datasets, and external APIs to generate personalized travel itineraries based on user input such as budget, destination, travel duration, and preferences. The main contributions of this study are:

- Design a conversational travel recommender system that understands user preferences in natural language
- Implement and comparing different LLM-based itinerary generation strategies
- Integrate real-time APIs to improve recommendation accuracy and relevance
- Evaluating usability and recommendation quality of the system

2. LITERATURE REVIEW

Udandara et al. [16] proposed Roamify, an LLM-based Google Chrome extension designed for personalized itinerary planning. The system generates day-by-day travel plans based on user inputs such as destination, travel duration, and preferences. The system uses web scraping techniques to obtain up-to-date travel information from websites to improve the recommendation relevance. The system was evaluated based on the user studies to evaluate the planning efficiency and satisfaction. Participants reported that the system helps them to save time on travel planning while also offering the recommendations that are matched with their preferences. The results show that most users preferred LLM-generated itineraries because they were more detailed and matched user preferences. However, the system relies heavily on the web scraping data which has a possibility to recommend incorrect or outdated information. Additionally, some recommendations may miss important travel data because the model is trained on general language data not on tourism-based content data.

Liu et al. [17] introduced Vaia which is a multi-agent LLM-based travel planning framework. Voyage divides the travel planning tasks into different parts and specialized agents are in charge of each part such as intent recognition, constraints validation, route construction, and API integration. The system was evaluated by the human-in-the-loop and the results show that the full systems have a higher feasibility and realism scores compared to ablated versions of the system. The study highlights that the structured planning and external API integrations can help in improving itinerary quality. However, the framework requires complex agent coordinates and a strong technical framework which increase the resources needed when deploying.

Otaki and Baba [18] proposed an interaction-based itinerary recommendation framework that uses user edits as augmented feedback data for model refinement. Through interactive training, the system improves suggestions accuracy by converting itinerary changes into ranking signals. The system has been experimented on multiple city datasets and the results show consistent improvement in top-k recommendation performance. Although the framework enhances the personalization through feedback learning, the performance was evaluated in an offline setting so the performance on real-world remains uncertain.

Banerjee et al. [19] proposed a framework called SynthTRIPS that uses LLMs to generate synthetic travel queries for tourism recommendation research. The main goal of this framework is to solve the problem of limited and less diverse tourism datasets. The system creates travel queries based on different user personas and uses a structured knowledge base to make the queries more realistic. The framework was evaluated using automatic metrics and human expert reviews. The results show that SynthTRIPS can generate realistic and varied travel queries and it is useful for training and testing recommender systems. However, this framework focuses on generating travel queries instead of building a conversational travel recommendation system.

Fajari and Baizal [20] developed a chatbot for culinary tourism recommendations using simple natural language processing methods. The system used Named Entity Recognition and content-based filtering techniques such as TF-IDF and cosine similarity to match user preferences with restaurant information obtained from the Google Places API. The chatbot was then integrated into Telegram, and user testing on the chatbot showed good usability. Although this study shows that it is possible to build a tourism recommendation chatbot, the system mainly depends on similarity matching. This chatbot also did not use generative AI, which limited its ability to generate complex multi-days itineraries.

In addition to system-level approaches, recent research has also focused on travel-specific datasets for itinerary generation. The TravelPlanner dataset in [21] provides structured tourism data including attractions, accommodations, transportation, and trip constraints to support itinerary generation tasks. The study shows that using structured datasets together with proper evaluation methods can improve LLM-based itinerary generation. This helps the system to generate more realistic and domain-specific travel plans compared to web-scraped LLM outputs. However, most existing studies mainly focus on evaluating models rather than building an interactive conversational recommender system. In addition, there is still limited research on combining structured datasets, real-time APIs, and different itinerary generation strategies into a single chatbot framework.

Although previous studies have explored traditional chatbots, LLM-based itinerary generation, synthetic data benchmarking, and multi-agent systems, but limited research has developed a complete conversational tourism recommender system that combines structured datasets, real-time API integration, comparison of different itinerary generation strategies, automatic evaluation metrics, and user satisfaction evaluation. However, existing works often focus on only one or two aspects without fully integrating all these components into a single system, resulting in a lack of comprehensive system design and evaluation. Especially for the automatic evaluation metrics and user satisfaction, most works focus on only one, even though both are important for assessing the system performance and user experience.

3. RESEARCH METHODOLOGY

This section explains the research methodology to design and develop a Generative AI Chatbot for Tourism Recommendation. The methodology follows a structured and chronological process which begins with research design and theoretical foundation, followed by data collection, system design, algorithm development, and lastly the system evaluation. This structured method helps to ensure that the system is practical in the real-world environment.

3.1 Research Design and Theoretical Framework

This research uses a system development based on a generative recommender system and transformer language models [22]. Unlike traditional recommender systems, which mainly rely on collaborative filtering or similarity-based matching which make suggestions from fixed lists, generative recommender systems can generate day-by-day travel itineraries based on user preferences and contextual information.

LLMs are the main part of the system. They use a transformer structure which has a self-attention mechanism that helps the model to understand the meaning of words and the relationship between them [23], [24]. This helps the system to understand words and focus on the information that are important such as destination, numbers of days, and budget when creating an itinerary. The scaled dot-product attention mechanism is defined as Equation (1).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (1)$$

where Q (query), K (key), and V (value) are vectors obtained from the input text, and the d_K represents the dimension of the key vectors. This mechanism allows the models to identify which words or phrases are most important when understanding user requests.

In addition to attention, LLMs are also trained using an autoregressive language modeling objective; it aims to predict the next word based on previously generated words [23]. This helps the model to learn natural language patterns which later generate fluent, human-like responses during the conversation. The autoregression objective is shown in Equation (2).

$$\log p(x) = \sum_{t=1}^T \log p(x_t | x_{<t}) \quad (2)$$

To evaluate the system performance, both automatic metrics and user-based evaluation metrics will be used. The automatic metrics such as Bilingual Evaluation Understudy (BLEU) [25] and Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence (ROUGE-L) [26] are used to measure the similarity and information quality of the generated itinerary compared to the reference plans. User-based evaluation metrics, such as user satisfaction survey is used to measure the usability and the practical value in real travel planning scenarios.

3.2 Data Acquisition

The system uses the TravelPlanner dataset that was obtained from HuggingFace, the `osunlp/TravelPlanner` dataset which contains real-world tourism information. Each record in the dataset contains attributes such as origin and destination city, days and dates of travel, budget, user queries, and semi-structured reference tourism information. The semi-structured reference tourism information contains destination information such as attractions, accommodations, restaurants, and flight details. The dataset contains three splits which are train, validation, and test split while this study will use the test split, which has 1000 records, to simulate real-world itinerary generation scenarios. The attribute descriptions of the TravelPlanner dataset are in Table 1.

3.3 System Architecture

The proposed system has six main parts that work together to create personalized travel itineraries. The user query input module takes the user travel requests in natural language then the user information extraction module will use LLM-based prompts to get details such as origin, destination, travel dates and duration, and budget. The dataset filtering and scoring module compares the user profile information with each dataset record using a rule-based scoring method to score each record. The reference information parsing module will then select the highest score record and organize the record data into attractions, restaurants, accommodations, and flights. The prompt construction module builds a controlled generation prompt with the structured data and itinerary rules and is sent to the itinerary generation module to generate the itinerary. Lastly, the conversational refinement loop allows users to modify their itineraries if they are not satisfied with.

3.4 Data Preprocessing

Instead of performing heavy machine learning preprocessing, this research applies lightweight preprocessing during runtime. The preprocessing steps include budget extraction, record filtering and scoring, and reference information parsing.

Table 1. Attribute Descriptions of the TravelPlanner Dataset

Attribute Name	Details	Field Type	Dataset Availability (set)
org	The city where the user starts their trip	string	Train, Validation, Test
dest	The city the user wants to travel to	string	Train, Validation, Test
days	Number of days for the trip	int64	Train, Validation, Test
visiting_city_number	Number of cities visited during the trip	int64	Train, Validation
date	The dates for the entire trip	string	Train, Validation, Test
people_number	Number of travelers	int64	Train, Validation
local_constraint	User-specified constraints or preferences for local travel	string	Train, Validation
budget	User's budget for the trip	int64	Train, Validation, Test
query	User's travel request in natural language	string	Train, Validation, Test
level	Difficulty of the query	string	Train, Validation
annotated_plan	Reference itinerary showing the day-by-day travel plan	string	Train
reference_information	Structured tourism data for a destination	string	Train, Validation, Test

3.4.1 Budget Extraction

The numerical budget value in the user query in each dataset record is extracted and stored into a new field in the dataset. This budget is used to compare with the user input query to see if this record in the dataset matches with the user budget. Figure 1 shows the process of budget extraction.

```
def extract_budget_from_text(text):
    match = re.search(r"\$(\d,)+", text)
    if match:
        return int(match.group(1).replace(",", ""))
    return None

def preprocess_dataset(dataset):
    processed = []
    for record in dataset:
        budget = extract_budget_from_text(record.get("query", ""))
        record["extracted_budget"] = budget
        processed.append(record)
    return processed
```

Figure 1. Budget Extraction Process

3.4.2 Record Filtering and Scoring

Each record in the dataset will be scored and filtered based on the user profile extracted by the LLM. The scoring is performed by matching the origin, destination, trip duration, travel dates, and budget between the user input and each dataset record. A match of destination or origin will add 3 points for each, a match of budget or duration of the trip will add 2 points for each, and a match of travel start dates or end dates will add 1 point for each. Then, only the record with the highest score was used to provide contextual input to the language model. Figure 2 shows the process of scoring each record in the dataset.

3.4.3 Reference Information Parsing

After the highest score record has been found, the system groups the semi-structured reference information for that record into different categories such as attractions, restaurants, accommodations, and flights. Then each category will

be further parsed into structured arrays to support controlled prompts suggestions. This structured representation helps the language model generate realistic itineraries and avoid giving incorrect information. Figure 3 shows the process of information parsing.

```

for record in dataset:
    score = 0
    if record.get("dest", "").lower() == user_destination:
        score += 3
    if record.get("org", "").lower() == user_origin:
        score += 3
    if record.get("days") == user_days:
        score += 2
    record_budget = record.get("extracted_budget")
    if user_budget is not None and record_budget is not None:
        if abs(user_budget - record_budget) / record_budget <= 0.2:
            score += 2
    record_dates = record.get("date", [])
    if isinstance(record_dates, str):
        try:
            record_dates = ast.literal_eval(record_dates)
        except:
            record_dates = []
    if user_start_date and user_end_date and record_dates:
        try:
            record_start = datetime.strptime(record_dates[0], date_format).date()
            record_end = datetime.strptime(record_dates[-1], date_format).date()
            user_start = datetime.strptime(user_start_date, date_format).date()
            user_end = datetime.strptime(user_end_date, date_format).date()
            if record_start == user_start:
                score += 1
            if record_end == user_end:
                score += 1

```

Figure 2. Scoring Process of Each Record

```

def extract_reference_info(reference_info):
    grouped_data = defaultdict(list)

    for item in reference_info:
        description = item.get("Description", "").strip().lower()
        content = item.get("Content", "").strip().lower()

        if not description or not content:
            continue

        lines = [line.strip() for line in content.split("\n") if line.strip()]
        grouped_data[description].extend(lines)

    return dict(grouped_data)

```

Figure 3. Process of Information Parsing

3.5 Itinerary Generation Strategy

The proposed system generates itineraries using a retrieval-guided method with a LLM without the need of fine-tuning. However, carefully designed prompts are used to guide the model to produce itineraries that are realistic, accurate, and user-centre. These prompts include structured user profile, selected tourism data from the dataset, and strict rules such as the number of travel days, no repetition on restaurants and attractions, budget limits, and correct placement of outbound and return flights. Stylistic instructions were also included to make the generated itineraries sound natural and human-like.

The LLM then generates a daily itinerary with planned activities, meals, accommodations, and transportation in a structured way. Users can also make changes to their generated itinerary if they are not satisfied with conversational refinement. This method combines structured data and generative AI to generate personalized itinerary and it shows it is flexible and reliable for personalized travel planning.

An example of the constructed prompt is explained below to better show the itinerary generation strategy. With this structure prompt, the LLM can generate a complete itinerary that match with user preferences using the provided data in the dataset.

Example User Input:

“Plan a 3-day trip from Sarasota to Chicago with a budget of \$1900.”

Constructed Prompt:

- User Profile:
 - Origin: Sarasota
 - Destination: Chicago
 - Days: 3 days
 - Budget: \$1900
- Available Data:
 - Attractions: Navy Pier, Art Institute of Chicago, Millennium Park
 - Restaurants: Giordano’s, Lou Malnati’s
 - Accommodation: Hilton Chicago
 - Flights: Sarasota -> Chicago, Chicago -> Sarasota
- Constraints:
 - Do NOT invent attractions, restaurants, accommodations, or flights
 - Generate exactly {Days} itinerary
 - Do NOT repeat restaurants or attractions
 - Place departure flight on Day 1 and return flight on Day 3
- Stylistic Instructions:
 - Use natural and conversational tone
 - Organize output by Day 1, Day 2, Day 3
 - Vary sentence structure

3.6 Overall System Workflow Diagram

Figure 4 shows the overall workflow diagram that has been described in detail in the earlier sections.

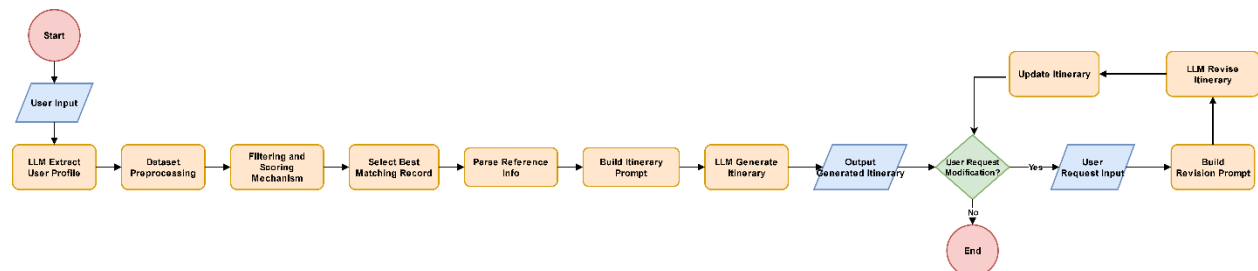


Figure 4. Overall System Workflow

3.7 Pseudocode

Algorithm 1 is the pseudocode for the overall system workflow.

Algorithm 1

Input: Natural language travel query from user

Output: Generated or revised structured travel itinerary

- 1: Import required libraries (datasets, requests, json, re, datetime, streamlit)
- 2: Load TravelPlanner dataset (test split)
- 3: Initialize NVIDIA NIM API configuration
- 4: Define USER_SCHEMA structure
- 5: Initialize Streamlit session state
- 6: context = None
- 7: chat_history = []
- 8: Function build_extraction_prompt(user_quer)
- 9: Construct prompt to extract: origin, destination, days, start_date, end_date, budget

```

10:     Return structured extraction prompt
11: End Function
12: Function extract_user_info_via_llm(user_query) // extract user info into a
                                     structured form
13:     prompt ← build_extraction_prompt(user_query)
14:     Send prompt to LLM via API and receive JSON response
15:     Parse JSON response into user_profile
16:     Return user_profile
17: End Function
18: Function preprocess_dataset(dataset)
19:     For each record in dataset:
20:         Extract budget from record query text
21:         Store extracted_budget in record
22:     END FOR
23:     Return processed dataset
24: End Function
25: Function find_matching_records(dataset, user_profile)
26:     Initialize empty match list
27:     For each record in dataset:
28:         score ← 0
29:         IF destination matches → score += 3
30:         IF origin matches → score += 3
31:         IF number of days matches → score += 2
32:         IF budget within ±20% → score += 2
33:         IF start_date matches → score += 1
34:         IF end_date matches → score += 1
35:         IF score > 0 → append (score, record) // save the score and
                                     record
36:     END FOR
37:     Sort matches in descending order
38:     Return ranked match list
39: End Function
40: Function extract_reference_info(reference_information)
41:     Group reference content by description type
42:     Return structured grouped_reference
43: End Function
44: Define Parsing Functions
45:     parse_attraction()
46:     parse_restaurant()
47:     parse_accommodation()
48:     parse_flight()
49: Function select_flight(flights, user_requested_hour, mode)
50:     IF user_requested_hour exists:
51:         Return flight matching requested hour
52:     ELSE IF mode = earliest:
53:         Return earliest departure flight
54:     ELSE IF mode = latest:
55:         Return latest departure flight
56:     ENDIF
57: End Function
58: Function build_itinerary_prompt(user_profile, structured_data)
59:     Insert structured user profile
60:     Insert selected attractions, restaurants, accommodation
61:     Insert outbound and return flight
62:     Enforce strict generation rules
63:     Apply writing style constraints
64:     Return generation prompt
65: End Function
66: Function generate_itinerary_via_llm(prompt)
67:     Send structured prompt to LLM
68:     Return generated itinerary text
69: End Function
70: Function run_itinerary_pipeline(user_query)
71:     user_profile ← extract_user_info_via_llm(user_query)
72:     dataset_processed ← preprocess_dataset(dataset)
73:     matches ← find_matching_records(dataset_processed, user_profile)
74:     Select highest score record
75:     Parse record reference_information

```

```

76:      Extract attractions, restaurants, accommodation, flights
77:      Select suitable outbound and return flights
78:      itinerary ← generate_itinerary_via_llm()
79:      Return itinerary and structured context
80: End Function
81: Streamlit App:
82:      Display title: "Travel Itinerary Planner Chatbot"
83:      Input box: user_input ← st.text_input("Enter your travel request:")
84:      IF submit button pressed AND user_input exists:
85:          IF st.session_state.context is None:
86:              result ← run_itinerary_pipeline(user_input)
87:              Store result in st.session_state.context
88:              Store user_input and itinerary to chat_history
89:          ELSE:
90:              Build revision prompt include current itinerary, user
              modification request
91:              Send revision prompt to LLM
92:              Update stored itinerary in context
93:              Store user_input and updated_itinerary to chat_history
92:          ENDIF
93:      END IF
94: Display chat history in Streamlit:
95:      For each sender and message in chat_history:
96:          IF sender = "You":
97:              Display user message
98:          ELSE:
99:              Display bot message
100:      End For

```

Line 1 to line 3 is to import libraries, load dataset, and set up the LLM AI model. The line 4 set up a user profile for LLM to extract from the user query and the line 5 to 7 set up the session to store itineraries as history chat. The line 8 is a function to build up what information should be extracted from the user query by the LLM model. The second function on line 12 is used to call the first function to extract user information into a structured user profile using LLM model.

The third function on line 18 is to extract the budget from the dataset record and store it, then later it will compare with the user query budget. The fourth function on line 25 is to find the matches record by using a scoring and filtering mechanism to see which record matches with the user query. The function on line 50 is used to group the reference information by categories of the selected record. Then the parsing function is to convert the raw data in the record for each attraction, accommodations, flights, and restaurants into an array and use it to generate an itinerary later.

The line 49 function is used to select the flight time that matches the user preferences. The line 58 function is used to build up how the itinerary should be generated with the provided data and some strict rules and writing style instructions. The line 66 is a function used to generate the itinerary using the LLM model by getting the information from the line 58 function.

Line 70 is the function showing the complete process execution stages. The line 81 shows how the interface is built using Streamlit and line 94 shows how the itinerary is being displayed in the Streamlit.

4. RESULTS AND DISCUSSIONS

4.1 Experimental Setup

To evaluate the proposed retrieval-guided itinerary generation framework, we randomly select 80 travel queries from the TravelPlanner dataset to test the system. Two type of itineraries were generated for each travel query which are the proposed retrieval-guided LLM and the pure LLM baseline which without retries, filtering or constraints enforcement.

The generated itineraries were evaluated using automatic metrics including BLEU and ROUGE-L against the referenced itineraries from the dataset. BLEU measures the n-gram similarity between the generated itinerary and the reference itinerary to show how close the generated text matches with the expected outputs. ROUGE-L evaluates the longest common subsequence between texts to capture content overlap and recall, this metric is suitable for longer response such as travel itineraries.

In addition, constraint accuracy was also calculated based on the duration consistency, budget, non repetition on restaurants or attractions, and correct flight placement.

4.2 Quantitative Evaluation

Table 2 shows the comparison between the proposed retrieval-guided method and the pure LLM baseline. The results show that the proposed method outperforms the pure LLM baseline across all evaluation metrics. The proposed method achieved a BLEU score of 0.59 compared to 0.41 for the baseline and a ROUGE-L score of 0.68 compared to 0.52. Moreover, the proposed system achieved 93% of constraints accuracy which is higher than the baseline 69%. These results show that structured dataset filtering and rule-based prompt control effectively reduce the logical inconsistencies.

Table 2: Comparison Table on Various Metrics

Method	BLEU	ROUGE-L	Constraints Accuracy
Pure LLM	0.41	0.52	69%
Proposed RG-LLM	0.59	0.68	93%

4.3 Constraint Satisfaction Analysis

Further analysis shows that the pure LLM baseline often fails to satisfy structured planning constraints. Although the generated itineraries were fluent, logical inconsistencies were observed, including incorrect flight placement, repeated attractions, incorrect travel duration, and exceeding the budget. These issues show that unconstrained language models do not always have proper control to follow numbers- and time-related requirements accurately in travel planning tasks.

The proposed retrieval-guided system achieved much higher compliance with user constraints. By using structured tourism data and clear prompt rules, the system reduces logic errors and improves consistency in travel planning. These results show that using data retrieval and the controlled text generation method is a more reliable method than just using a pure LLM.

4.4 Qualitative Evaluation

Figure 5 shows the developed chatbot prototype interface. The interface allows users to submit their travel requests in natural language and refine the generated itinerary through interactive conversation.

Travel Itinerary Planner Chatbot

Enter your travel request:

Submit

Figure 5. Chatbot Interface

To evaluate the system performance on generating an itinerary, a user query in the dataset was selected as the input which is "Please plan a trip for me starting from Sarasota to Chicago for 3 days, from March 22nd to March 24th, 2022. I need a cheap trip as I only have a budget of \$1,900". The system successfully extracted the information needed from the query, such as destination, origin, days, dates, and budget. Then, based on the scoring and filtering mechanism in Section 3, the highest score record is selected and parsed into structured tourism components. Figure 6 shows the full itinerary generated.

👤 **You:** Please plan a trip for me starting from Sarasota to Chicago for 3 days, from March 22nd to March 24th, 2022. I need a cheap trip as I only have the budget of \$1,900.

🤖 **Bot:** Here is your 3-day travel itinerary to Chicago:

Day 1: 2022-03-22

Morning: Start your day with a delicious breakfast at The Black Pearl, a cozy cafe and bakery with an average cost of \$63 and a rating of 4.1. Then, head over to Navy Pier, a lakefront attraction that offers stunning views of Lake Michigan and the city skyline. Take a leisurely walk along the pier and enjoy the street performers and live music.

Afternoon: For lunch, grab a bite at Pantry D'or, a unique eatery that serves a mix of tea, BBQ, fast food, cafe, and Indian cuisine with an average cost of \$53 and a rating of 4.4. After lunch, visit the Skydeck Chicago, which offers breathtaking views of the city from the 103rd floor of the Willis Tower.

Evening: For dinner, head to Fio Cookhouse and Bar, a restaurant that serves a variety of desserts, Mexican, fast food, cafe, and seafood dishes with an average cost of \$31 and a rating of 4.0. End your evening with a visit to Millennium Park, a beautiful urban park that features iconic landmarks like the "Cloud Gate" sculpture and the Jay Pritzker Pavilion.

Accommodation check-in: Newly Renovated Greenpoint Abode, Entire home/apt, \$981.0, Rating: 3.0

Flight info: Outbound flight F3984576, departing Sarasota at 05:14 and arriving in Chicago at 06:50, price \$279.

Day 2: 2022-03-23

Morning: Begin your day with a visit to the Lincoln Park Zoo, a free zoo that's home to over 1,100 animals from around the world. Then, grab breakfast at Gyan Vaishnav, a bakery and dessert shop with an average cost of \$14 and a rating of 3.9.

Afternoon: For lunch, head to Bro's Kitchenette, a casual eatery that serves Mexican, desserts, and seafood dishes with an average cost of \$38 and a rating of 3.3. After lunch, take a stroll along the Riverwalk, a scenic pedestrian path that runs along the south bank of the Chicago River.

Evening: For dinner, try Mini's Royal Cafe, a tea room and bakery with an average cost of \$72 and a rating of 3.3. End your evening with a visit to Buckingham Fountain, a beautiful landmark that features a large fountain and stunning views of the city skyline.

Day 3: 2022-03-24

Morning: Spend your morning exploring Grant Park, a large urban park that's home to several famous landmarks, including the Buckingham Fountain and the Art Institute of Chicago. Then, grab lunch at Starbucks, a popular coffee shop with an average cost of \$46 and a rating of 3.8.

Afternoon: Visit the Shedd Aquarium, a world-class aquarium that's home to over 32,000 animals, including dolphins, beluga whales, and penguins. Then, head back to your accommodation to pick up your bags and head to the airport for your return flight.

Flight info: Return flight F4010566, departing Chicago at 20:10 and arriving in Sarasota at 23:49, price \$368.

Figure 6. Full Itinerary

The generated itinerary shows that the system successfully meets multiple user requirements at one time which include the itinerary duration matches with user requests, flight departure and return dates are placed correctly, attractions and

restaurants are not repeated, and budgets are within the range given by the user. These results show that the prompt-controlled generation strategy successfully combines structured dataset information with the LLM natural language generation ability.

The same query was also input into a pure LLM baseline system to compare with the proposed method. This baseline system generate itineraries without dataset retrieval, filtering, or constraint enforcement. The baseline output frequently contained incorrect information, repeated attractions or restaurants, lack of information for the flights, and exceeding the budgets. This shows that unconstrained LLM generation alone cannot reliably satisfy multiple user requirements. In contrast, the retrieval-guided system met all the constraints, showing that using structured data and strict prompt rules makes itinerary planning more reliable and accurate. Figure 7 shows the full itinerary generated by the baseline system.

 **You:** Please plan a trip for me starting from Sarasota to Chicago for 3 days, from March 22nd to March 24th, 2022. I need a cheap trip as I only have the budget of \$1,900.

 **Bot:** Welcome to your 3-day travel itinerary to Chicago!

Day 1: March 22, 2022

- **Morning:** Fly from Sarasota-Bradenton International Airport (SRQ) to Chicago O'Hare International Airport (ORD) on a morning flight. I've booked a round-trip economy ticket on American Airlines for \$450.
- **Check-in:** After arriving in Chicago, take a taxi or Uber to your hotel, the Kimpton Hotel Monaco Chicago (approximately \$150 per night). This stylish hotel is located in the heart of the Loop, within walking distance to many attractions.
- **Lunch:** Head to Lou Mitchell's, a Chicago institution since 1923, for a classic American breakfast or lunch. Try their famous waffles or eggs benedict. (20—30 per person)
- **Afternoon:** Visit the Art Institute of Chicago, one of the top art museums in the country. Spend a few hours exploring the impressive collection, including works by Monet, Van Gogh, and Picasso. (\$20 per person)
- **Evening:** Take a stroll along the Magnificent Mile, enjoying the city lights and views of Lake Michigan. Stop for dinner at The Purple Pig, a Mediterranean-inspired restaurant with a lively atmosphere. Try their small plates and wine flights. (40—60 per person)

Day 2: March 23, 2022

- **Morning:** Start the day with a scenic architecture river cruise, which offers a unique perspective on Chicago's iconic buildings and history. (\$80 per person)
- **Lunch:** Grab a bite at Portillo's, a Chicago staple for hot dogs and Italian beef sandwiches. (10—15 per person)
- **Afternoon:** Explore Millennium Park, including the famous "Cloud Gate" sculpture (also known as "The Bean"). Take a leisurely walk through the beautiful Lurie Garden and enjoy the city views. (free admission)
- **Evening:** Enjoy dinner at Girl & the Goat, a trendy restaurant from Top Chef winner Stephanie Izard. Savor their creative, globally-inspired dishes and craft cocktails. (50—70 per person)

Day 3: March 24, 2022

- **Morning:** Visit Wrigley Field, the historic home of the Chicago Cubs, for a guided tour of the stadium. (\$25 per person)
- **Lunch:** Head to Xoco, a casual Mexican spot from Rick Bayless, for delicious street food-style eats like tacos and soups. (10—15 per person)
- **Afternoon:** Take some time to explore the Lincoln Park Zoo, a beautiful green oasis in the heart of the city. (free admission)
- **Evening:** End your trip with a bang at the Signature Room at the 95th, located on the 95th floor of the Willis Tower (formerly Sears Tower). Enjoy stunning views of the city while sipping cocktails and snacking on appetizers. (30—50 per person)

Figure 7. Full Itinerary Generated by Baseline System

In addition to the initial generation, the system conversational refinement was also tested. The users were allowed to modify their itinerary if they were not satisfied with it by submitting additional requests through the interface. The figure 8 shows the system generated an updated itinerary, and the original plan will remain at the top of the interface. This shows that the conversational loops work properly and allows step-by-step updates.

👤 **You:** change my plan to 2 days

🤖 **Bot:** Here is the revised 2-day travel itinerary to Chicago:

Day 1: 2022-03-22

Morning: Start your day with a delicious breakfast at The Black Pearl, a cozy cafe and bakery with an average cost of \$63 and a rating of 4.1. Then, head over to Navy Pier, a lakefront attraction that offers stunning views of Lake Michigan and the city skyline. Take a leisurely walk along the pier and enjoy the street performers and live music.

Afternoon: For lunch, grab a bite at Pantry D'or, a unique eatery that serves a mix of tea, BBQ, fast food, cafe, and Indian cuisine with an average cost of \$53 and a rating of 4.4. After lunch, visit the Skydeck Chicago, which offers breathtaking views of the city from the 103rd floor of the Willis Tower.

Evening: For dinner, head to Fio Cookhouse and Bar, a restaurant that serves a variety of desserts, Mexican, fast food, cafe, and seafood dishes with an average cost of \$31 and a rating of 4.0. End your evening with a visit to Millennium Park, a beautiful urban park that features iconic landmarks like the "Cloud Gate" sculpture and the Jay Pritzker Pavilion.

Accommodation check-in: Newly Renovated Greenpoint Abode, Entire home/apt, \$981.0, Rating: 3.0

Flight info: Outbound flight F3984576, departing Sarasota at 05:14 and arriving in Chicago at 06:50, price \$279.

Day 2: 2022-03-23

Morning: Begin your day with a visit to the Lincoln Park Zoo, a free zoo that's home to over 1,100 animals from around the world. Then, grab breakfast at Gyan Vaishnav, a bakery and dessert shop with an average cost of \$14 and a rating of 3.9.

Afternoon: For lunch, head to Bro's Kitchenette, a casual eatery that serves Mexican, desserts, and seafood dishes with an average cost of \$38 and a rating of 3.3. After lunch, take a stroll along the Riverwalk, a scenic pedestrian path that runs along the south bank of the Chicago River.

Evening: For dinner, try Mini's Royal Cafe, a tea room and bakery with an average cost of \$72 and a rating of 3.3. End your evening with a visit to Buckingham Fountain, a beautiful landmark that features a large fountain and stunning views of the city skyline.

Flight info: Return flight F4010566, departing Chicago at 20:10 and arriving in Sarasota at 23:49, price \$368.

I removed Day 3 and kept the rest of the itinerary unchanged. Let me know if you need any further modifications!

Figure 8. Updated Itinerary

The system conversational refinement was also tested for the baseline system, The revised itinerary has issues like returning flights are missing. This happened because the system directly removes the third day plan in response to the user request and leaving the other parts of the plan unchanged. Figure 9 shows the updated itinerary revised by the baseline system.

👤 **You:** change my plan to 2 days

🤖 **Bot:** Here is the revised itinerary:

Welcome to your 2-day travel itinerary to Chicago!

Day 1: March 22, 2022

- **Morning:** Fly from Sarasota-Bradenton International Airport (SRQ) to Chicago O'Hare International Airport (ORD) on a morning flight. I've booked a round-trip economy ticket on American Airlines for \$450.
- **Check-in:** After arriving in Chicago, take a taxi or Uber to your hotel, the Kimpton Hotel Monaco Chicago (approximately \$150 per night). This stylish hotel is located in the heart of the Loop, within walking distance to many attractions.
- **Lunch:** Head to Lou Mitchell's, a Chicago institution since 1923, for a classic American breakfast or lunch. Try their famous waffles or eggs benedict. (20—30 per person)
- **Afternoon:** Visit the Art Institute of Chicago, one of the top art museums in the country. Spend a few hours exploring the impressive collection, including works by Monet, Van Gogh, and Picasso. (\$20 per person)
- **Evening:** Take a stroll along the Magnificent Mile, enjoying the city lights and views of Lake Michigan. Stop for dinner at The Purple Pig, a Mediterranean-inspired restaurant with a lively atmosphere. Try their small plates and wine flights. (40—60 per person)

Day 2: March 23, 2022

- **Morning:** Start the day with a scenic architecture river cruise, which offers a unique perspective on Chicago's iconic buildings and history. (\$80 per person)
- **Lunch:** Grab a bite at Portillo's, a Chicago staple for hot dogs and Italian beef sandwiches. (10—15 per person)
- **Afternoon:** Explore Millennium Park, including the famous "Cloud Gate" sculpture (also known as "The Bean"). Take a leisurely walk through the beautiful Lurie Garden and enjoy the city views. (free admission)
- **Evening:** Enjoy dinner at Girl & the Goat, a trendy restaurant from Top Chef winner Stephanie Izard. Savor their creative, globally-inspired dishes and craft cocktails. (50—70 per person)

Figure 9. Updated Itinerary Revised by Baseline System

4.5 Discussion

The experimental findings show that the retrieval-guided generation enhances the itinerary accuracy and constraints compared to pure LLM-based generation. Even though LLMs are able to produce fluent text, they do not naturally have mechanisms to strictly follow logical and numerical constraints in planning tasks.

The proposed method achieved improved control, reliability, and personalization by combining structured dataset filtering and rule-based prompt engineering. However, the method still depends on dataset completeness so when no suitable dataset record exists, the generated itinerary diversity may decrease.

5. CONCLUSION AND FUTURE WORK

This study proposed a retrieval-guided generative AI framework for personalized travel itinerary planning. The system combines a structured tourism dataset, rule-based constraints, and an LLM to generate travel itineraries that match user needs. The system is different from other purely generative methods; this framework filters and scores dataset records to make sure the generated itineraries match user needs such as budget, destinations, and travel days.

The results show that the system can extract structured information from user natural language input, select relevant tourism data, and generate day-by-day itineraries that fit with user preferences. The proposed method achieved higher performance compared to the baseline with a BLEU score of 0.59, ROUGE-L score of 0.68, and constraints accuracy of 93%, indicating better text quality and constraints satisfaction. The rule-based controls help to prevent some common problems in LLM-generated travel plans such as repeated attractions, incorrect flight placement, and mismatch of trip duration. In addition, the conversational refinement features allow users to modify their itinerary based on their needs which makes the system more flexible and interactive.

However, there are some limitations in the system. The system depends on the dataset quality and completeness, and the evaluation mainly focuses on preliminary testing rather than user studies. Future work will be focusing on improving itinerary generation consistency, enhancing the system's ability to dynamically revise user plans based on user input, and combining explicit user preference weights to refine recommendations. First, adding real-time APIs such as flight booking systems and weather services can make the itineraries more realistic and useful. Second, improving the scoring and filtering process by giving different weights to user preferences can increase personalization accuracy. Third, evaluation methods such as automatic metrics and user satisfaction surveys can better measure itinerary quality.

In addition, future studies can explore hybrid generation methods that combine retrieval-based data with creative generation when there is no exact data match in the dataset. The system can also expand the dataset to include more regions around the world and can also improve diversity and global usefulness. Lastly, the user interface can be improved, and conducting more usability testing can further enhance the experiences of interactive travel planning.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for the suggestions to improve the paper.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Wei-Jun Lim: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;
Nabhan D. Salih: Project Administration, Writing – Review & Editing.
Hairulnizam Mahdin: Validation, Writing – Review & Editing.
Ammar Alazab: Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. <https://publicationethics.org/>

DATA AVAILABILITY

The data that support the findings of this study are openly available in the TravelPlanner dataset at <https://arxiv.org/abs/2402.01622>. These data were derived from publicly available tourism sources.

REFERENCES

- [1] S. Park and J. L. Nicolau, "Asymmetric effects of online consumer reviews," *Annals of Tourism Research*, vol. 50, pp. 67–83, Jan. 2015, doi: 10.1016/j.annals.2014.10.007.
- [2] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Introduction and challenges," in *Recommender Systems Handbook*, 2nd ed. Boston, MA, USA: Springer, 2015, pp. 1–34, doi: 10.1007/978-1-4899-7637-6_1.
- [3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005, doi: 10.1109/TKDE.2005.99.
- [4] K.-Z. Liao and H. A. Santoso, "The role of generative AI in e-commerce recommender systems: Methods, trends and insights," *Journal of Informatics and Web Engineering*, vol. 4, no. 3, pp. 35–63, Oct. 2025, doi: 10.33093/jiwe.2025.4.3.3.
- [5] Z. B. Ter, P. Naveen, and J. Jayapradha, "Generative AI-based meal recommender system," *Journal of Informatics and Web Engineering*, vol. 4, no. 2, pp. 315–338, Jun. 2025, doi: 10.33093/jiwe.2025.4.2.20.
- [6] E. A. Anaam, S.-C. Haw, and P. Naveen, "Applied fuzzy and analytic hierarchy process in hybrid recommendation approaches for E-CRM," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 2-2, pp. 553–560, Aug. 2022, doi: 10.30630/joiv.6.2-2.1043.
- [7] D. Jannach, A. Manzoor, W. Cai, and L. Chen, "A survey on conversational recommender systems," *ACM Computing Surveys*, vol. 54, no. 5, Art. no. 105, pp. 1–36, 2021, doi: 10.1145/3453154.
- [8] A. Banerjee, A. Satish, F. N. Aisyah, W. Wörmel, and Y. Deldjoo, "Collab-REC: An LLM-based agentic framework for balancing recommendations in tourism," *ACM Transactions on Recommender Systems*, 2026, doi: 10.1145/3837862.
- [9] W. X. Zhao et al., "A survey of large language models," *Frontiers of Computer Science*, vol. 20, Art. no. 2012627, 2026, doi: 10.1007/s11704-026-60308-3.
- [10] M. L. Ferreira and E. Kastenholz, "Generative AI in hospitality and tourism: A dual-stakeholder perspective on tourist and workforce experience dynamics," *Tourism Review*, 2026, doi: 10.1108/TR-08-2025-0968.
- [11] S. Mai and Z. Liu, "Generative AI and the tourist experience: Reconfiguring en-route and post-trip value co-creation and co-destruction," *Current Issues in Tourism*, 2026, doi: 10.1080/13683500.2026.2640203.
- [12] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24824–24837.
- [13] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.
- [14] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, Art. no. 421425, 2009, doi: 10.1155/2009/421425.
- [15] A. D. Solano-Barliza, I. Arregocés-Julio, M. Aarón-Gonzalez, R. Zamora-Musa, E. De-La-Hoz-Franco, J. Escorcia-Gutierrez, and M. Acosta-Coll, "Recommender systems applied to the tourism industry: A literature review," *Cogent Business & Management*, vol. 11, no. 1, Art. no. 2367088, 2024, doi: 10.1080/23311975.2024.2367088.
- [16] V. Udandara, N. A. Tiju, M. Vairamuthu, H. Mistry, and D. Kumar, "Roamify: Designing and evaluating an LLM based Google Chrome extension for personalised itinerary planning," *arXiv preprint arXiv:2504.10489*, 2025, doi: 10.48550/arXiv.2504.10489.
- [17] B. Liu, J. Ge, and J. Wang, "Vaiage: A multi-agent solution to personalized travel planning," *arXiv preprint arXiv:2505.10922*, 2025, doi: 10.48550/arXiv.2505.10922.

-
- [18] K. Otaki and Y. Baba, “Travel itinerary recommendation using interaction-based augmented data,” *Expert Systems with Applications*, vol. 269, Art. no. 126294, Apr. 2025, doi: 10.1016/j.eswa.2024.126294.
- [19] A. Banerjee, A. Satish, F. N. Aisyah, W. Wörndl, and Y. Deldjoo, “SynthTRIPs: A knowledge-grounded framework for benchmark query generation for personalized tourism recommenders,” in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, Padua, Italy, 2025, pp. 3743–3752, doi: 10.1145/3726302.3730321.
- [20] A. N. Fajari and A. Baizal, “Chatbot-based culinary tourism recommender system using named entity recognition,” *JIPPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 4, pp. 1131–1138, 2022, doi: 10.29100/jipi.v7i4.3210.
- [21] J. Xie, K. Zhang, J. Chen, T. Zhu, R. Lou, Y. Tian, Y. Xiao, and Y. Su, “TravelPlanner: A benchmark for real-world planning with language agents,” in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, vol. 235, 2024, pp. 54590–54613.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
- [23] T. B. Brown et al., “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.
- [24] A. Vaswani et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.
- [25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, USA, 2002, pp. 311–318, doi: 10.3115/1073083.1073135.
- [26] E. Doostmohammadi, O. Holmström, and M. Kuhlmann, “How reliable are automatic evaluation methods for instruction-tuned LLMs?,” in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, FL, USA, 2024, pp. 6321–6336, doi: 10.18653/v1/2024.findings-emnlp.367.

BIOGRAPHIES OF AUTHORS

	Wei-Jun Lim is a final-year Data Science Student in Multimedia University Cyberjaya. His research interests include artificial intelligence, generative AI, recommender systems, natural language processing, and data analytics. His current research focuses on developing a Generative AI-based chatbot for personalized tourism recommendation using large language models and structured travel datasets. He continuously seek for new challenges and learning opportunities to enhance his skills. He can be reached out at LIM.WEI.JUN@student.mmu.edu.my.
	Nbhan D. Salih is currently a lecturer at Ohio State University, USA, with prior academic experience as an assistant professor in the Faculty of Computing and Informatics at Multimedia University (MMU), Malaysia. He holds a Ph.D. in Engineering, and his research interests lie at the intersection of computer vision, digital image processing, and machine learning. Dr. Salih has been actively involved in advancing technologies that enable intelligent image analysis and pattern recognition, contributing to both the academic and applied aspects of visual computing. He can be contacted at email: salih.43@osu.edu.
	Hairulnizam Mahdin is a Professor at the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia. He has authored more than 100 publications across Web of Science (WoS), Scopus, and Google Scholar-indexed journals and conferences. His scholarly work spans data engineering, the Internet of Things (IoT), and artificial intelligence (AI). He has also contributed to numerous international conferences in various leadership capacities, including chair, program committee, and vice-chair. He can be contacted at hairuln@uthm.edu.my.
	Ammar Alazab is a Senior Lecturer in Cybersecurity at Torrens University Australia and a member of the Centre for Artificial Intelligence Research and Optimisation (AIRO). He holds a PhD in Computer Science from Deakin University, with expertise spanning AI, cybersecurity, malware detection, federated learning, and blockchain. Author of 60+ publications with 1,400+ citations (H-index 16), he has delivered 50+ talks across academia, industry, and government. His research focuses on AI-powered cybercrime prevention and privacy-preserving cybersecurity analytics.